Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Тамбовский государственный технический университет»

**И. Г. КАРПОВ, Г. Н. НУРУТДИНОВ, А. В. ЯКОВЛЕВ,
А. И. ЕЛИСЕЕВ, Д. В. ПОЛЯКОВ, В. Г. ОДНОЛЬКО**

ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ

ПРАКТИКУМ

Допущено Учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебного пособия для студентов высших учебных заведений,
обучающихся по направлениям подготовки 230400 «Информационные
системы и технологии» и 220100 «Системный анализ и управление»

Тамбов
Издательство ФГБОУ ВО «ТГТУ»
2016

УДК 621.396.6(075)
ББК 32.811.3я73
И71

Рецензенты:

Профессор кафедры «Защита информации» ФГБОУ ВПО «Московский государственный технический университет им. Н. Э. Баумана» (Национальный исследовательский университет техники и технологий), доктор технических наук, профессор
С. В. Скрыль

Доктор технических наук, доцент, заведующий кафедрой «Механотроника и технологические измерения» ФГБОУ ВО «Тамбовский государственный технический университет»
А. Г. Дивин

И71 **Инфокоммуникационные** системы и сети. Практикум : учебное пособие / И. Г. Карпов, Г. Н. Нурутдинов, А. В. Яковлев и др. – Тамбов : Изд-во ФГБОУ ВО «ТГТУ», 2016. – 236 с. – 100 экз.
ISBN 978-5-8265-1597-6.

Представлен материал, необходимый для выполнения практических занятий в рамках учебных дисциплин, объединённых теорией передачи информации, что позволяет студентам совершенствовать навыки и умение в решении задач анализа характеристик инфокоммуникационных систем и сетей и оценки их эффективности.

Рассматриваются основные концепции, определяющие современное состояние и тенденции развития компьютерных сетей. Теоретический материал дополнен примерами конфигурации сетевого оборудования Cisco Systems.

Предназначено для студентов высших учебных заведений, обучающихся по направлениям подготовки 09.03.02 (230400) «Информационные системы и технологии» и 27.03.03 (220100) «Системный анализ и управление».

УДК 621.396.6(075)
ББК 32.811.3я73

ISBN 978-5-8265-1597-6

ВВЕДЕНИЕ

Системы передачи информации (СПИ) представляют собой сложные комплексы, состоящие из различных функционально зависимых элементов. Эти системы характеризуются не только большим числом элементов, но и иерархичностью структуры, наличием между элементами прямых, обратных и перекрёстных связей. Свойства любой системы, прежде всего, определяются её целевым назначением, которое трактуется как совокупность задач, решаемых данной системой. Для получения желаемого результата необходимо совершить определённое число операций, реализуемых за счёт имеющихся ресурсов системы. В СПИ такими операциями являются кодирование, модуляция, усиление сигнала, демодуляция, декодирование, селекция сигналов, маршрутизация, адресование, коммутация и т.п., а ресурсами системы – ёмкость каналов, возможности центров коммутации, мощность сигнала, полоса частот канала и др. Однако СПИ, как техническая система, имеет ряд специфических особенностей, среди которых наиболее существенны объект (продукт) передачи и среда (условия) распространения сигналов. Объектом передачи в СПИ является информация. Для организации обмена информацией между многими источниками и получателями информации каналы и системы передачи объединяют в сети связи – единый комплекс систем передачи и распределения информации. При этом отправителей и получателей информации называют пользователями или абонентами. Понятие распределения информации охватывает задачи распределения маршрутов передачи информации и связанные с этим задачи анализа и синтеза сетей связи и центров коммутации. Задача распределения информации возникла сразу же вслед за созданием устройств её передачи. Примером простейшей СПИ является полносвязная сеть, где абонентские (оконечные) пункты соединены по принципу «каждый с каждым». Такие сети относятся к некоммутируемым сетям, здесь связь между пользователями осуществляется по закреплённым некоммутируемым каналам. К достоинствам некоммутируемых сетей относится то, что в них информация передаётся без потерь времени на ожидание соединения с оконечным пунктом получателя и нет необходимости в передаче адреса вызываемого пункта. Вместе с тем при увеличении числа оконечных пунктов резко возрастает число необходимых линий связи.

В целях сокращения числа необходимых каналов используются коммутируемые СПИ, в которых оконечные пункты соединяют между собой не непосредственно, а через один или несколько коммутационных центров. Во многих случаях (особенно при односторонней передаче информации) можно использовать способ коммутации сообщений. В таких системах, построенных на базе ЭВМ, передаваемые сообщения, сопровождаемые адресом, принимаются на оконечных пунктах отправителя без отказа, обрабатываются и накапливаются в памяти коммутационных центров. Передача информации в адрес оконечного пункта получателя производится по мере освобождения необходимых каналов. При этом естественно возникает задержка, затрудняющая передачу информации в реальном масштабе времени, например при телефонном разговоре.

Для сокращения времени задержки в современных СПИ используется разновидность способа коммутации сообщений, называемая коммутацией пакетов. В сетях с коммутацией пакетов от источника к получателю передаются короткие блоки данных, называемые пакетами. Наиболее типичные виды нагрузки создаёт передача данных в интерактивном диалоговом режиме, когда между терминалами (в том числе и компьютерами) передаются короткие пачки сообщений. Под пакетом понимается часть сообщения, представленная в виде блока с заголовками, имеющего установленный формат и ограниченную длину, передаваемая по сети как часть единого целого. Функции и требования к центрам коммутации каналов и коммутации пакетов совершенно разные. Однако по мере развития интегральных сетей (в том числе и информационно-вычислительных), в которых обрабатываются речь, данные и другие виды сообщений и используется техника коммутации пакетов, каналов и гибридная техника, функции и требования к коммутационным центрам становятся сходными. При увеличении числа оконечных пользователей и их территориальной разобщённости возникает задача выбора структуры сети, размещения коммутационных центров, определения пропускной способности информационных направлений и числа каналов.

В последние годы интенсивно развиваются процессы конвергенции и интеграции современных компьютерных и традиционных сетей связи и появляются инфокоммуникационные сети, начиная от корпоративных и заканчивая сетями национального и глобального масштабов. Сетевые технологии, такие, как синхронная цифровая иерархия, асинхронный режим передачи, сверхплотное волновое мультиплексирование и другие, не только открывают новые возможности в построении современных СПИ, но и требуют специального изучения.

Разобраться в существующем многообразии различных систем и методов передачи информации, понять общие принципы их построения и функционирования поможет данное учебное пособие.

Часть I
МОБИЛЬНЫЕ СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

СПИСОК СОКРАЩЕНИЙ

АИМ – амплитудно-импульсная модуляция
АДИКМ – адаптивная дифференциальная импульсно-кодовая модуляция
АМн – амплитудная манипуляция
АЦП – аналого-цифровой преобразователь
АЧХ – амплитудная частотная характеристика
БС – биортогональный сигнал
ДИКМ – дифференциальная импульсно-кодовая модуляция
ДМ – дельта-модуляция
ДО – детектор огибающей
ДП – деперемежитель
ДЧМн – дискретная частотная манипуляция
ИДМ – импульсный демодулятор
ИМ – импульсный модулятор
КАМ – квадратурная амплитудная манипуляция
КФ – корреляционная функция
КЧХ – комплексная частотная характеристика
ЛЗ – линия задержки
ОС – ортогональный сигнал
ОШИМ – односторонняя широтно-импульсная модуляция
П – перемежитель
ПРВ – плотность распределения вероятностей
ПРМ – приёмник
ПРД – передатчик
ПСП – псевдослучайная последовательность
ПФ – полосовой фильтр
ППРЧ – псевдослучайная перестройка рабочей частоты
СМ – смеситель
СФ – согласованный фильтр
СП – случайный процесс
СПИ – система передачи информации
УПС – узкополосный сигнал
ФИМ – фазоимпульсная модуляция
ФМн – фазовая манипуляция
ФНЧ – фильтр нижних частот
ФЧХ – фазочастотная характеристика

ЧМн	– частотная манипуляция	
ЧВМ	– частотно-временная матрица	
ЦАП	– цифро-аналоговый преобразователь	
ШИМ	– широтно-импульсная модуляция	
ШСПИ	– широкополосные системы передачи информации	
ШПС	– широкополосный сигнал	
RC	Radio Configuration	– набор параметров модуляции расширения спектра
SR	Spreading Rate	– число несущих
BPSK	Binary Phase Shift Keying	– двоичная фазовая манипуляция
QPSK	Quadrature Phase Shift Keying	– квадратурная фазовая манипуляция
PSK	Phase Shift Keying	– фазовая манипуляция
QAM	Quadrature – Amplitude Modulation	– квадратурная амплитудная манипуляция
CDMA	Code Division Multiple Access	– множественный доступ с кодовым разделением

Практическое занятие 1

КОРРЕЛЯЦИОННАЯ ФУНКЦИЯ И СПЕКТРАЛЬНАЯ ПЛОТНОСТЬ СЛУЧАЙНЫХ ПРОЦЕССОВ

Цель: с помощью прикладного пакета программ Mathcad проанализировать зависимость между видом спектральной плотности и корреляционной и функции случайного процесса (СП).

В результате выполнения практического занятия обучаемые *должны*:

− *знать* основные сведения о спектральной плотности и корреляционной функции СП;

− *уметь* провести спектральный и корреляционный анализ сигналов.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [1, с. 42 – 67].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. СЛУЧАЙНЫЕ ПРОЦЕССЫ В СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

В качестве математических моделей сигналов и помех в системах передачи информации (СПИ) используют случайные процессы. *Случайным процессом* $\xi(t)$ принято называть случайную функцию аргумента t, где t – текущее время. Стационарным случайным процессом в узком смысле называется СП, у которого n-мерная плотность распределения вероятностей (ПРВ) не изменится, если все отсчёты времени сместить на одну и ту же величину.

Есть много физических ситуаций, когда статистические характеристики процесса не изменяются на интервале времени наблюдения. В этих случаях предположение о стационарности приводит к удобной математической модели, которая является достаточно точной аппроксимацией реальной ситуации.

Для стационарного СП двумерная ПРВ и, соответственно, корреляционная функция (КФ) зависят не от t_1 и t_2 в отдельности, а только от их

разности $\tau = t_2 - t_1$. В соответствии с этим КФ стационарного СП определяется выражением

$$R_\xi(\tau) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x_1 - m_\xi)(x_2 - m_\xi)\, p_\xi(x_1, x_2; \tau)dx_1dx_2 = \\ = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1x_2\, p_\xi(x_1, x_2; \tau)dx_1dx_2 - m_\xi^2, \tag{1}$$

где m_ξ – математическое ожидание стационарного СП; x_1, x_2 – возможные значения СП, соответственно, в моменты времени t_1, t_2; $\tau = t_2 - t_1$ – интервал времени между сечениями; $p_\xi(x_1, x_2; \tau)$ – двумерная ПРВ стационарного процесса.

Корреляционная функция $R_\xi(\tau)$ стационарного СП является действительной функцией аргумента τ. При этом $R_\xi(\tau)$ характеризует $\xi(t)$ с двух сторон. *Во-первых*, определяет среднюю удельную мощность флюктуаций, а *во-вторых*, позволяет судить о степени линейной связи между двумя сечениями СП, отстоящими друг от друга на интервал времени τ. Размерность $R_\xi(\tau)$ совпадает с размерностью квадрата СП. Корреляционная функция при $\tau = 0$ равна дисперсии процесса:

$$R_\xi(0) = D_\xi. \tag{2}$$

Корреляционная функция может быть представлена в виде

$$R_\xi(\tau) = D_\xi r(\tau), \tag{3}$$

где $r(\tau)$ – *нормированная корреляционная функция*, имеющая смысл коэффициента корреляции, зависящего от τ и заключённая в пределах

$$-1 \le r(\tau) \le 1. \tag{4}$$

Она характеризует только степень линейной связи между сечениями случайного процесса, взятыми через интервал τ. В свою очередь, дисперсия D_ξ процесса характеризует только среднюю удельную мощность флюктуаций СП.

На практике важным параметром является *интервал корреляции* τ_k, который характеризует эффективную ширину КФ. С общих позиций интервал корреляции определяется выражением

$$\tau_k = \int_0^{\infty} |r(\tau)|\, d\tau. \tag{5}$$

Численно τ_k равен основанию прямоугольника с высотой $r(0) = 1$, имеющего ту же площадь, что и фигура, ограниченная кривой $r(\tau)$, при $\tau \geq 0$. Интервал корреляции τ_k определяет тот временной интервал τ между сечениями случайного процесса, при превышении которого эти сечения считаются некоррелированными.

Спектральной плотностью стационарного СП $\xi(t)$ называется функция частоты $S_\xi(\omega)$, являющаяся преобразованием Фурье от корреляционной функции этого процесса:

$$S_\xi(\omega) = \int_{-\infty}^{\infty} R_\xi(\tau) e^{-j\omega\tau} d\tau. \qquad (6)$$

Если существует прямое преобразование, то существует и обратное преобразование Фурье, которое по известной $S_\xi(\omega)$ определяет $R_\xi(\tau)$, если $\tau = 0$, получим

$$D_\xi = R_\xi(0) = \int_{-\infty}^{\infty} S_\xi(f) df. \qquad (7)$$

Как известно, D_ξ определяет среднюю удельную мощность флюктуаций случайного процесса. Следовательно, функция частоты $S_\xi(\omega)$, от которой берётся интеграл по всем частотам, в результате чего находится D_ξ, характеризует среднюю мощность процесса, приходящуюся на единицу полосы частот. Размерностью $S_\xi(\omega)$ является размерность квадрата СП, поделённая на размерность частоты. Если $\xi(t)$ напряжение, то размерностью $S_\xi(f)$ является [В^2/Гц]. Заметим, что размерность $S_\xi(f)$ совпадает с размерностью энергии [В^2/Гц] = [В^{2}с]. Поэтому $S_\xi(f)$ иногда называют *энергетическим спектром*.

Спектральную плотность $S_\xi^+(f)$, определённую на $f \geq 0$, будем называть *физическим спектром*, а спектральную плотность $S_\xi(f)$, определённую на $-\infty < f < \infty$, – *математическим спектром* случайного процесса.

Ширина $S_\xi^+(f)$ оценивается *эффективной шириной спектра* $\Delta f_{эф}$:

$$\Delta f_{эф} = \frac{1}{S_{\xi\max}^+} \int_0^{\infty} S_\xi^+(f) df, \qquad (8)$$

которая определяет основание прямоугольника с высотой 1, имеющего ту же площадь, что и фигура, ограниченная кривой $S_\xi^+(f) / S_{\xi\max}^+$.

Наиболее применимы следующие частные модели сигналов:

1. *Детерминированный сигнал*, т.е. сигнал с полностью известными параметрами

$$s(t) = A(t)\cos(\omega_0 t + \psi(t) - \varphi_0). \tag{9}$$

Его удобно использовать для получения потенциальных (предельных) характеристик оптимальных приёмников.

2. *Сигнал со случайной начальной фазой*

$$s(t, \varphi) = A(t)\cos(\omega_0 t + \psi(t) - \varphi). \tag{10}$$

Здесь начальная фаза φ принимаемого сигнала полагается случайной величиной с равномерным законом распределения.

3. *Сигнал со случайными амплитудой и начальной фазой*

$$s(t, a, \varphi) = aA(t)\cos(\omega_0 t + \psi(t) - \varphi). \tag{11}$$

При этом начальная фаза φ распределена равномерно на интервале $[-\pi, \pi]$, а безразмерный параметр a, определяющий амплитуду сигнала, распределён по релеевскому закону.

Белым шумом называется стационарный СП, имеющий нулевое математическое ожидание $m_n = 0$ и одинаковое значение спектральной плотности $S_n(\omega) = N_0/2$ на всей оси частот от $-\infty$ до $+\infty$ (рис. 1).

Корреляционная функция белого шума

$$R_n(\tau) = \frac{1}{4\pi}\int_{-\infty}^{\infty} N_0 \exp(j\omega\tau)\, d\omega = \frac{N_0}{2}\delta(\tau), \tag{12}$$

т.е. имеет дельтообразный вид, а интервал корреляции $\tau_к = 0$. Таким образом, в белом шуме отсутствует взаимосвязь между предыдущими и последующими значениями. Это свойство выделяет его среди других случайных процессов и определяет его особую роль как простейшей модели помех. Белый шум физически нереализуем, так как ему соответствует бесконечно большая дисперсия, а, следовательно, бесконечно большая мгновенная мощность.

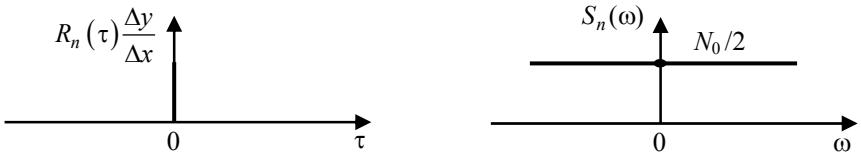


Рис. 1. График КФ и спектральной плотности белого шума

Дисперсия выходного процесса $\eta(t)$, когда на входе линейной цепи действует белый шум,

$$D_\eta = N_0 K_0^2 \Delta f_ш . \qquad (13)$$

Здесь K_0^2 квадрат модуля комплексной частотной характеристики (КЧХ) цепи, а $\Delta f_ш$ – *шумовая полоса* этой цепи, которая определяет при белом шуме на входе эффективную ширину спектральной плотности выходного процесса. Таким образом, спектральный анализ с использованием КЧХ линейной системы позволяет определить энергетический спектр выходного стационарного процесса, который равен произведению квадрата модуля КЧХ линейной системы на энергетический спектр входного процесса.

Стационарным случайным процессом в узком смысле называется СП, у которого *n*-мерная ПРВ не изменится, если все отсчёты времени сместить на одну и ту же величину:

$$p_\xi(x_1, ..., x_n ; t_1, ..., t_n) = p_\xi(x_1, ..., x_n) = p_\xi(x, ..., x_n ; t_1 - \Delta t, ..., t_n - \Delta t). \qquad (14)$$

Математическое ожидание и дисперсия стационарного процесса не зависят от времени, а КФ зависит от τ:

$$m_\xi = \int_{-\infty}^{\infty} x p_\xi(x) dx , \qquad (15)$$

$$D_\xi = \int_{-\infty}^{\infty} (x - m_\xi)^2 p_\xi(x) dx, \qquad (16)$$

$$R_\xi(\tau) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_1 - m_\xi)(x_2 - m_\xi) p_\xi(x_1, x_2; \tau) dx_1 dx_2 . \qquad (17)$$

Математическое ожидание m_ξ постоянно и поэтому для стационарного процесса характеризует постоянную составляющую процесса; постоянность D_ξ характеризует то, что в каждой точке времени t средняя удельная мощность флюктуаций (т.е. мощность переменной составляющей) одна и та же; зависимость $R_\xi(\tau)$ от τ означает, что для стационарного процесса неважно, в каких точках t_1 и t_2 берутся сечения, важна только разность между ними $\tau = t_2 - t_1$.

Контрольные вопросы

1. Дать определение спектральной плотности СП.
2. Дать определение корреляционной функции СП.

3. Записать выражение, связывающее корреляционную функция и спектральную плотность СП.

4. Дать определение белого шума. Нарисовать графики его спектральной плотности и корреляционной функции.

Задачи для самостоятельного решения

Задача 1. По известной спектральной плотности (либо корреляционной функции (КФ)) определить корреляционную функцию (либо спектральную плотность) на входе цепи, используя выражения

$$R_\xi(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_\xi(\omega)\exp(j\omega\tau)\,d\omega,$$

$$S_\xi(\omega) = \int_{-\infty}^{\infty} R_\xi(\tau)\exp(-j\omega\tau)\,d\tau.$$

Построить графики $R_\xi(\tau) = 16\exp\left(-\alpha^2\tau^2\right)\cos(\omega_0\tau)$ и соответствующую КФ $S_\xi(\omega)$. Скопировать графики в отчёт. По графику спектральной плотности найти эффективную ширину спектральной плотности. Определить дисперсию случайного процесса

$$D_\xi = R_\xi(0).$$

Задача 2. Построить график нормированной КФ, пользуясь выражением

$$r_\xi(\tau) = \frac{R_\xi(\tau)}{D_\xi} = \frac{1}{2\pi D_\xi}\int_{-\infty}^{\infty} S_\xi(\omega)\exp(j\omega\tau)\,d\omega,$$

и скопировать полученные графики в свой отчёт. По графику нормированной КФ найти интервалы корреляции СП.

Практическое занятие 2

СОГЛАСОВАННЫЕ ФИЛЬТРЫ

Цель: с помощью прикладного пакета программ Mathcad произвести временной анализ прохождения сигнала либо суммы сигнала и гауссовского шума через согласованный фильтр (СФ).

В результате выполнения практического занятия обучаемые *должны*:

– *знать* основные сведения о согласованном с сигналом фильтре;

– *уметь* синтезировать структуру СФ.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [1, с. 59 – 72].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. МЕТОД СОГЛАСОВАННОЙ ФИЛЬТРАЦИИ

Согласованным фильтром называется линейная цепь, которая для определённой аддитивной смеси сигнала и шума обеспечивает на выходе наибольшее отношение сигнал/шум. Согласованный фильтр можно рассматривать как оптимальный, у которого критерием оптимальности является достижение максимума отношения сигнал/шум. Для согласованного фильтра неважно как искажается выходной сигнал по отношению к входному. Важно, чтобы при этом достигалось максимально возможное по отношению к любым другим фильтрам отношение сигнал/шум на выходе.

Импульсная характеристика СФ $h_{\text{СФ}}(t)$ определяется выражением

$$h_{\text{СФ}}(t) = ks(t_0 - t)\,, \qquad (1)$$

где k – коэффициент пропорциональности.

В момент времени t_0 окончания сигнала достигается максимум отношения сигнал/шум по мощности на выходе СФ

$$q_{\text{СФ}} = \frac{2E_s}{N_0}, \qquad (2)$$

где $E_s = \int_0^{t_0} s^2(t)\,dt$ – энергия сигнала.

На рисунке 1 показана методика построения импульсной характеристики $h_{\text{СФ}}(t)$, когда известна форма сигнала $s(t)$. Пусть сигналом является треугольный импульс с амплитудой U_m и длительностью t_0 (рис. 1, *а*). Строим его зеркальное отображение $s(–t)$ путём поворота импульса вокруг оси ординат (рис. 1, *б*). Затем задерживаем импульс на время t_0 и изменяем масштаб по оси ординат, т.е. учитываем коэффициент пропорциональности k (рис. 1, *в*).

Согласованный фильтр – как коррелятор. Пусть СФ согласован с сигналом $S(t)$, т.е. импульсная характеристика фильтра определяется выражением (1). Подадим на вход фильтра реализацию $x(t)$ произвольного процесса $\xi(t)$ и найдём отклик фильтра в момент времени t_0, равный длительности сигнала $s(t)$, с которым фильтр согласован.

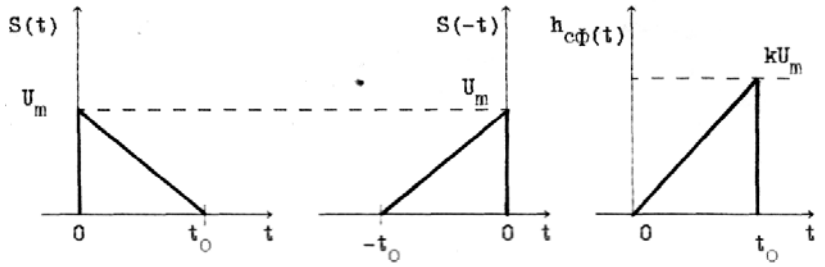
В произвольный момент времени t процесс на выходе равен $y(t) = \int_0^t h(t_1)\,x(t-t_1)\,dt_1$.

Для СФ справедливо выражение

$$y(t) = k \cdot \int_0^t s(t_0 - t_1)x(t - t_1)dt_1, \qquad (3)$$

которое при $t = t_0$ имеет вид

$$y(t_0) = k \cdot \int_0^{t_0} s(t_0 - t_1)x(t_0 - t_1)dt_1 \cdot$$



а) *б*) *в*)

Рис. 1. Пример построения импульсной характеристики СФ

В свою очередь, заменяя под интегралом ($t_0 - t_1$) на t, а t_0 на T, получим

$$y(T) = k \cdot \int_0^T s(t)x(t)dt \,. \tag{4}$$

Выражение (4) пропорционально взаимному корреляционному интегралу (3) между наблюдаемой реализацией $x(t)$ процесса $\xi(t)$ и копией сигнала $s(t)$, с которым фильтр согласован. Если выбрать $k = 2/N_0$ то совпадение (4) и (3) будет полным. Поэтому согласованный фильтр широко используется в оптимальном приёме для вычисления взаимного корреляционного интеграла (4).

Комплексная частотная характеристика (КЧХ) согласованного фильтра может быть найдена как преобразование Фурье от $h_{\text{сф}}(t)$, определяемой выражением (1):

$$K_{\text{СФ}}(j\omega) = \int_{-\infty}^{\infty} h_{\text{СФ}}(t)\, e^{-j\omega t} dt = k \int_{-\infty}^{\infty} s(t_0 - t)\, e^{-j\omega t} dt \,.$$

Сделав замену переменных $\tau = t_0 - t$, получим

$$K_{\text{СФ}}(j\omega) = k e^{-j\omega t_0} \int_{-\infty}^{\infty} s(\tau)\, e^{j\omega t} d\tau \,. \tag{5}$$

Интеграл в формуле (5) определяет комплексно-сопряжённый спектр сигнала

$$\int_{-\infty}^{\infty} s(\tau)\, e^{j\omega t} d\tau = \int_{-\infty}^{\infty} s(t)\, e^{j\omega t} dt = S(-j\omega) = S^*(j\omega),$$

так как в показателе экспоненты стоит знак плюс, а не минус, как это необходимо для определения спектра сигнала.

Таким образом, КЧХ согласованного фильтра

$$K_{\text{СФ}}(j\omega) = k S^*(j\omega)\; e^{-j\omega t_0} \tag{6}$$

пропорциональна произведению комплексно-сопряжённого спектра сигнала $S^*(j\omega)$ на множитель задержки $e^{-j\omega t_0}$. Представим комплексный спектр $S(j\omega)$ сигнала $S(t)$ в виде

$$S(j\omega) = S(\omega)\, e^{j\varphi_S(\omega)}, \tag{7}$$

где $S(\omega)$ и $\varphi_S(\omega)$ – соответственно амплитудный и фазовый спектры сигнала.

Комплексно-сопряжённый спектр сигнала будет отличаться от (7) только знаком показателя экспоненты:

$$S^*(j\omega) = S(-j\omega) = S(\omega)\, e^{j\varphi_S(\omega)}. \qquad (8)$$

Подставив (8) в (6), получим

$$K_{\text{СФ}}(j\omega) = K_{\text{СФ}}(\omega) e^{j\varphi_{\text{СФ}}(\omega)}, \qquad (9)$$

где $K_{\text{СФ}}(\omega) = kS(\omega)$ – амплитудно-частотная характеристика (АЧХ) СФ; $\varphi_{\text{СФ}}(\omega) = -[\varphi_S(\omega) + \omega t_0]$ – фазочастотная характеристика (ФЧХ) СФ.

Пропорциональность АЧХ согласованного фильтра амплитудному спектру сигнала приводит к тому (рис. 2), что коэффициенты передачи фильтра больше на тех частотах, на которых выше амплитуда спектральных составляющих сигнала, и меньше там, где амплитуда ниже.

ФЧХ согласованного фильтра определяется взятой с обратным знаком суммой фазового спектра сигнала $\varphi_S(\omega)$ и пропорционального частоте ω угла задержки ωt_0. Возьмём одну гармоническую составляющую спектра сигнала на произвольной частоте ω, имеющую (для простоты изложения) конечную амплитуду $S(\omega)$:

$$s_\omega(t) = S(\omega)\cos[\omega t + \varphi_S(\omega)].$$

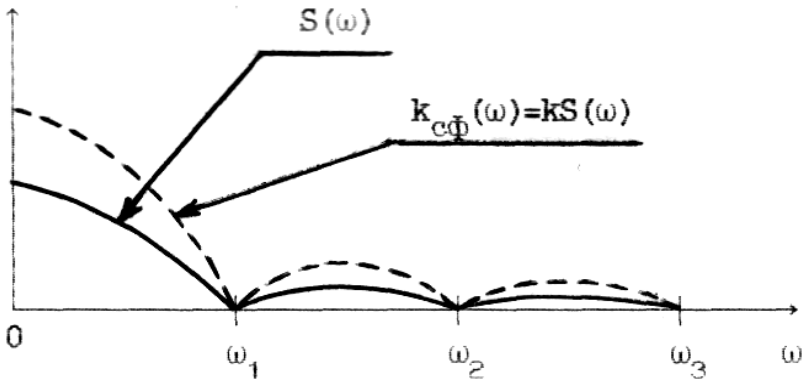


Рис. 2. АЧХ СФ и амплитудный спектр сигнала

Эта составляющая, пройдя через фильтр, увеличит свою амплитуду в $kS(\omega)$ раз и получит фазовую задержку, равную $-[\varphi_S(\omega) + \omega t_0]$:

$$s_{\omega\,\text{вых}}(t) = kS^2(\omega)\cos[\omega t + \varphi_S(\omega) - \varphi_S(\omega) - \omega t_0].$$

В момент времени $t = t_0$ гармоническая составляющая будет равна своей амплитуде:

$$s_{\omega\text{вых}}(t_0) = k\,S^2(\omega)\,. \qquad (10)$$

Так как частота составляющей спектра $s_\omega(t)$ была выбрана произвольно, то можно сделать следующий вывод: на выходе согласованного фильтра в момент времени $t = t_0$ все гармонические составляющие равны своим амплитудным составляющим. Благодаря этому выходной сигнал $s_{\text{вых}}(t)$ в момент времени $t = t_0$ формируется в результате арифметического сложения всех амплитуд гармонических составляющих выходного спектра.

Таким образом, $K_{\text{СФ}}(\omega)$ и $\varphi_{\text{ср}}(\omega)$ подобраны так, чтобы обеспечить максимум пика выходного сигнала при $t = t_0$ и в соответствии с этим получить наибольшее отношение сигнал/шум. При этом форма выходного сигнала не будет совпадать с формой входного сигнала. Более того, искажение формы здесь принципиально необходимо, чтобы получить наибольшее пиковое отношение сигнал/шум на выходе. Кроме того, заметим, что все характеристики СФ, например $h_{\text{СФ}}(t)$ и $K_{\text{СФ}}(\omega)$, при белом шуме на входе полностью определяются характеристиками сигнала $s(t)$. Момент t_0 совпадает с длительностью импульсного сигнала, если импульс одиночный, или с длительностью пачки импульсов, если сигнал представляется в виде нескольких импульсов, образующих пачку.

Рассмотрим теперь на конкретном примере построение структурной схемы СФ по комплексной частотной характеристике.

Пример. Записать выражение для комплексной частотной характеристики СФ и изобразить его функциональную схему, если сигнал представляет собой одиночный прямоугольный видеоимпульс с длительностью $\tau_и$.

Решение: 1. Определяем КЧХ фильтра по формуле (6). Комплексно-сопряжённый спектр входного сигнала $S^*(j\omega)$ получим на основе известных соотношений:

$$S(j\omega) = \int_{-\infty}^{\infty} s(t)\, e^{-j\omega t} dt = \int_{0}^{\tau_u} U_m\, e^{-j\omega t} dt = \frac{U_m}{-j\omega}(e^{-j\omega\tau_u} - 1),$$

$$S^*(j\omega) = S(-j\omega) = \frac{U_m}{j\omega}(e^{-j\omega\tau_u} - 1).$$

Следовательно,

$$K_{\text{СФ}}(j\omega) = \frac{kU_m}{j\omega}(e^{j\omega\tau_u} - 1)\, e^{-j\omega t_0}.$$

Полагая время наблюдения $t_0 = \tau_u$, получим

$$K_{\text{СФ}}(j\omega) = \frac{kU_m}{j\omega}(e^{j\omega\tau_u} - 1)\, e^{-j\omega\tau_u} = \frac{kU_m}{j\omega}(e^{j\omega\tau_u} - 1). \qquad (11)$$

2. Строим функциональную схему фильтра. Выражение (11) для $K_{\text{СФ}}(j\omega)$ позволяет реализовать фильтр двумя способами, представленными на рис. 3.

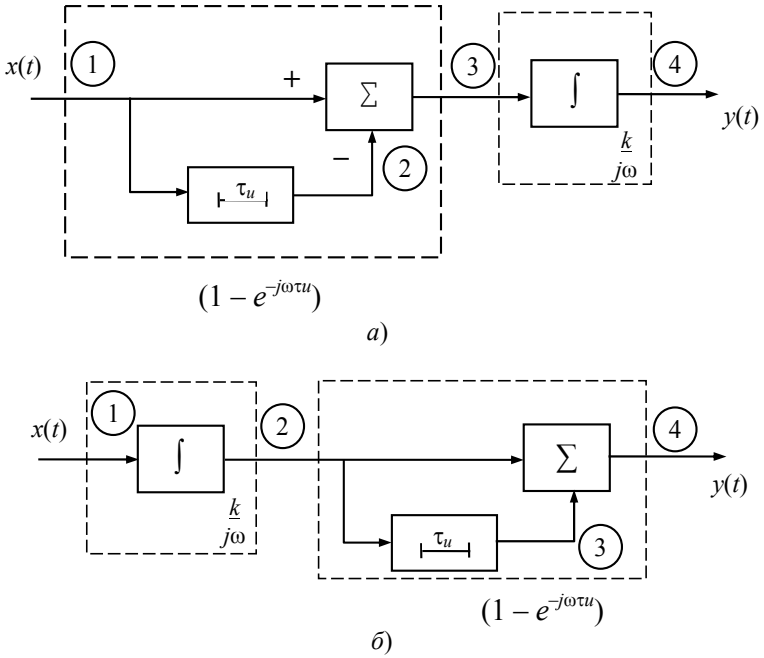


Рис. 3. Способы реализации согласованного фильтра:
а – сначала СФ; *б* – сначала избирательная система

Если на вход СФ поступает пачка из n видеоимпульсов с периодом повторения τ, то выражение для комплексной частотной характеристики имеет вид

$$K(j\omega) = \frac{kU_m}{j\omega}(1 - e^{j\omega\tau_u})(1 + e^{-j\omega\tau} + e^{-j2\omega\tau} + \cdots + e^{-j(n-1)\omega\tau}). \qquad (12)$$

В этом случае в схему фильтра (рис. 3) необходимо дополнительно добавить многоотводную линию задержки и сумматор. Сигнал на выходе СФ будет состоять из $2n - 1$ треугольных импульсов, имеющих различную амплитуду. Амплитуда n-го импульса будет в n раз больше амплитуды первого и последнего импульсов.

Если на вход СФ вместо видеоимпульса подаётся радиоимпульс, то в структурной схеме на рис. 3 необходимо интегратор заменить на колебательный контур.

Контрольные вопросы

1. Что понимают под потенциальной помехоустойчивостью?
2. Сформулируйте основные задачи оптимального приёма.
3. Поясните понятие функции правдоподобия при дискретном и непрерывном наблюдениях.
4. Нарисуйте схему корреляционного приёмника и поясните принцип его работы.
5. Поясните суть метода частотной фильтрации.
6. Поясните суть метода накопления и когда его можно использовать.
7. Какой линейный фильтр называется согласованным? Запишите формулу для отношения сигнал/шум на выходе СФ.
8. Запишите выражение для импульсной характеристики СФ? Как можно построить импульсную характеристику СФ?
9. Объясните принцип когерентного сложения спектральных составляющих при согласованной линейной фильтрации.
10. Запишите формулы для КЧХ, АЧХ и ФЧХ СФ.
11. Изобразите график сигнала на выходе СФ, если на его вход подаётся пачка из n видеоимпульсов.
12. Изобразите структурную схему СФ для пачки из n видеоимпульсов.

Задачи для самостоятельного решения

Задача 1. Рассчитать комплексную частотную характеристику (КЧХ) согласованного фильтра (СФ). Расчёт КЧХ производить по формуле

$$K(j\omega) = \int_0^{\infty} h(t)\exp(-j\omega t)\,dt\,, \tag{13}$$

где $h(t)$ – импульсная характеристика СФ, определяемая выражением

$$h(t) = k\,s(T_0 - t)\,. \tag{14}$$

В формуле (14) k – коэффициент пропорциональности; T_0 – момент окончания наблюдения сигнала; $s(t)$ – пачка из трёх видеоимпульсов с длительностью $\tau = 2$ мс и периодом повторения $T = 4\tau$.

Построить структурную схему СФ, используя полученное выражение для КЧХ СФ.

Определить амплитудную частотную характеристику (АЧХ) СФ, используя соотношение

$$K(\omega) = \left|K(j\omega)\right| = \sqrt{K(j\omega)\cdot K(-j\omega)}\,. \tag{15}$$

Построить графики АЧХ и импульсной характеристики СФ и скопировать их в свой отчёт. Для построения графика импульсной характеристики использовать выражение (14), а графика АЧХ – выражение (15). При этом вначале строятся графики АЧХ и импульсной характеристики СФ для одиночного импульса, а затем для заданной пачки импульсов.

С помощью соотношений

$$y(t_1) = \int_0^{t_1} s(t)\,h(t_1 - t)\,dt\,, \tag{16}$$

$$z(t_1) = \int_0^{t_1} s(t_1)\,s(t)\,dt \tag{17}$$

построить графики сигналов на выходе СФ и коррелятора (вначале для одиночного импульса, а затем для заданной пачки импульсов) и скопировать их в свой отчёт.

Для моделирования гауссовского шума $n(t)$ использовать имеющуюся в пакете программ Mathcad статистическую функцию $r_{\text{norm}}(Z,\ m,\ \sigma)$,

где Z – количество отсчётов шума, m и σ – параметры гауссовского закона. Построить гистограмму и кривую гауссовской плотности вероятности и скопировать их в свой отчёт. Значения Z, m, σ задать самостоятельно.

С помощью соотношений

$$y(t_1) = \int_0^{t_1} \left[s(t) + n(t) \right] h(t_1 - t)\, dt\,, \tag{18}$$

$$z(t_1) = \int_0^{t_1} \left[s(t) + n(t) \right] s_0(t)\, dt \tag{19}$$

построить графики смеси сигнала и шума либо только шума на выходе СФ и коррелятора и скопировать их в свой отчёт. В формулах (18) и (19) сигнал $s(t)$ по форме представляет собой отрезок синусоиды, $s_0(t)$ – опорный сигнал (полагать, что $s_0(t) = s(t)$). Построить также графики сигнала $s(t)$ и импульсной характеристики СФ и скопировать их в свой отчёт.

По графикам АЧХ СФ найти приближённо ширину полосы пропускания СФ для одиночного импульса, а затем для заданной пачки импульсов.

Практическое занятие 3

ЦИФРОВЫЕ СИГНАЛЫ В МОБИЛЬНЫХ СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

Цель: совершенствование теоретических знаний по радиоинтерфейсу мобильных систем

В результате выполнения практического занятия обучаемые *должны*:

– *знать* методику оценки помехоустойчивости приёма цифровых сигналов;

– *уметь* оценить потенциальную помехоустойчивость приёма цифровых сигналов в условиях помех и провести сравнительный анализ эффективности применения различных сигналов.

Практическое занятие включает три этапа:

1. Предварительная подготовка к занятию – проработка теоретического материала студентами на занятии и в ходе самостоятельной работы.
2. Основная часть – письменный опрос и решение задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [1, с. 88 – 96]; [2, с. 7 – 21].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. СТРУКТУРНАЯ СХЕМА СИСТЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Обобщённая структурная схема системы передачи информации (СПИ) представлена на рис. 1.

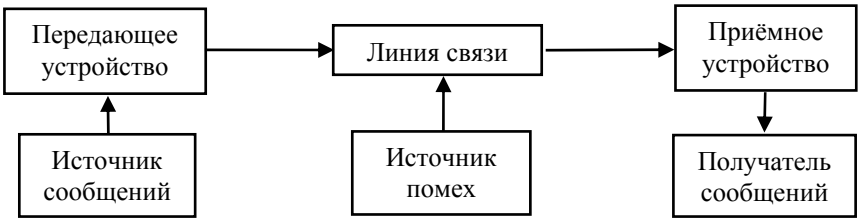


Рис. 1. Обобщённая структурная схема системы передачи информации

Под *источником сообщений* понимают источник сообщений разной природы и преобразователь неэлектрической величины в первичный электрический сигнал.

Передающее устройство предназначено для преобразования сообщения в сигнал. Полезная информация в такие сигналы вводится в процессе *модуляции*, которая заключается в изменении одного или нескольких параметров передаваемого сигнала по закону передаваемого сообщения. Устройство, осуществляющее эту операцию в передатчике, называется *модулятором*.

Полезная информация в них закладывается в процессе модуляции, которая заключается в изменении амплитуды, частоты или фазы высокочастотных колебаний по закону первичного электрического сигнала соответствующего сообщению.

В системах передачи дискретных сигналов передающее устройство несколько усложняется и его структура, в общем случае, соответствует приведённой на рис. 2 структурной схеме системы передачи дискретных сообщений.

Помимо операции модуляции при передаче дискретных сообщений в передающем устройстве реализуются операция кодирования.

Линия связи – это физическая среда, используемая для передачи сигналов. В радиолиниях подобной средой служит область пространства, в которой распространяются электромагнитные волны от передатчика к приёмнику.

В реальных системах сигнал передаётся при наличии *помех*, под которыми понимаются любые случайные воздействия, накладывающиеся на сигнал и затрудняющие его приём. В радиосистемах помехи подразделяются на внешние и внутренние.

Внешние помехи принимаются антенной вместе с полезным сигналом и создаются электромагнитными процессами, происходящими в среде распространения радиоволн и средствами преднамеренной постановки помех.

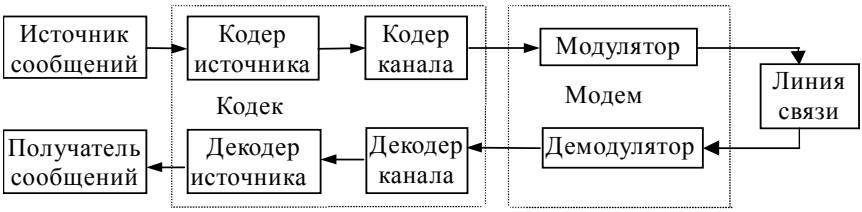


Рис. 2. Структурная схема системы передачи дискретных сообщений

Внутренние помехи локализованы в различных элементах системы передачи информации. Характеристики внутренних помех приёмного устройства обычно пересчитывают к его входу (приводятся к внешним помехам).

Большинство внешних и внутренних помех относятся к классу *аддитивных*, когда сигнал на входе приёмного устройства (наблюдение) можно представить в виде

$$\xi(t) = s(t) + x(t),$$

где $s(t)$ – передаваемый сигнал; $x(t)$ – помеха (случайная функция времени).

Аддитивные помехи подразделяются на флуктуационные, импульсные и синусоидальные.

Флуктуационные помехи – это внутренние шумы приёмника, а также шумы среды распространения сигнала (линии связи). Их спектр обычно намного шире полосы пропускания приёмника, поэтому флуктуационную помеху часто рассматривают как аддитивный белый гауссовский шум (БГШ).

Импульсные помехи представляют собой непериодическую последовательность одиночных радиоимпульсов различной формы. Они создаются атмосферными и промышленными источниками помех, а в отдельных случаях и другими системами связи.

Синусоидальными помехами являются помехи, сосредоточенные по спектру, ширина спектра которых мала по сравнению с полосой пропускания приёмника. Источниками такого рода помех являются: станции преднамеренных помех, генераторы высокой частоты, радиостанции эталонных частот и т.д. К синусоидальным можно отнести и комбинационные помехи внутри самого приёмника.

Хаотические изменения коэффициента передачи физической среды (линии связи), в которой распространяется сигнал, обычно приводят к его искажениям. Обычно такие искажения называют *мультипликативной помехой*. При этом принимаемый сигнал представляет произведение передаваемого сигнала $s(t)$ и помехи $\mu(t)$, т.е.

$$\xi(t) = \mu(t)s(t).$$

В общем случае на сигнал воздействуют и мультипликативные, и аддитивные помехи.

Основной задачей *приёмного устройства* является выделение передаваемого сообщения из принятого сигнала (наблюдения). В общем случае это достигается выполнением операции *демодуляции*, а при приёме дискретных сообщений и *декодирования*. Устройства, выполняющие эти операции, называются соответственно *демодулятором* и *декодером*.

Операция демодуляции заключается в преобразовании принятого модулированного сигнала, искажённого помехами, в модулирующий сигнал. Операция декодирования является обратной операции кодирования.

В системах передачи непрерывных сообщений (при аналоговой модуляции) сигнал на выходе демодулятора должен совпадать с первичным электрическим сигналом, отображающим сообщение. В системах передачи дискретных сообщений для восстановления исходного сообщения реализуется ещё и операция декодирования.

Совокупность кодирующего и декодирующего устройства образует подсистему, называемую *кодеком*. Совокупность модулятора и демодулятора образует подсистему, называемую *модемом*.

Получатель сообщения – это устройство или человек, для которого предназначено сообщение.

Совокупность технических средств передачи информации и линии связи называется *каналом связи*. Конкретный состав канала связи определяется кругом решаемых задач.

1.2. ЦИФРОВЫЕ СИГНАЛЫ

Амплитудная манипуляция (АМн). Аналитическое выражение АМн сигнала для любого момента времени t имеет вид

$$s_{\text{АМн}}(t, \theta) = A_0 \theta(t) \cos(\omega_0 t + \varphi), \qquad (1)$$

где A_0, ω_0 и φ – амплитуда, циклическая несущая частота и начальная фаза АМн радиосигнала; $\theta(t)$ – дискретный информационный параметр сигнала. Реализация АМн сигнала приведена на рис. 3.

Спектральная плотность АМн сигнала имеет как непрерывную, так и дискретную составляющую на частоте несущего колебания ω_0. Непрерывная составляющая представляет собой спектральную плотность передаваемого цифрового сигнала $\theta(t)$, перенесённую в область несущей частоты. Следует отметить, что дискретная составляющая спектральной плотности имеет место только при постоянной начальной фазе сигнала φ.

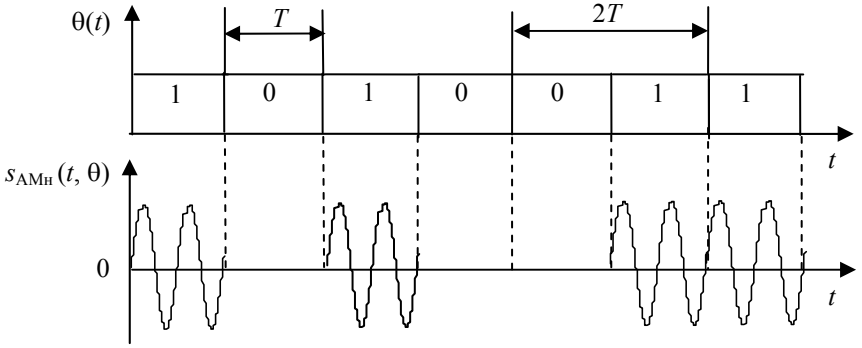


Рис. 3. Реализация АМн радиосигнала

На практике это условие не выполняется, так как в результате различных дестабилизирующих факторов начальная фаза сигнала случайным образом изменяется во времени, т.е. является случайным процессом $\varphi(t)$ и равномерно распределена в интервале $[-\pi, \pi]$. Наличие таких фазовых флюктуаций приводит к «размыванию» дискретной составляющей. Эта особенность характерна и для других видов манипуляции. На рисунке 4 приведена спектральная плотность АМн радиосигнала.

Средняя мощность АМн радиосигнала равна $P_{АМн} = A_0^2/4$. Эта мощность поровну распределяется между непрерывной и дискретной составляющими спектральной плотности. Следовательно, в АМн радиосигнале на долю непрерывной составляющей, обусловленной передачей полезной информации, приходится лишь половина мощности излучаемой передатчиком.

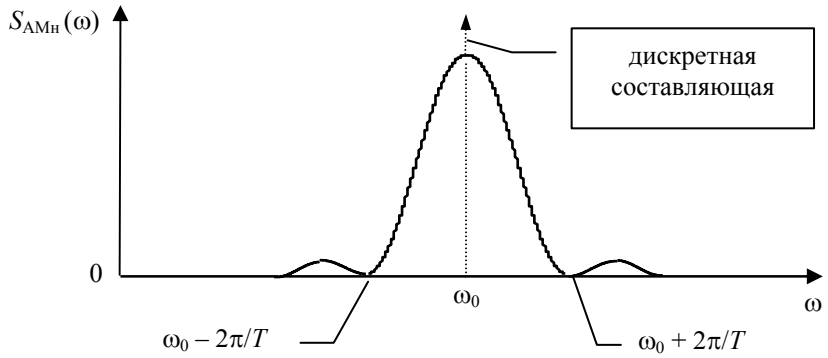


Рис. 4. Спектральная плотность АМн радиосигнала со случайной, равномерно распределённой в интервале [–π; π] начальной фазой φ

Частотная манипуляция (ЧМн). Аналитическое выражение сигнала имеет вид:

$$s_{\text{ЧМн}}(t,\theta) = [1 - \theta(t)]\, A_0 \cos(\omega_1 t + \varphi_1) + \theta(t)\, A_0 \cos(\omega_2 t + \varphi_2), \qquad (2)$$

где ω_1 и ω_2 – циклические частоты, соответствующие информационным посылкам сигнала $\theta(t)$, фазы φ_1 и φ_2 могут отличаться друг от друга.

При рассмотрении ЧМн радиосигналов используются следующие основные понятия:

– *средняя частота передачи*

$$f_0 = (f_1 + f_2)/2,$$

где $f_i = \omega_i / 2\pi$, $i = 0, 1, 2$;

– *частотный разнос* (*сдвиг*)

$$f_{\text{р}} = |f_2 - f_1|;$$

– *девиация частоты*

$$f_{\text{д}} = f_{\text{р}} / 2;$$

– *индекс частотной манипуляции*

$$\beta = 2f_{\text{д}} / V_{\text{м}},$$

где $V_{\text{м}} = 1/T$ – скорость манипуляции.

Сигналы с частотной манипуляцией подразделяются на сигналы с разрывом фазы и без разрыва фазы.

Реализация во времени ЧМн радиосигнала с разрывом фазы и без разрыва фазы приведены на рис. 5.

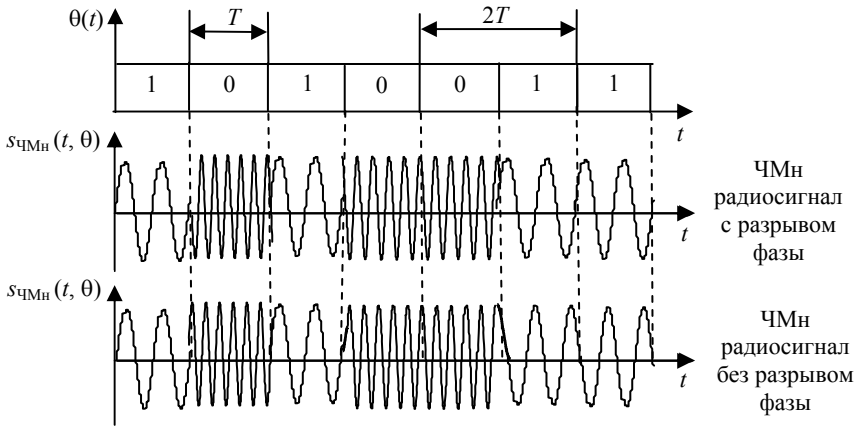


Рис. 5. Реализации ЧМн радиосигнала с разрывом и без разрыва фазы

Спектральные плотности ЧМн радиосигналов с разрывом и без разрыва фазы приведены на рис. 6, *а* и *б* соответственно. Из рисунка видно, что спектральная плотность ЧМн сигнала без разрыва фазы занимает практически в два раза меньшую полосу частот и имеет значительно меньший уровень боковых лепестков по сравнению с сигналом, в котором имеет место разрыв фазы. Это представляется важным с точки зрения экономии частотного ресурса радиолиний и лучшей электромагнитной совместимости различных систем связи. Такое сужение спектра объясняется отсутствием скачков фазы при манипуляции частоты.

Здесь $\omega_0 = 2\pi f_0 = (\omega_1 + \omega_2)/2$ – средняя (центральная) круговая частота передачи, а $\beta = 3$ и 0,5 – значения индекса частотной манипуляции.

Из рисунков 6, *а* и *б* видно, что дискретная составляющая, обусловленная наличием скачков фазы, присутствует только в спектре ЧМн сигнала с разрывом фазы.

Средняя мощность ЧМн сигнала с разрывом фазы как и в случае с АМн сигналом, распределена поровну между составляющей, несущей полезную информацию и дискретной составляющими спектральной плотности.

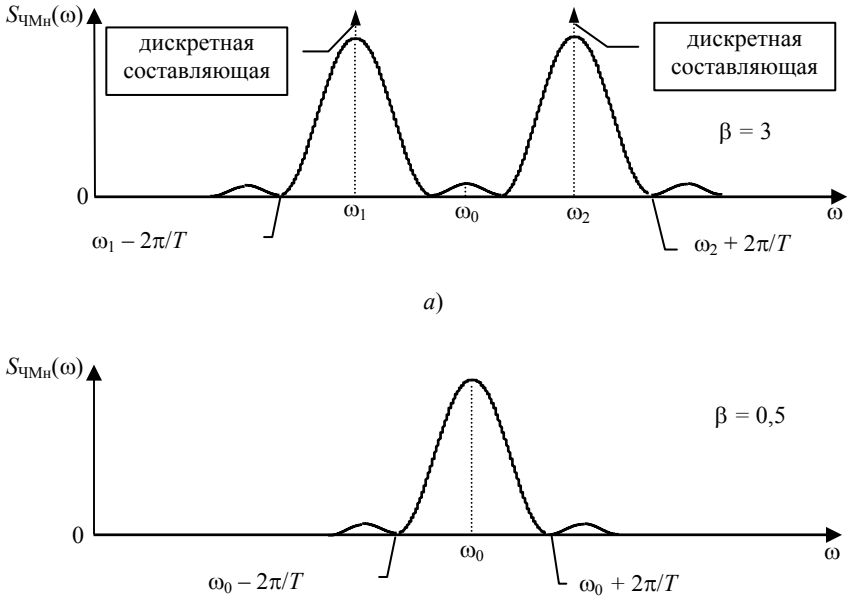


Рис. 6. Спектральные плотности ЧМн сигналов с разрывом и без разрыва фазы

Однако, при одинаковой амплитуде сигналов средняя мощность ЧМн сигнала в два раза превышает мощность АМн сигнала ($P_{\text{ЧМн}} = 2P_{\text{АМн}} = \frac{A_0^2}{2}$), поскольку ЧМн радиосигнал является сигналом с активной паузой и при передаче логического нуля также в линию связи выдаётся энергия. Для ЧМн радиосигнала без разрыва фазы энергетические соотношения из-за отсутствия дискретной составляющей спектра становятся ещё более предпочтительными.

Фазовая манипуляция (ФМн). При передаче бинарных ФМн сигналов, как правило, применяется манипуляция на π. Аналитическое выражение ФМн сигнала в этом случае имеет вид

$$s_{\text{ФМн}}(t,\theta) = A_0 \cos[\omega_0 t + \theta(t)\pi + \varphi]. \qquad (3)$$

Реализация ФМн радиосигнала представлена на рис. 7.

Спектральная плотность ФМн радиосигнала приведена на рис. 8.

Спектральная плотность ФМн радиосигнала (рис. 8) включает в себя только непрерывную составляющую. Дискретная составляющая, содержащая информацию о фазе, отсутствует. Поскольку дискретная составляющая в спектральной плотности ФМн радиосигнала отсутствует, то вся мощность передатчика расходуется на передачу полезной информации. Учитывая, что $s(t) \neq 0$ при $\theta = 0$ и ФМн сигнал является сигналом с активной паузой, его средняя мощность, как и при ЧМн, в два раза превышает среднюю мощность АМн сигнала $P_{\text{ФМн}} = 2P_{\text{АМн}} = \frac{A_0^2}{2}$.

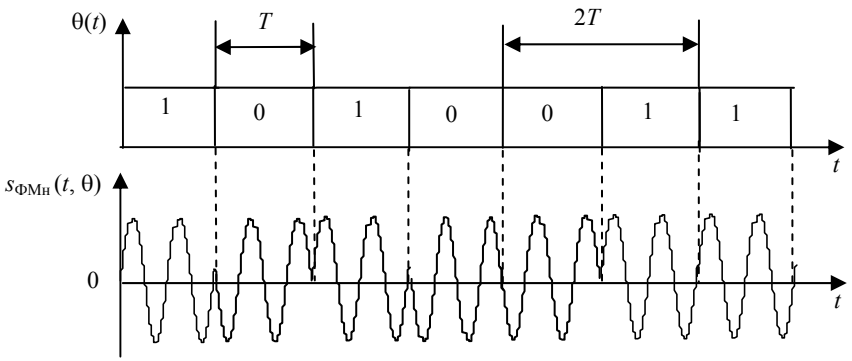


Рис. 7. Реализация ФМн радиосигнала

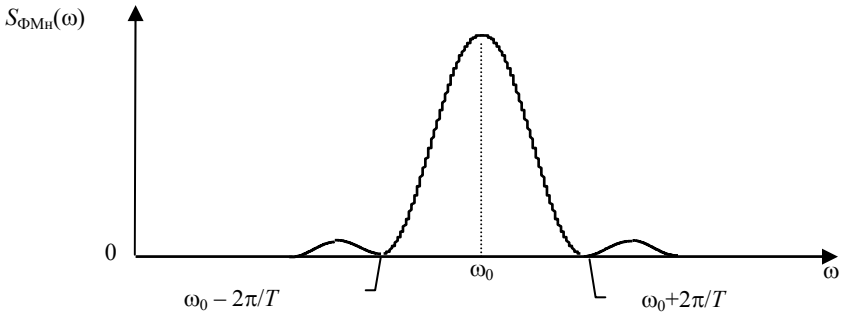


Рис. 8. Спектральная плотность ФМн радиосигнала

Различение детерминированных сигналов. Получим алгоритм различения двух детерминированных сигналов в условиях аддитивной флуктуационной помехи. В этом случае наблюдение на входе приёмного устройства имеет вид

$$\xi(t) = s[\theta(t), \quad \lambda(t)] + n(t), \tag{4}$$

где $n(t)$ – белый гауссовский шум (БГШ) с нулевым математическим ожиданием и односторонней спектральной плотностью с интенсивностью N_0; $\theta(t)$ – информационный параметр; $\lambda(t)$ – вектор сопутствующих параметров, описывающий амплитудные замирания, случайные колебания фазы и т.д. То, что сигнал детерминирован означает, что вектор $\lambda(t)$ на приёмной стороне точно известен. Будем считать, что при приёме выполняется точная тактовая синхронизация, т.е. моменты изменения цифрового сигнала точно известны. Неизвестный параметр θ принимает одно из двух значений $\theta = 1$ или $\theta = 0$, что позволяет представить сигнал следующим образом:

$$s(t, \theta) = (1 - \theta)s_1(t) + \theta\, s_2(t).$$

В цифровых системах связи критерием помехоустойчивости наиболее часто выступает полная вероятность ошибки P_e. Поэтому различение сигналов $s_1(t)$ и $s_2(t)$ целесообразно выполнять в соответствии с критерием идеального наблюдателя, минимизирующим P_e:

$$\frac{P_{\text{ps}}(\theta = 1)}{P_{\text{ps}}(\theta = 0)} \underset{\theta=0}{\overset{\theta}{\gtrless}} 1, \tag{5}$$

где $P_{\text{ps}}(\theta = i)$ – апостериорная вероятность, содержащая всю информацию о переданном символе $\theta = i$ ($i = 1, 2$), заключённом в наблюдении $\xi(t)$

на интервале времени от 0 до T. После преобразований алгоритм приёма примет вид

$$\int_0^T \xi(t)\, s_2(t)\, dt - \int_0^T \xi(t)\, s_1(t)\, dt \underset{\hat{\theta}=0}{\overset{\hat{\theta}=1}{\lessgtr}} \frac{E_2 - E_1}{2} = h. \qquad (6)$$

Формула полной вероятности ошибки может быть представлена в виде

$$P_e = P(0)P(1|0) + P(1)P(0|1), \qquad (7)$$

где $P(0) = P_{\text{pr}}(\theta = 0) = P(1) = P(\theta = 1) = 0,5$ – априорные вероятности передачи $\theta = 0$ и $\theta = 1$; $P(1/0)$ – условная вероятность того, что при переданном $\theta = 0$ вынесено решение о значении оценки $\hat{\theta} = 1$; $P(1|0)$ – условная вероятность того, что при переданном $\theta = 1$ вынесено решение о значении оценки $\hat{\theta} = 0$. Полная вероятность ошибки (21) для сигналов, в которых появление $\theta = 0$ и $\theta = 1$ равновероятны, может быть определена как

$$P_e = P(1|0) = P(0|1). \qquad (8)$$

Таким образом, для определения P_e достаточно определить выражение условной вероятности $P(1|0)$ или $P(1|0)$ для радиосигналов с различными видами манипуляции.

В общем случае полная вероятность ошибки может быть записана как

$$P_e = \int_{-\infty}^{-0,5\rho} p(\gamma)\, d\gamma = \Phi\left(\frac{-0,5\rho}{\sqrt{D_\gamma}}\right) = 1 - \Phi\left(\frac{-0,5\rho}{\sqrt{D_\gamma}}\right) = 1 - \Phi\left(\sqrt{\frac{\rho}{2N_0}}\right), \qquad (9)$$

где $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-0,5x} dx$ – интеграл вероятностей; $\rho = \int_0^T \left[s_1(t) - s_2(t)\right]^2 dt =$ $= E_1 + E_2 - 2R$, $E_{1(2)}$ – энергии сигнальных посылок $s_1(t)$ и $s_1(t)$, а R – коэффициент взаимной корреляции сигналов $s_1(t)$ и $s_1(t)$; $E_i = \int_0^T s_i^2(t)\, dt = E$, N_0 – интенсивность спектральной плотности БГШ.

С учётом значения ρ для различных видов манипуляции, полная вероятность ошибки будет иметь вид:

– для радиосигналов с АМн

$$P_e = 1 - \Phi\left(\sqrt{\frac{E}{2N_0}}\right); \qquad (10)$$

– для радиосигналов с ЧМн

$$P_e = 1 - \Phi\left(\sqrt{\frac{E}{N_0}}\right); \qquad (11)$$

– для радиосигналов с ФМн

$$P_e = 1 - \Phi\left(\sqrt{\frac{2E}{N_0}}\right). \qquad (12)$$

Здесь $E = E_1 = E_2$ – энергии сигналов. Значения интеграла вероятностей приведены в табл. 1.

1. Значения интеграла вероятностей $\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-0{,}5x^2} dx$

z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$
0,00	0,5000				
0,10	0,5398	0,50	0,6915	0,90	0,8159
0,11	0,5438	0,51	0,6950	0,91	0,8186
0,12	0,5478	0,52	0,6985	0,92	0,8212
0,13	0,5517	0,53	0,7019	0,93	0,8238
0,14	0,5557	0,54	0,7054	0,94	0,8264
0,15	0,5596	0,55	0,7088	0,95	0,8289
0,16	0,5636	0,56	0,7123	0,96	0,8315
0,17	0,5675	0,57	0,7157	0,97	0,8340
0,18	0,5714	0,58	0,7190	0,98	0,8365
0,19	0,5753	0,59	0,7224	0,99	0,8389
0,20	0,5793	0,60	0,7257	1,00	0,8413
0,21	0,5832	0,61	0,7291	1,01	0,8437
0,22	0,5871	0,62	0,7324	1,02	0,8461
0,23	0,5910	0,63	0,7357	1,03	0,8485

Продолжение табл. 1

z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$
0,24	0,5948	0,64	0,7389	1,04	0,8508
0,25	0,5987	0,65	0,7422	1,05	0,8531
0,26	0,6026	0,66	0,7454	1,06	0,8554
0,27	0,6064	0,67	0,7486	1,07	0,8577
0,28	0,6103	0,68	0,7517	1,08	0,8599
0,29	0,6141	0,69	0,7549	1,09	0,8621
0,30	0,6179	0,70	0,7580	1,10	0,8643
0,31	0,6217	0,71	0,7611	1,11	0,8665
0,32	0,6255	0,72	0,7642	1,12	0,8686
0,33	0,6293	0,73	0,7673	1,13	0,8708
0,34	0,6331	0,74	0,7703	1,14	0,8729
0,35	0,6368	0,75	0,7734	1,15	0,8749
0,36	0,6406	0,76	0,7764	1,16	0,8770
0,37	0,6443	0,77	0,7794	1,17	0,8790
0,38	0,6480	0,78	0,7823	1,18	0,8810
0,39	0,6517	0,79	0,7852	1,19	0,8830
0,40	0,6554	0,80	0,7881	1,20	0,8849
0,41	0,6591	0,81	0,7910	1,21	0,8869
0,42	0,6628	0,82	0,7939	1,22	0,8888
0,43	0,6664	0,83	0,7967	1,23	0,8907
0,44	0,6700	0,84	0,7995	1,24	0,8925
0,45	0,6736	0,85	0,8023	1,25	0,8944
0,46	0,6772	0,86	0,8051	1,26	0,8962
0,47	0,6808	0,87	0,8078	1,27	0,8980
0,48	0,6844	0,88	0,8106	1,28	0,8997
0,49	0,6879	0,89	0,8133	1,29	0,9015
1,30	0,9032	1,60	0,9452	1,90	0,9713
1,31	0,9049	1,61	0,9463	1,91	0,9719
1,32	0,9066	1,62	0,9474	1,92	0,9726

Продолжение табл. 1

z	$\Phi(z)$	z	$\Phi(z)$	z	$\Phi(z)$
1,33	0,9082	1,63	0,9484	1,93	0,9732
1,34	0,9099	1,64	0,9495	1,94	0,9738
1,35	0,9115	1,65	0,9505	1,95	0,9744
1,36	0,9131	1,66	0,9515	1,96	0,9750
1,37	0,9147	1,67	0,9525	1,97	0,9756
1,38	0,9162	1,68	0,9535	1,98	0,9761
1,39	0,9177	1,69	0,9545	1,99	0,9767
1,40	0,9192	1,70	0,9554	2,00	0,9772
1,41	0,9207	1,71	0,9564	2,10	0,9821
1,42	0,9222	1,72	0,9573	2,20	0,9861
1,43	0,9236	1,73	0,9582	2,30	0,9893
1,44	0,9251	1,74	0,9591	2,40	0,9918
1,45	0,9265	1,75	0,9599	2,50	0,9938
1,46	0,9279	1,76	0,9608	2,60	0,9953
1,47	0,9292	1,77	0,9616	2,70	0,9965
1,48	0,9306	1,78	0,9625	2,80	0,9974
1,49	0,9319	1,79	0,9633	2,90	0,9981
1,50	0,9332	1,80	0,9641	3,0	0,9986
1,51	0,9345	1,81	0,9649	3,10	0,9990
1,52	0,9357	1,82	0,9556	3,20	0,9993
1,53	0,9370	1,83	0,9664	3,30	0,9995
1,54	0,9382	1,84	0,9671	3,40	0,9997
1,55	0,9394	1,85	0,9678	3,50	0,9998
1,56	0,9406	1,86	0,9686	3,60	0,9998
1,57	0,9418	1,87	0,9693	3,70	0,9999
1,58	0,9429	1,88	0,9699	3,80	0,9999
1,59	0,9441	1,89	0,9706	3,90	1,0000

Контрольные вопросы

1. Дать определение случайного процесса? Назвать основные способы описания случайных процессов (СП).
2. Дать определение апостериорной плотности вероятности.
3. Дать определение корреляционной функцией стационарного СП и что она характеризует? Перечислите её основные свойства.
4. Что характеризует собой интервал корреляции СП?
5. Что называется спектральной плотностью стационарного СП, и что она характеризует?
6. Опишите свойства и характеристики белого шума.
7. Пояснить понятие информационного параметра сообщения.
8. Изобразить обобщённую структурную схему системы связи, пояснить назначение составляющих схемы.
9. Дать определение канала связи, перечислить основные виды каналов связи и основные виды помех в нём.
10. Какие виды манипуляции используются для формирования цифровых сигналов?
11. Пояснить понятие «детерминированный сигнал».
12. Пояснить алгоритм оптимального различения двух детерминированных цифровых сигналов.
13. Что характеризует коэффициент взаимной корреляции сигналов R?
14. Пояснить критерий оценки помехоустойчивости приёма цифровых сигналов.
15. Какой из видов манипуляции позволяет получить наибольшую потенциальную помехоустойчивость приёма?
16. Чем отличаются спектры сигналов с частотной манипуляцией и частотной манипуляцией и минимальным частотным сдвигом?

Задачи для самостоятельного решения

Задача 1. Найти энергии элементов цифровых АМн, ЧМн и ФМн сигналов, если амплитуда A_0 и длительность T для всех сигналов одинаковы и составляют 2 мВ и 3 мкс соответственно.

Задача 2. Полоса частот Δf, занимаемая АМН, ФМн и ЧМн с разрывом фазы сигналами, составляют 10 МГц. Частотный разнос ЧМн сигнала $f_р$ составляет 5МГц. Найти длительности элементов T и скорости передачи элементов V_T каждого из сигналов.

Задача 3. Радиосигнал с АМн поступает на оптимальный приёмник на фоне белого гауссовского шума (БГШ). Известно, что априорные веро-

ятности появления сигнальных посылок «0» и «1» равны. Определите полную вероятность ошибки P_e, если уровень спектральной плотности БГШ $N_0 = 2$ мВ^2/Гц, а энергия сигнальной посылки $E = 16$ мВ^{2}с.

Задача 4. Энергия сигнальной посылки цифрового сигнала с частотной манипуляцией (ЧМн) составляет $E = 5$ мВ^{2}с. Определите полную вероятность ошибки при приёме этого сигнала на фоне белого гауссовского шума (БГШ) со спектральной плотностью $N_0 = 1{,}25$ мВ^2/Гц.

Задача 5. Определите значение полной вероятности ошибки при разрешении сигнальных посылок ФМн радиосигнала, если для АМн радиосигнала её значение определяется как $P_e = 10^{-6}$. Оба радиосигнала имеют одинаковую энергию, длительность и амплитуду.

Задача 6. Отношение сигнал-шум при приёме цифрового сигнала на фоне белого гауссовского шума (БГШ) составляет $\sqrt{2E/N_0} = 2{,}0$. Определите полные вероятности ошибки при приёме сигналов с амплитудной, частотной и фазовой манипуляцией (АМн, ЧМн и ФМн).

Задача 7. Энергия сигнальной посылки двоичного (бинарного) цифрового радиосигнала имеет значение $E = 5$ мВ^{2}с. Сигнальные посылки равновероятны, равны по длительности и амплитуде. Определите значения полной вероятности ошибки при разрешении этих посылок для радиосигналов с АМн, ЧМн, ФМн, если приём осуществляется на фоне БГШ с уровнем спектральной плотности $N_0 = 1{,}25$ мВ^2/Гц.

Задача 8. Построить (например) в среде Matcad график зависимости вероятности ошибки P_e от отношения сигнал–шум $\sqrt{2E/N_0}$ для АМн, ЧМн и ФМн сигналов.

Задача 9. Отношения мощности сигнала к мощности помехи в точке приёма $P_с/P_п = 2, 3, 5$. Найти вероятность ошибки P_e на бит для АМн, ЧМн и ФМн сигналов.

Задача 10. Найти, как изменится отношение мощности ЧМн сигнала к мощности помехи $P_с/P_п$, если разнос частот $f_р$ увеличится в два раза. Пояснить.

Практическое занятие 4

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ В МОБИЛЬНЫХ СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

Цель: совершенствование теоретических знаний по процессам кодирования и декодирования в циклических кодах

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: теоретический материал по теме занятия; методику построения кодовых комбинаций циклического кода, а также обнаружения и исправления ошибок в процессе декодирования кодовых комбинаций при использовании циклического кодирования;

– у*меть*: строить кодовые комбинации циклических кодов; обнаруживать и исправлять ошибки в кодовых комбинациях циклических кодов при их декодировании.

Проведение практического занятия включает три этапа:

1. Предварительная подготовка к занятию – проработка теоретического материала студентами на занятии и в ходе самостоятельной работы.
2. Основная часть – письменный опрос и решение задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 32 – 75].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. ОБЩИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

На рисунке 1 представлена структурная схема системы передачи информации. Источник выдаёт последовательность знаков дискретного сообщения, которое должно быть передано получателю по каналу связи. Знаками сообщения могут быть, например, буквы русского алфавита, двоичные символы и т.п.

Сообщение поступает на вход кодера, который осуществляет кодирование. Под *кодированием* понимают преобразование знаков дискретного сообщения в кодовые комбинации (кодовые слова), осуществляемое по определённому правилу[1]. Множество всех кодовых комбинаций, возможных при данном правиле кодирования, называется *кодом.*

[1] В широком смысле под кодированием понимают преобразование любого сообщения (дискретного и непрерывного) в последовательность кодовых символов.

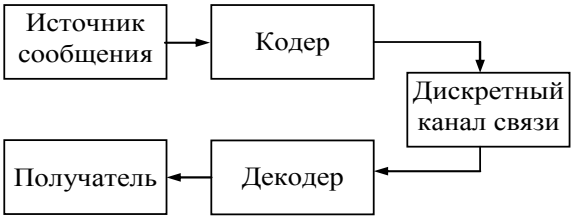


Рис. 1. Структурная схема системы передачи информации

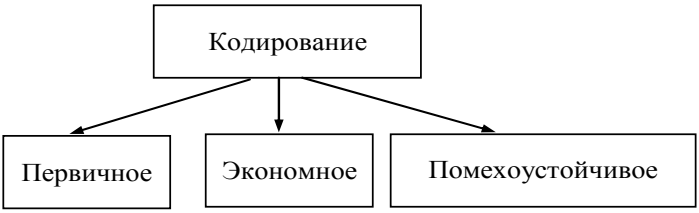


Рис. 2. Классификация методов кодирования

Используются три метода кодирования: первичное, экономное и помехоустойчивое (рис. 2).

Первичное кодирование позволяет представить множество дискретных величин кодовыми комбинациями. Например, для передачи по каналу связи текста на русском языке 32 буквы русского алфавита преобразуются в двоичные числа. В рассматриваемом случае каждой букве можно поставить в соответствие пятизначное двоичное число (количество различных пятизначных двоичных чисел равно $N = 2^5 = 32$). Если все возможные кодовые комбинации используются для передачи сообщений, то код называется *безызбыточным* или *первичным***.** Примером таких кодов являются международный код обмена информацией КОИ-8 и др.

Экономное кодирование позволяет сократить избыточность передаваемых сообщений (сжать данные). Такой метод кодирования учитывает статистические характеристики источника сообщений и даёт возможность представить знаки сообщения в среднем меньшим числом кодовых символов. Например, значительно сократить избыточность передаваемых сообщений можно, если наиболее часто встречающимся символам сообщений ставить в соответствие более короткие кодовые комбинации, а менее вероятным символам – более длинные. Операцию сокращения избыточности источника сообщений называют *экономным кодированием*. Экономное кодирование широко используется в цифровых системах передачи инфор-

мации для сжатия речевых, факсимильных и телевизионных сообщений, которые обладают большой избыточностью. Первичное и экономное кодирование называют также *кодированием источника.*

Помехоустойчивое кодирование позволяет обнаруживать и исправлять ошибки, возникающие при передаче сообщений по каналу связи. Помехоустойчивое кодирование осуществляется за счёт введения в состав передаваемых кодовых символов достаточно большого объёма избыточной информации (например, проверочных символов). Операцию введения избыточности для повышения помехоустойчивости канала связи называют *канальным кодированием*. Ошибка при приёме одной кодовой комбинации состоит в том, что вследствие действия помехи в решающем устройстве приёмника нуль заменяется единицей или, наоборот, единица – нулём.

Классификация помехоустойчивых кодов по некоторым признакам приведена на рис. 3. По способу кодирования помехоустойчивые коды обычно разбивают на два класса: блочные и непрерывные.

Блочное кодирование состоит в том, что последовательность символов источника сообщений (последовательность нулей и единиц) разделяется на блоки[2], которые обычно называют кодовыми комбинациями. Блоки, содержащие k символов каждый, по определённому закону преобразуются кодером в n-символьные блоки, причем $n > k$.

Непрерывные коды характеризуются тем, что кодирование и декодирование информационной последовательности символов осуществляется без её разбиения на блоки.

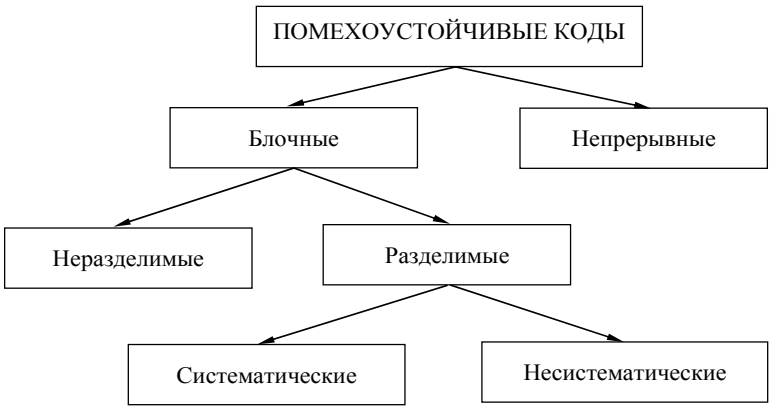


Рис. 3. Классификация помехоустойчивых кодов

[2] На практике количество символов в блоке может лежать в пределах от трёх до нескольких сотен.

Каждый символ выходной последовательности получается как результат некоторых операций над символами входной последовательности. Кодирование и декодирование непрерывных кодов носит непрерывный характер. При этом результат декодирования предыдущих или последующих символов может повлиять на декодирование текущего символа. Среди непрерывных кодов наиболее часто применяют *свёрточные коды*.

Основными характеристиками помехоустойчивых кодов являются: длина кода n, его основание m, общее число кодовых комбинаций N, число разрешённых кодовых комбинаций $N_р$, избыточность кода $K_и$ и минимальное кодовое расстояние d_{min}.

Длина кода[3] n – это число символов в кодовой комбинации. Например, комбинация 11010 состоит из пяти символов, следовательно, $n = 5$. Если все кодовые комбинации содержат одинаковое число символов, то код называется *равномерным*. В неравномерных кодах длина кодовых комбинаций может быть разной.

Основание кода m – это число различных символов в коде. Для двоичных кодов символами являются 1 и 0, поэтому $m = 2$.

Число кодовых комбинаций для равномерного кода равно $N = m^n$. Например, для равномерного двоичного кода, имеющего длину $n = 6$, число различных кодовых комбинаций равно $N = 2^6 = 64$.

Число разрешённых кодовых комбинаций $N_р$ – это количество кодовых комбинаций кода, используемых для передачи сообщений. Для помехоустойчивых кодов $N_р < N$. Оставшиеся кодовые комбинации $N - N_р$ называют *запрещёнными*. Если $N_р = N$, то код является безызбыточным. Для разделимых кодов $N_р = 2^k$.

Избыточность кода $K_и$ в общем случае определяется выражением

$$K_и = 1 - \frac{\log_2 N_р}{\log_2 N} \qquad (1)$$

и показывает, какая доля длины кодовой комбинации не используется для передачи информации, а используется для повышения помехоустойчивости кода. Для разделимых кодов

$$K_и = 1 - \frac{k}{n} = \frac{r}{n}, \qquad (2)$$

где величина k/n называется *относительной скоростью кода*.

[3] Под длиной кода также понимают значность, или разрядность, кода.

Кодовое расстояние $d(A, B)$ – это число позиций, в которых две кодовые комбинации A и B отличаются друг от друга. Например, если A = 01101, B = 10111, то $d(A, B) = 3$. Кодовое расстояние между комбинациями A и B может быть найдено в результате сложения по модулю 2 одноименных разрядов комбинаций, а именно

$$d(A, B) = \sum_{i=1}^{n} a_i \oplus b_i , \qquad (3)$$

где a_i и b_i – i-е разряды кодовых комбинаций A и B; символ $\oplus$ обозначает сложение по модулю 2. Например, чтобы получить кодовое расстояние между комбинациями 1101011 и 0111101, достаточно подсчитать число единиц в сумме этих комбинаций по модулю 2:

$$\begin{array}{ll} & 1101011 \\ \oplus & \underline{0111101} \\ & 1010110 \Rightarrow \ d(A, B) = 4. \end{array}$$

Кодовое расстояние между различными комбинациями конкретного кода может быть различным. Так, для первичных кодов это расстояние для различных пар кодовых комбинаций может принимать значения от единицы до величины длины кода.

Минимальное кодовое расстояние $d_{\min}$ – это минимальное расстояние между разрешёнными кодовыми комбинациями данного кода. Минимальное кодовое расстояние является основной характеристикой корректирующей способности кода. В первичных (безызбыточных) кодах все комбинации являются разрешёнными, поэтому минимальное кодовое расстояние для них равно единице ($d_{\min} = 1$). Такие коды не способны обнаруживать и исправлять ошибки. Для того чтобы код обладал корректирующими способностями, его минимальное кодовое расстояние должно быть не менее двух ($d_{\min} \geq 2$).

Для обнаружения всех ошибок кратностью[4] s и менее минимальное кодовое расстояние должно удовлетворять условию

$$d_{\min} \geq s + 1. \qquad (4)$$

Если код используется для исправления ошибок кратности не более t, то минимальное кодовое расстояние должно иметь значение

$$d_{\min} \geq 2t + 1. \qquad (5)$$

Например, из (4), (5) следует, что для обнаружения однократных ошибок ($s = 1$) требуется код с $d_{\min} = 2$, а для того, чтобы исправить такие ошибки, требуется код с кодовым расстоянием $d_{\min} = 3$.

[4] Напомним, что *кратность ошибки* – это число позиций кодовой комбинации, в которых под воздействием помех одни символы оказались замененными другими (нули – единицами, единицы – нулями).

Для обнаружения s ошибок и исправления t ошибок должно выполняться условие

$$d_{\min} \geq s + t + 1. \quad (6)$$

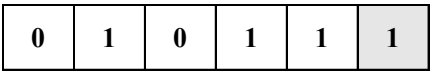
Таким образом, задача построения кода с заданной корректирующей способностью сводится к обеспечению необходимого кодового расстояния. Увеличение $d_{\min}$ приводит к росту избыточности кода. При этом желательно, чтобы число проверочных символов r было минимальным. В настоящее время известен ряд верхних и нижних границ, которые устанавливают связь между кодовым расстоянием и числом проверочных символов.

1.2. ПРОСТЕЙШИЕ ПОМЕХОУСТОЙЧИВЫЕ КОДЫ

Одним из простейших блочных неразделимых кодов является код с постоянным весом. Весом кодовой комбинации называют число содержащихся в ней единиц. Код с постоянным весом способен обнаруживать ошибки большей кратности (двойные, тройные и т.д.), при которых нарушается весовое соотношение единиц и нулей. Код не обнаруживает только ошибки, при которых число единиц, перешедших в нули, равно числу нулей, перешедших в единицы. Код с проверкой на чётность является наиболее простым блочным разделимым кодом, содержащим всего лишь один проверочный символ. Проверочный символ выбирается равным единице или нулю, таким образом, чтобы число единиц во вновь образованной $n = k + 1$ – элементной кодовой комбинации было чётным (рис. 4).

Формирование проверочных символов кодовых комбинаций кода с чётным числом единиц осуществляется путём сложения по модулю 2 информационных символов. Например, к первичной комбинации 01011 следует добавить справа единицу, так как $0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1$, следовательно, разрешённая кодовая комбинация запишется так: 010111.

Легко убедиться, что проверка на чётность позволяет выявить одиночную ошибку, поскольку она изменяет чётность. Кроме однократных ошибок этот код позволяет обнаруживать все ошибки нечётной кратности.



Информационные символы ($k = 5$) Проверочный символ ($r = 1$)

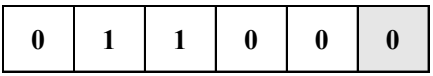


Рис. 4. Примеры разрешённых кодовых комбинаций кода с проверкой на чётность (6,5)

Ошибки чётной кратности код не обнаруживает. Например, если в кодовой комбинации будут искажены два символа, то чётность не изменится и ошибка не будет обнаружена.

Код Бергера или код с контрольным суммированием является несистематическим кодом, у которого проверочные символы представляют собой двоичную запись числа единиц в последовательности информационных символов. Например, информационная последовательность 01011 содержит три единицы. Десятичному числу 3 соответствует двоичная запись 011, поэтому разрешённая кодовая комбинация кода Бергера примет вид: 01011011. Примеры разрешённых кодовых комбинаций данного кода приведены на рис. 5, где проверочные символы обозначены затемнением.

Минимальное кодовое расстояние кода Бергера равно двум. Коды Бергера обнаруживают все одиночные ошибки и некоторую часть многократных ошибок. Эти коды обладают примерно такой же корректирующей способностью, что и код МТК-3, но имеют важное преимущество – они являются разделимыми, что упрощает реализацию кодирующих и декодирующих устройств.

Код с повторением является простейшим кодом, в котором информационные символы повторяются несколько раз. При этом может дублироваться ν раз каждый символ или каждая информационная кодовая комбинация. Например, пятибитной информационной кодовой комбинации 11010 соответствует разрешённая комбинация кода с повторением при $\nu = 2$: 1111001100 или 1101011010. Таким образом, проверочными символами данного кода являются информационные символы. Код с повторением является разделимым систематическим кодом.

Основные характеристики кода зависят от числа повторений и равны

$$n = \nu k\ ,\ K_{\text{и}} = \frac{\nu - 1}{\nu}\ ,\ d = \nu\ .$$

Код с повторением способен не только обнаруживать ошибки, но и исправлять их. Однако для исправления даже одиночных ошибок требуется трёхкратное повторение, т.е. трёхкратное увеличение длины кода.

0	**1**	**0**	**1**	**1**	**0**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**0**	**1**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**

Рис. 5. Примеры разрешённых кодовых комбинаций кода Бергера

Это слишком большая цена. Поэтому код с повторением хотя и применяется, но является низкоскоростным, поскольку имеет высокую избыточность. Например, при двукратном повторении минимальное кодовое расстояние равно двум ($d_{\min} = 2$), а избыточность $K_и = 0{,}5$, которая существенно превышает избыточность рассмотренных ранее кодов.

Код Бауэра является разновидностью кода с двукратным повторением, обеспечивающим минимальное кодовое расстояние, равное четырём ($d_{\min} = 4$). При образовании данного кода кодовые комбинации с чётным числом единиц повторяются в неизменном виде (рис. 6, *а*), а кодовые комбинации с нечётным числом единиц – в инвертированном (рис. 6, *б*).

Код Бауэра называют также инверсным кодом. При небольших длинах кодовых комбинаций ($n = 10…14$) инверсный код по своим корректирующим способностям не уступает более сложным кодам с такой же величиной избыточности.

Рекуррентный код (цепной код) – это непрерывный код, у которого проверочные символы чередуются с информационными по всей длине кодовой последовательности. Формирование проверочных символов в таких кодах осуществляется по рекуррентным правилам. В простейшем случае проверочные символы формируются путём суммирования по модулю 2 двух соседних информационных символов:

$$b_i = a_i \oplus a_{i+1}, \tag{7}$$

где b_i – i-й проверочный символ; a_i – i-й информационный символ, принимающий значение 0 или 1. Далее проверочные символы, сформированные по правилу (7), перемежаются с информационными, образуя непрерывную последовательность (рис. 7).

1	1	0	1	1	1	1	0	1	1

а)

1	0	0	1	1	0	1	1	0	0

б)

Рис. 6. Примеры разрешенных кодовых комбинаций кода Бауэра

a_1	b_1	a_2	b_2	…	b_{i-1}	a_i	b_i	a_{i+1}	b_{i+1}	...

Рис. 7. Непрерывная последовательность рекуррентного кода

Цепной код позволяет установить, какой из символов был искажён: информационный или проверочный. При приёме проверяют выполнение условия

$$\hat{b}_i = \hat{a}_i \oplus \hat{a}_{i+1}. \qquad (8)$$

Искажение одного информационного символа a_i приводит к невыполнению (8) для двух символов: $\hat{b}_{i-1}$ и $\hat{b}_i$. Таким образом, если условие (8) не выполняется для двух соседних проверочных элементов, то искажённым элементом является информационный элемент, находящийся между ними. Это означает, что его следует изменить на противоположный.

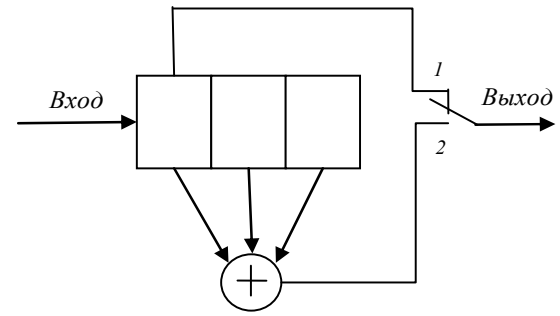
Свёрточные коды относятся к классу непрерывных кодов. В таких кодах информационная последовательность символов не разбивается на отдельные блоки. Передаваемые символы свёрточного кода формируются из K предшествующих информационных символов. Величина K называется *длиной кодового ограничения.*

По аналогии с блочными кодами, свёрточные коды можно классифицировать на разделимые и неразделимые. Разделимым свёрточным кодом является такой код, для которого в выходной последовательности символов содержится без изменений породившая её последовательность информационных символов. В противном случае свёрточный код является неразделимым.

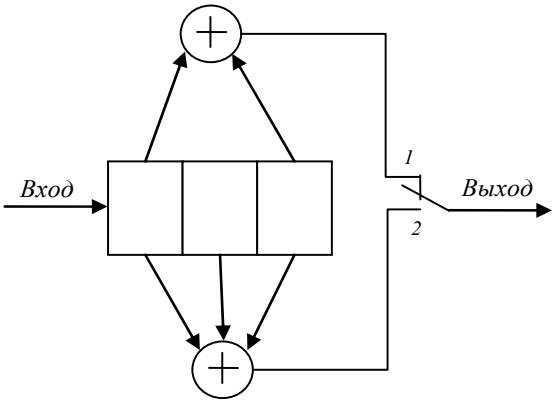
Кодирующее устройство свёрточного кода может быть реализовано с помощью K-разрядного сдвигающего регистра и сумматоров по модулю 2. На рисунках 8 *а*, *б* представлены примеры кодеров разделимого и неразделимого свёрточного кода соответственно.

В каждом из этих кодеров входные двоичные информационные символы поступают в сдвигающий регистр, состоящий из трёх ячеек и находящийся в исходном нулевом состоянии. На каждый входной информационный символ ($k = 1$) кодер вырабатывает два выходных символа ($n = 2$), которые последовательно во времени через коммутатор подаются в канал связи. После прихода очередного входного символа последовательность символов сдвигается в регистре на один символ вправо, в результате чего последний символ выходит за пределы регистра. Свёрточный кодер с параметрами k, n, K обозначается k, n, K. Отношение k/n, как и в блочном кодере, называется относительной скоростью кода. В рассматриваемых примерах $k/n = 1/2$.

В случае разделимого свёрточного кода (рис. 8, *а*) первым из выходных символов всегда будет очередной информационный символ. Из рисунка 8, *б* понятно, что выходная последовательность символов не содержит входные информационные символы в неизменном виде, поэтому кодер на рис. 8, *б* будет порождать неразделимый свёрточный код. На практике обычно используют неразделимые свёрточные коды, поскольку они обладают лучшими корректирующими свойствами, чем разделимые при прочих равных условиях.



а)



б)

Рис. 8. Примеры кодеров разделимого (*а*) и неразделимого (*б*) свёрточного кода ($k = 1$, $n = 2$, $K = 3$)

Для пояснения процессов кодирования и декодирования свёрточных кодов удобно использовать так называемую решётчатую диаграмму. Такая диаграмма для кодера (рис. 8, *б*) представлена на рис. 9.

Рассматриваемая решётчатая диаграмма состоит из узлов и ветвей (рёбер). Число узлов равно 2^{K-1} (K – число ячеек в регистре сдвига). Каждому из четырёх узлов соответствует содержание двух левых ячеек регистра. Это двоичное число называют *состоянием* кодера. Ветви, соединяющие узлы, характеризуют переход из одного состояния кодера в другое. Число ветвей, исходящих из каждого узла, равно основанию кода. Ветвь в виде штриховой линии соответствует входному информационному символу «1», а ветвь в виде сплошной линии – символу «0». Над ветвью записываются формируемые выходные символы.

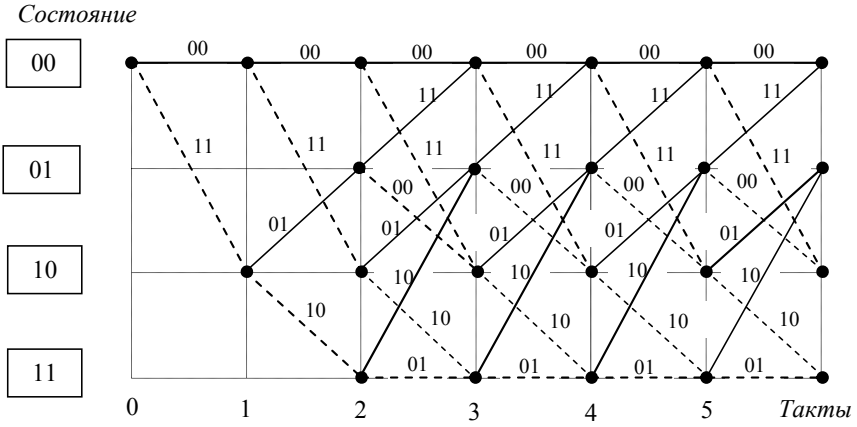


Рис. 9. Решётчатая диаграмма свёрточного кода с *K*=3

Таким образом, каждой входной информационной последовательности соответствует некоторый путь на решётчатой диаграмме. Например, входная последовательность 101100 даёт выходную последовательность 110100101011. Декодирование свёрточных кодов можно осуществлять различными методами. Метод декодирования разделимых свёрточных кодов по своей сущности ничем не отличается от метода декодирования блочных кодов.

На приёмной стороне из принятых информационных символов формируют проверочные символы по тому же закону, что и на передающей стороне, которые затем сравнивают с принимаемыми проверочными символами. В результате сравнения (суммирования по модулю 2) образуется синдром или проверочная последовательность, которая при отсутствии ошибок состоит из одних нулей. При наличии ошибок на определённых позициях последовательности появляются единичные символы. Закон формирования проверочных символов выбирается таким образом, чтобы по виду проверочной последовательности можно было определить позиции искажённых символов. Основное достоинство этого метода декодирования – простота реализации. Однако этот метод применим только для разделимых кодов, которые не полностью реализуют потенциальные корректирующие способности свёрточных кодов.

Наиболее распространённым методом декодирования свёрточных кодов является метод Витерби, в основе которого лежит принцип максимума правдоподобия. Декодирование по этому принципу сводится к сравнению принятой последовательности со всеми другими возможными последовательностями. В качестве истинной (наиболее правдоподобной) последовательности выбирается та, которая в меньшем числе позиций

отличается от принятой кодовой последовательности. Однако при большой длине информационной последовательности N такой метод практически нереализуем, поскольку необходимо перебирать 2^N возможные кодовые последовательности. Существенное упрощение процедуры декодирования по максимуму правдоподобия предложил Витерби. Характерной особенностью метода Витерби является то, что при декодировании запоминается только 2^{K-1} наиболее правдоподобных путей, т.е. количество путей равно числу узлов в сечении решётчатой диаграммы (рис 9 – число узлов в сечении равно четырём).

Турбо-коды являются результатом совместного использования идей свёрточного кодирования с мягким решением и перемежения символов (перемежение символов хорошо себя зарекомендовало, если возникает группирование ошибок). Блок A из $к$ информационных через перемежители поступает на N элементарных систематических свёрточных кодеров. Они могут быть различными и иметь разные скорости. Структурная схема кодера при $N = 2$ показана на рис. 10.

На выходах элементарных кодеров 1 и 2 формируются две последовательности проверочных бит B_1 и B_2, что даёт скорость, равную 1/3. Декодирование турбо-кода выполняется элементарными декодерами с мягким решением с учётом перемежения символов, выполненного на передающей стороне. Структурная схема турбо-декодера для $N = 2$ показана на рис. 11. Перемежители (П), идентичные перемежителю кодера, согласовывают порядок поступления бит A и оценок этих бит, вырабатываемых декодером *1*. Деперемежители (ДП) восстанавливают порядок поступления оценок с выхода декодера *2* для первого декодера. Таким образом, при оценке символа учитываются не только принятые биты, но и мягкие решения, вынесенные каждым элементарным декодером. Турбо-коды являются единственными из известных, позволяющими работать со скоростями, близкими к пропускной способности канала с ограниченной полосой.

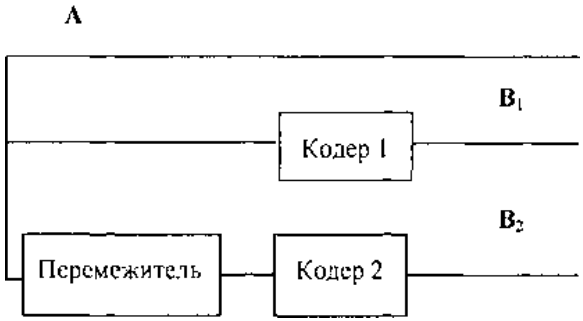


Рис. 10. Структурная схема турбо-кодера

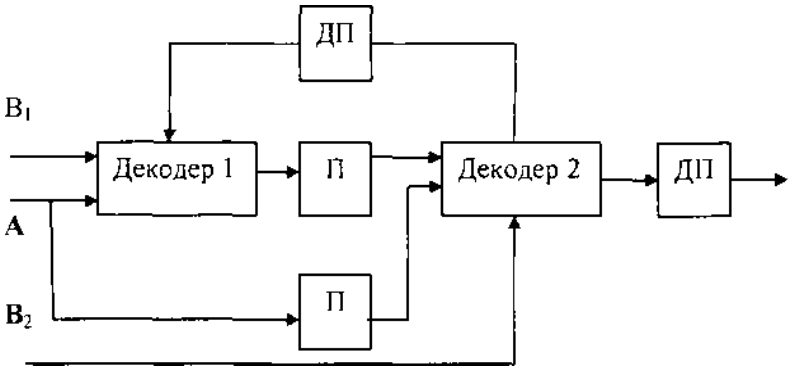


Рис. 11. Декодер турбо-кода

Большинство из известных хороших кодов ориентированы на модель случайных независимых ошибок, т.е. канал без памяти. Для систем же мобильной связи (в числе многих других) характерны глубокие замирания радиосигнала, означающие корреляцию ошибок, в результате которой послед группируются в *пакеты.* При этом появление на выходе демодулятора последовательности неверных символов может стать более вероятным, чем появление только одного. В принципе существуют специальные коды, корректирующие пакетные ошибки большей кратностью, чем кратность контролируемых случайных ошибок, однако в практике чаще прибегают к более испытанному средству, которым является *перемежение,* приспосабливающее традиционные коды к каналам с памятью.

Предложено много алгоритмов перемежения, в частности по периодическим и псевдослучайным законам, блочные и свёрточные. Если некоторые параметры правил перемежения сделать секретными, то получится шифрование данных методом перестановки.

1.3. ЦИКЛИЧЕСКИЕ КОДЫ

Циклические коды относятся к классу линейных систематических кодов. Они названы циклическими потому, что циклический сдвиг разрешённой кодовой комбинации приводит также к разрешённой комбинации. Например, если $b_{n-1}b_{n-2}...b_1b_0$ – разрешённая кодовая комбинация, тогда после её циклической перестановки получим новую кодовую комбинацию $b_{n-2}b_{n-3}...b_0b_{n-1}$, которая также является разрешённой комбинацией циклического кода. Циклическое свойство этих кодов позволило упростить реализацию кодеров и декодеров и дало возможность в системах связи строить эффективные блочные коды большой длины с большим количеством разрешённых кодовых комбинаций.

Сущность циклической перестановки заключается в том, что крайний левый символ кодовой комбинации (символ старшего разряда) занимает место первого, первый – второго и т.д. до тех пор, пока предпоследний символ не займёт место последнего. Запишем совокупность кодовых комбинаций, получающихся циклическим сдвигом *n*-разрядной кодовой комбинации, например пятиразрядной 10011:

10011, 00111, 01110, 11100, 11001.

При работе с циклическими помехоустойчивыми кодами кодовые комбинации $b_{n-1}b_{n-2}...b_1b_0$ принято рассматривать в виде полиномов (многочленов) некоторой степени

$$G(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + ...b_1x^1 + b_0,$$

где x – основание системы счисления; b_i – цифры данной системы счисления.

Для двоичной системы счисления $x = 2$, а b_i равны 0 или 1. Например, двоичную последовательность 010101 можно представить в виде многочлена $x^4 + x^2 + 1$. Действительно,

$$G(2) = 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0;$$

$$G(x) = 0 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x + 1 \cdot x^0 = x^4 + x^2 + 1.$$

Представление кодовых комбинаций в виде многочленов позволяет установить однозначное соответствие между ними и свести действия над кодовыми комбинациями к действиям над многочленами. Над многочленами можно производить все алгебраические действия. С учётом того, что суммирование и вычитание по модулю два дают один и тот же результат, используется только операция сложения по модулю два.

Вышеупомянутый циклический сдвиг некоторого многочлена соответствует простому умножению на x. Умножив, например, многочлен $x^3 + x^2 + 1$, соответствующий комбинации 001101, на x, получим многочлен $x^4 + x^3 + x$, соответствующий комбинации 011010.

Циклические коды характеризуются тем, что многочлены разрешённых кодовых комбинаций $F(x)$ делятся без остатка на так называемый *порождающий* (образующий) многочлен $P(x)$. Порождающим многочленом может быть любой многочлен $P(x)$ степени $r = n - k$, который делит без остатка двучлен $x^n + 1$. Например, порождающий многочлен $x^2 + x + 1$ делит без остатка двучлен $x^6 + 1$, а многочлен $x^3 + x + 1$ делит двучлен $x^7 + 1$. В теории циклических кодов доказывается, что наилучшими свойствами обладают коды с порождающими многочленами в виде неприводимых (примитивных) многочленов. Такие многочлены, как и простые числа, делятся только на себя и единицу. Некоторые неприводимые многочлены приведены в табл. 1.

Таблица 1

Степень многочлена r	Порождающий многочлен $P(x)$
2	$x^2 + x + 1$
3	$x^3 + x + 1$ $x^3 + x^2 + 1$
4	$x^4 + x + 1$ $x^4 + x^3 + 1$
5	$x^5 + x^2 + 1$ $x^5 + x^3 + 1$ $x^5 + x^2 + x + 1$ $x^5 + x^4 + x^3 + x^2 + 1$
6	$x^6 + x + 1$ $x^6 + x^3 + 1$ $x^6 + x^5 + x^2 + x + 1$
7	$x^7 + x^3 + 1$ $x^7 + x^3 + x^2 + x + 1$ $x^7 + x^4 + x^3 + x^2 + 1$
8	$x^8 + x^7 + x^6 + x^5 + x^2 + x + 1$ $x^8 + x^4 + x^3 + x^2 + 1$ $x^8 + x^6 + x^5 + x + 1$
9	$x^9 + x^5 + x^3 + x^2 + 1$ $x^9 + x^8 + x^7 + x^6 + x^5 + x^3 + 1$
10	$x^{10} + x^4 + x^3 + x + 1$ $x^{10} + x^9 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$
11	$x^{11} + x^{10} + x^9 + x^8 + x^3 + x + 1$ $x^{11} + x^8 + x^6 + x^2 + 1$

Процесс циклического кодирования сводится к отысканию многочлена $F(x)$ по известным многочленам $G(x)$ и $P(x)$, который бы делился на $P(x)$ без остатка. Здесь $G(x)$ – многочлен степени $k - 1$, соответствующий информационной последовательности символов. Очевидно, что правило построения циклического кода можно представить в виде произведения информационной кодовой комбинации на порождающий многочлен:

$$F(x) = G(x)\,P(x)\,. \qquad (9)$$

Найдём разрешённую кодовую комбинацию циклического кода (7,4) для кодовой комбинации первичного кода 0111. Так как, число информационных элементов известно ($k = 4$), то число проверочных элементов $r = n - k = 3$. По таблице порождающих многочленов (табл. 1) при $r = 3$ выбираем любой многочлен (один из двух). Пусть $P(x) = x^3 + x + 1$. Представляем кодовую комбинацию 0111 в виде многочлена:

$$G(x) = x^2 + x + 1.$$

Умножаем $G(x)$ на порождающий многочлен $P(x)$:

$$F(x) = (x^2 + x + 1)(x^3 + x + 1) = x^5 + x^4 + 1.$$

Записываем полученный многочлен $F(x)$ в виде разрешённой кодовой комбинации:

0 1 1 0 0 0 1.

На практике применяется другой способ нахождения многочлена $F(x)$:

$$F(x) = G(x)\,x^r \oplus R(x), \qquad (10)$$

где $R(x)$ – остаток от деления произведения $G(x)\,x^r$ на порождающий многочлен $P(x)$. Результат такого деления можно представить в виде

$$\frac{G(x)\cdot x^r}{P(x)} = Q(x) \oplus \frac{R(x)}{P(x)}.$$

Здесь $Q(x)$ – частное от деления.

Построение разрешённых кодовых комбинаций данным способом сводится к следующему:

1. Представляем информационные кодовые комбинации длиной k в виде многочленов $G(x)$.
2. Умножаем $G(x)$ на одночлен x^r, т.е. осуществляем сдвиг кодовой комбинации на r разрядов.
3. Делим $G(x)x^r$ на порождающий многочлен $P(x)$ степени r (табл. 1), чтобы получить остаток $R(x)$.
4. Добавляем $R(x)$ к $G(x)x^r$.
5. Представляем многочлен $F(x) = G(x)\,x^r \oplus R(x)$ в виде кодовой комбинации.

Этот способ кодирования широко применяется на практике.

Построим разрешённую кодовую комбинацию циклического кода (7,4), вышеуказанным способом. В качестве информационной кодовой комбинации возьмем 0111, и порождающий многочлен $P(x) = x^3 + x + 1$.

Записываем кодовую комбинацию 0111 в виде многочлена:

$$G(x) = x^2 + x + 1.$$

Умножаем $G(x)$ на x^3 (так как $r = 3$):

$$G(x)\,x^3 = (x^2 + x + 1)\,x^3 = x^5 + x^4 + x^3.$$

Делим $G(x)\,x^3$ на порождающий многочлен $P(x)$. Остаток от деления $R(x) = x$.

Получаем многочлен

$$F(x) = G(x)\,x^r \oplus R(x) = x^5 + x^4 + x^3 + x$$

и записываем его в виде разрешённой кодовой комбинации.

Видно, что в полученной кодовой комбинации можно выделить информационные и проверочные символы.

Обнаружение ошибок при циклическом кодировании сводится к делению принятой кодовой комбинации на тот же порождающий многочлен, который использовался при кодировании. Если в принятой кодовой комбинации нет ошибок, то деление произойдет без остатка. Если при делении получится остаток, то это свидетельствует о наличии ошибки.

Исправление ошибок при циклическом кодировании основано на анализе остатка $R(x)$, полученного при делении принятой кодовой комбинации $\widehat{F}(x)$ на порождающий полином $P(x)$.

Следует отметить, что для всех разрешённых кодовых комбинаций ошибка на одной и той же позиции даёт одинаковый остаток. Поэтому в принципе ошибки можно исправлять на основе таблицы соответствий между видом остатка и номером ошибочной позиции в кодовой комбинации. Однако свойство цикличности позволяет избавиться от запоминания в памяти декодера остатков для исправления ошибок и тем самым существенно упростить процедуру декодирования.

Существует несколько методов декодирования циклических кодов. Один из методов исправления ошибок циклическим кодом осуществляют следующим образом:

1. Принятую кодовую комбинацию $\widehat{F}(x)$ делят на порождающий многочлен $P(x)$ и вычисляют вес остатка g, т.е. подсчитывают число единиц в остатке $R(x)$. Если в результате деления вес $g \le t$ (t – число исправляемых данным кодом ошибок), то принятую кодовую комбинацию складывают по модулю 2 с остатком, и она будет исправленной.

2. Если в результате деления вес остатка $g > t$, то производят циклический сдвиг вправо принятой комбинации. Полученную комбинацию опять делят на $P(x)$ и вычисляют вес остатка. Если вес нового остатка $g \le t$, то делимую комбинацию складывают по модулю 2 с остатком и полученную комбинацию циклически сдвигают влево на один разряд, что даёт исправленную кодовую комбинацию.

3. Если после второго деления вес нового остатка $g > t$, то процедуру циклического сдвига вправо, деления на $P(x)$ и вычисления веса остатка продолжают до тех пор, пока не будет выполнено условие $g \le t$. Как только это условие выполнится на некотором i-м шаге, полученную комбинацию суммируют с остатком и производят обратный циклический сдвиг суммы на i-разрядов. В итоге получаем исправленную кодовую комбинацию.

Контрольные вопросы

1. В чём состоит принцип помехоустойчивого кодирования?
2. Какие помехоустойчивые коды вам известны?
3. Какие характеристики помехоустойчивых кодов вам известны?
4. Как зависит от минимального кодового расстояния кода его свойства по обнаружению исправлению ошибок в кодовых комбинациях?
5. Какие кодовые комбинации циклического кода считаются разрешёнными?
6. Какая основная логическая операция используется в процессе кодирования и декодирования в циклических кодах?
7. Что такое образующий (порождающий) полином циклического кода и как от его степени зависят корректирующие свойства кода?
8. Чем отличаются друг от друга две возможные процедуры кодирования в циклических кодах, и какая из них используется на практике?
9. Чему должен быть равен остаток от деления принятой кодовой комбинации циклического кода на использовавшийся в процессе кодирования образующий полином, если в ней нет ошибок?

Задачи для самостоятельного решения

Задача 1. Записать кодовые комбинации в виде многочленов:

110110110; 011011101.

Задача 2. Сложить два многочлена: $x^4 + x^3 + x + 1$ и $x^4 + x^2 + x + 1$. Записать указанные многочлены в виде двоичных чисел и произвести их сложение.

Задача 3. Умножить многочлен $x^4 + x^3 + x + 1$ на x^4 и полученный результат разделить на $x^3 + x^2 + 1$. Произвести эти же операции в форме двоичных чисел.

Задача 4. Задана кодовая комбинация первичного семиэлементного кода 1001011. Образовать кодовую комбинацию циклического кода (9,7). (Произведите проверку путём деления полученной комбинации на образующий полином).

Задача 5. Образовать кодовую комбинацию циклического кода (11,7), если кодовая комбинация первичного кода КОИ-7 (1111011) соответствует передаче знака «Ш». Произведите проверку.

Задача 6. Образовать кодовую комбинацию циклического кода (12,8), если кодовая комбинация первичного кода КОИ-8 (11000100) соответствует передаче знака «Ф». Произведите проверку.

Задача 7. Принятая кодовая комбинация циклического кода (7,4) 1001101 содержит один ошибочный символ. Произвести декодирования комбинации и получить исправленную кодовую комбинацию, учитывая, что при образовании циклического кода использовался образующий полином $x^3 + x + 1$.

Задача 8. Принятая кодовая комбинация циклического кода (7,4) 1111001 содержит один ошибочный символ. Произвести декодирования комбинации и получить исправленную кодовую комбинацию, учитывая, что при образовании циклического кода использовался образующий полином $x^3 + x + 1$.

Задача 9. Закодировать свою фамилию циклическим помехоустойчивым кодом на основе образующего полинома степени $r = 3$. Внести во вторую позицию слева, полученной кодовой комбинации, две ошибки подряд. Обнаружить и исправить ошибки. Если исправить ошибки не удаётся, то необходимо увеличить r и повторить процедуру исправления ошибки.

Практическое занятие 5

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ЦИКЛИЧЕСКИХ КОДОВ

Цель: получение практических навыков в выполнение исследований характеристик помехоустойчивых кодов мобильных систем

В результате выполнения практического занятия обучаемые *должны*:

– *знать* теоретический материал по теме занятия и методику исследования эффективности помехоустойчивых кодов на основе моделирования потока ошибок;

– *уметь* строить кодовые комбинации циклических кодов и обнаруживать и исправлять ошибки в кодовых комбинациях циклических кодов при их декодировании.

Проведение практического занятия включает три этапа:

1. Предварительная подготовка к занятию – проработка теоретического материала студентами на занятии и в ходе самостоятельной работы.

2. Основная часть – письменный опрос и решение задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 32 – 75].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. ОБЩИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ И ПОТОКОВ ОШИБОК В МОБИЛЬНЫХ СИСТЕМАХ

Если скорость передачи информации меньше максимально возможной скорости передачи в данном канале, то существует метод помехоустойчивого кодирования, позволяющий получить сколь угодно малую вероятность ошибки на символ.

Однако это утверждение не указывало на пути создания таких помехоустойчивых кодов, но оно заставило учёных ещё раз вернуться к этому вопросу. Рассмотрим только самые общие принципы, на которых базируется помехоустойчивое кодирование. При этом не следует путать понятия *первичного* и помехоустойчивого кодирования. Под первичным кодированием понимается процесс преобразования дискретных сообщений в цифровые сигналы; каждому возможному сообщению из некоторого множе-

ства однозначно определяется кодовая комбинация единичных элементов цифрового сигнала. Первичный код обычно задаётся в виде таблицы, в которой приведены возможные сообщения и соответствующие им кодовые комбинации.

Любой код (первичный или помехоустойчивый) характеризуется следующими параметрами:

n – длиной кодовой комбинации (числом элементов);

m – основанием кода;

N – числом кодовых комбинаций;

$d_{\min}$ – минимальным кодовым расстоянием;

ω – весом.

Длина кодовой комбинации определяется количеством входящих в неё элементов.

В цифровых системах связи используются двоичные коды с *основанием кода* $m = 2$, при которых возможные значения амплитуды единичных импульсов отождествляются с символами 1 и 0.

Число кодовых комбинаций для равномерного кода с основанием m и длиной n определяется выражением

$$N = m^n.$$

Под весом *двоичного кодового слова* понимается число ненулевых символов в кодовом слове $x = 100110 \rightarrow \omega(x) = 3$.

Кодовым расстоянием между двумя комбинациями одинаковой длины называется число позиций кодовых комбинаций, в которых данные комбинации отличаются. Оно равно числу единиц (весу) суммы кодовых комбинаций по модулю два.

Минимальным кодовым расстоянием $d_{\min}$ называется наименьшее из кодовых расстояний всех возможных пар комбинаций данного кода. Для первичных кодов $d_{\min} = 1$.

Помехоустойчивое кодирование часто называют *избыточным* кодированием. Одним из простейших примеров избыточного кодирования является повторение элемента цифрового сигнала несколько раз. Помехоустойчивые коды можно разделить на коды *исправляющие ошибки* и коды, только *обнаруживающие* ошибки. Рассмотрим идею помехоустойчивого кодирования и выясним, от чего зависят обнаруживающие и исправляющие способности кодов. Выполним это на примере простейшего помехоустойчивого кода с повторением, для минимального кодового расстояния $d_{\min} = 3$. Для упрощения будем считать, что алфавит состоит только из двух букв: А и Б. Первичный код может иметь вид: А – 0; Б – 1. Введём в первичный код избыточность: А – 000; Б – 111. Поясним алгоритм декодирования на рис. 1.

А – 000 разрешённая кодовая комбинация 1

001, 100, 010 } запрещённые кодовые комбинации 1 } передавалась буква А

010, 011, 110 } запрещённые кодовые комбинации 2 } передавалась буква Б

101

Б – 111 разрешённая кодовая комбинация 2

Рис. 1. Алгоритм декодирования помехоустойчивого кода с повторением

Под разрешёнными кодовыми комбинациями понимаются кодовые комбинации первичного кода, в этой задаче их только две. Остальные кодовые комбинации, которые могут быть приняты, в результате возникновения ошибок, называются запрещёнными.

Обнаружение ошибки сводится к следующему: если принята одна из разрешённых кодовых комбинаций – 1 или 2, то выносится решение, что передавались буквы А или Б соответственно. Если принята одна из запрещённых комбинаций – выносится решение, что обнаружена ошибка. Понятно, что рассматриваемый код способен обнаруживать только две ошибки. Если ошибок больше – принимается разрешённая кодовая комбинация, т.е., количество обнаруженных ошибок $S = d_{\min} - 1$. Если количество ошибок три, то ошибку уже обнаружить невозможно.

Исправление ошибок возможно только в том случае, если запрещённые кодовые комбинации можно разделить таким образом, что попадание в запрещённую кодовую комбинацию 1 возможно только из разрешённой кодовой комбинации 1, а в запрещённую кодовую комбинацию 2 – только из разрешённой кодовой комбинации 2. Это возможно, если количество ошибок не превышает одной, т.е., количество исправляемых ошибок $t = (d_{\min} - 1)/2$. Если ошибок две и более, ошибку исправить уже невозможно.

Таким образом, число обнаруживаемых s и исправляемых t ошибок связано с минимальным кодовым расстоянием следующими соотношениями:

$$s \le d_{\min} - 1,$$
$$t \le (d_{\min} - 1)/2. \qquad (1)$$

Эта формула справедлива и для любых других помехоустойчивых кодов, но её использование в практических приложениях не всегда удобно, к тому же неравенства нестрогие. Поэтому водятся следующие характеристики:

1) вероятность необнаруживаемой ошибки $P_{\text{н.о}}$ – вероятность выдачи декодером ошибочной кодовой комбинации;

2) коэффициент повышения достоверности $K_{\text{п.д}}$, показывающий, во сколько раз уменьшается вероятность появления ошибочных кодовых комбинаций на выходе декодирующего устройства, по сравнению с вероятностью ошибочного приёма кодовой комбинации в канале связи. Для кодов, обнаруживающих ошибки, коэффициент повышения достоверности

$$K_{\text{п.д}} = P_{\text{о.ш}} / P_{\text{н.о}}.$$

Вероятность не обнаружения ошибки и коэффициент повышения достоверности определяют эффективность избыточного кода.

1.2. ОЦЕНКА ЭФФЕКТИВНОСТИ ЦИКЛИЧЕСКОГО КОДИРОВАНИЯ

Одной из важнейших характеристик эффективности избыточных кодов является вероятность необнаруженной ошибки $P_{\text{н.о}}$. Обнаруживающая способность циклических кодов зависит как от минимального кодового расстояния, так и от степени образующего полинома и его вида. Оценку вероятности необнаруженной ошибки можно произвести по приближённой формуле

$$P_{\text{н.о}} \approx \frac{1}{2^r} \sum_{i=d_{\min}}^{n} P(t=i), \qquad (2)$$

либо путём имитационного моделирования цифрового канала связи на ПЭВМ. И в том и в другом случае для оценки эффективности кодов необходимо располагать статистическими данными потока ошибок. Эти данные сводятся, например, в табл. 1, где $P(t=i) \approx N(t=i)/N$.

Таблица 1

n	N	$N(t=0)$	$N(t=01$	$N(t=2)$	$N(t=3)$	$N(t=4)$	$P(t=1)$	$P(t=2)$	$P(t=3)$	$P(t=4)$

Непосредственная оценка обнаруживающих свойств циклических кодов, проводится путём имитационного моделирования кодера и декодера циклического кода. При этом выясняются обнаруживающие способности кода для выбранного образующего полинома. Данные потока ошибок сводятся в таблицу. В этом случае

$$P_{\text{н.о}} \approx N_{\text{ош}} / N , \tag{3}$$

где $N_{\text{ош}}$ – количество блоков, в которых ошибка есть, но она не обнаружена; N – количество переданных блоков. Поток ошибок получают на основе действующего цифрового канала связи. При выполнении научных исследований на первых этапах исследования можно ограничиться математической моделью источника ошибок на ЭВМ.

1.3. МОДЕЛИ ИСТОЧНИКА ОШИБОК

В случае, когда *ошибки в канале появляются независимо* с вероятностью P_e, вероятность появления в n-элементной комбинации t ошибок $P(t, n)$ определяется биномиальным распределением

$$P(t,\ n) = C_n^t\, P_e^t (1 - P_e)^{n-1},$$
$$C_n^t = \frac{n!}{t!(n-1)!} .$$

Вероятность приёма неискажённой комбинации $(t = 0)$

$$P(0,\ n) = (1 - P_e)^n ,$$

а вероятность появления m и более ошибок

$$P(\geq m,\ n) = \sum_{t=m}^{n} C_n^t\, P_e^t (1 - P_e)^{n-t} .$$

Модель Гильберта и модель Беннета–Фройлиха позволяют учитывать группирование ошибок в канале связи. Простейшей моделью, учитывающей группирование ошибок, является модель Бергера–Мандельброта. Обобщением модели Беннета–Фройлиха является модель Попова–Турина. Недостатком этих моделей является то, что они не учитывают нестационарность потока ошибок в каналах (часовые, дневные и недельные вариа-

ции). Поэтому, с точки зрения адекватности модели реальным каналам, наиболее перспективной следует считать модель дискретного канала с переменными параметрами.

Поясним *моделирование потока ошибок в дискретном канале*, для этого обозначим через 0 правильно принятый элемент, а через 1 – элемент, принятый неправильно. Это так называемый посимвольный метод описания ошибок в дискретном канале. На практике чаще всего используется другой метод – поинтервальный. Сущность его заключается в записи потока ошибок в виде последовательности чисел, равных длинам интервалов между двумя последовательными ошибками и выраженных в единичных элементах. Если записать поток ошибок…01001010000011…, то используя поинтервальный метод описания, придем к последовательности интервалов между ошибками … 2150… .

Реализация математической модели источника ошибок на ЭВМ состоит в генерировании случайной последовательности неискажённых интервалов в соответствии с законом их распределения. Для каналов с независимыми ошибками (каналов без памяти) длина интервала между ошибками τ подчиняется геометрическому закону

$$p(\tau) = P_e(1 - P_e^{\tau-1}). \qquad (4)$$

Для канала с памятью, описываемого моделью Бергера–Мандельброта, интегральная функция распределения длин интервалов между ошибками подчиняется закону Парето с показателем $\alpha < 1$:

$$F(\tau) = \begin{cases} 1 - \tau^{-\alpha}, & \tau \geq 1; \\ 0, & \tau = 0. \end{cases} \qquad (5)$$

В (4) коэффициент $0 < \alpha < 1$ зависит от скорости и вида модуляции, типа канала связи, степени группирования ошибок в канале связи и др. Если $\alpha \approx 0{,}1$ – это соответствует биноминальной модели. В этом случае нет пакетирования (группирования) ошибок. Наибольшее значение α (от 0,5 до 0,7) наблюдается на кабельных линиях связи (кратковременное прерывание связи). В радиорелейных линиях (где бывают интервалы с большой интенсивностью ошибок и интервалы с редкими ошибками) $\alpha = 0{,}3...0{,}5$; для некоторых линий коротковолновой радиосвязи $\alpha = 0{,}3...0{,}4$.

Одним из основных способов получения случайных чисел (неискажённых интервалов между ошибками) с заданным распределением является способ, основанный на использовании следующей теоремы:

Если случайная величина y^* $(y^* \geq 0)$ имеет плотность распределения f_y, то распределение случайной величины

$$\eta = \int_0^{y^*} f(y)\, dy$$

является равномерным в интервале [0, 1].

Известно, что для получения случайных величин, принадлежащих совокупности $\{\tau_i\}$ и имеющих плотность распределения $f(\tau)$, необходимо разрешить уравнения относительно τ_i:

$$\int_0^{\tau_i} f(\tau)\, d\tau = \eta_i . \qquad (6)$$

Решая уравнение (6) относительно фиксированной последовательности реализаций η_i равномерно распределённой случайной величины η, получаем соответствующую последовательность реализаций τ_i, имеющих заданную плотность $f(\tau)$.

Контрольные вопросы

1. В чём заключается отличие между первичным и помехоустойчивым кодированием, какие первичные коды Вы знаете?
2. Перечислить и пояснить основные характеристики кодов.
3. Объяснить общую идею помехоустойчивого кодирования.
4. От какого показателя зависит количество обнаруживаемых и исправляемых ошибок ПК? Привести формулу.
5. Дать классификацию ПК, определить в ней место циклических кодов (ЦК).
6. Пояснить на конкретных примерах процедуры кодирования и декодирования ЦК.
7. Чем обусловлена необходимость оценки эффективности ПК, привести основные показатели?
8. Почему не всегда достаточно при оценке эффективности ПК ограничиться только аналитическими расчётами?

Задачи для самостоятельного решения

Структура исследуемой системы связи представлена на рис. 2.

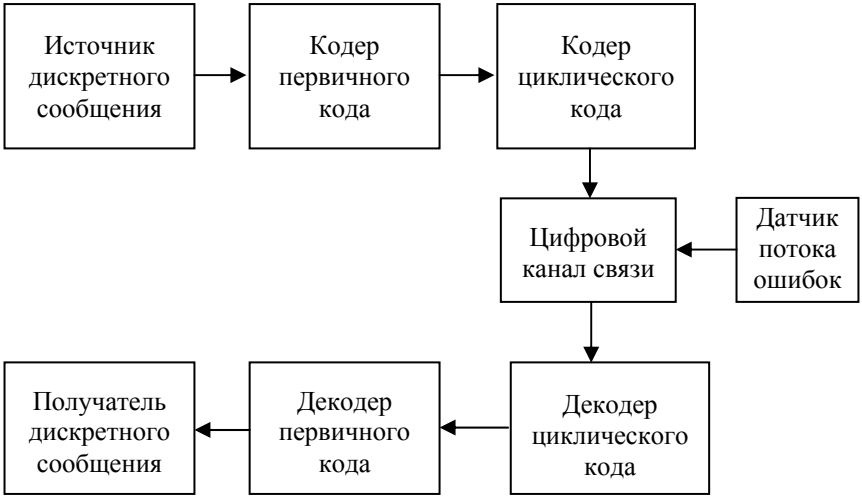


Рис. 2. Структура цифрового канала

В программной среде DELPHI реализованы:

1) кодер первичного кода КОИ – 8;

2) датчик потока ошибок;

3) кодер циклического кода;

4) декодер (обнаружение ошибок) циклического кода.

Необходимо количественно оценить коэффициент необнаружения ошибок ЦК по выражениям (5) и (6), в соответствии с методикой, изложенной выше. Для чего необходимо выполнить следующую последовательность действий:

1. Сформировать дискретное сообщение, например:

…ТРУД СДЕЛАЛ ИЗ ОБЕЗЬЯНЫ ЧЕЛОВЕКА … .

2. Вызвать программу, моделирующую кодеры первичного и циклического кодов, и поблочно кодировать текст первичным, а затем циклическим кодами.

3. Вызвать программу датчика потока ошибок и сформировать последовательность ошибок. Пример последовательности может иметь вид

…. 2 5 4 7 13 …..

4. В соответствии с потоком ошибок внести искажения в закодированное сообщение. Допустим, неискажённая кодовая последовательность с $n = 7$ имеет вид

1001100 0011110 0100010 1010101 0101001.

После искажения в соответствии с потоком ошибок получим

$$10\ \underbrace{\mathbf{1}}_{2}\ 11\underbrace{000}_{5}\ \underbrace{\mathbf{0}}\ 1\underbrace{111}_{4}\ \underbrace{\mathbf{0}}\ 0\underbrace{100010}_{7}\ \mathbf{1}\ \underbrace{0101010101001}_{12},$$

где жирным шрифтом обозначены внесённые ошибки.

5. Проанализировать искажённое сообщение и заполнить табл. 1.

6. Получить количественную оценку коэффициента необнаружения ошибок в соответствии с выражением (5).

7. Искажённое сообщение декодировать, и по синдрому ошибки (остатку) вынести решение об обнаружении ошибки или её отсутствии. Если синдром ошибки равен 0, а ошибка на самом деле присутствует, значит, она не обнаружена. Подсчитав количество блоков $N_{\text{ош}}$ в которых ошибка не обнаружена, и подставив эту величину в выражение (6), получим количественную оценку коэффициента необнаружения ошибок.

Практическое занятие 6

ПЕРЕДАЧА НЕПРЕРЫВНЫХ СООБЩЕНИЙ ПО ЦИФРОВЫМ КАНАЛАМ

Цель: углубление теоретических знаний по передаче непрерывных сообщений

В результате выполнения практического занятия обучаемые *должны*:

– *знать* принцип преобразования непрерывных сообщений в цифровой сигнал;

– *уметь* анализировать требования к цифровым каналам передачи непрерывных сообщений.

Практическое занятие включает три этапа:

1. Предварительная подготовка к занятию – проработка теоретического материала студентом самостоятельно.
2. Основная часть – устный или письменный опрос, решение задач и рассмотрение технологий в современных системах мобильной связи.
3. Оформление отчёта и защита полученных результатов

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 76 – 110].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. ОБЩИЕ СВЕДЕНИЯ

Импульсными называются системы связи, в которых сигналы представляют собой радиоимпульсы, параметры которых (амплитуда, длительность или время появления) изменяются по закону передаваемого сообщения. Исходя из изменяемого информационного параметра импульсов, различают системы связи с *амплитудно-импульсной* (АИМ), *широтно-импульсной* (ШИМ) и *фазоимпульсной* (ФИМ) модуляцией. Упрощенная структура системы радиосвязи с импульсной модуляцией представлена на рис. 1, где ИМ – импульсный модулятор; ИДМ – импульсный демодулятор; ПРД – передатчик (фактически усилитель); ПРМ – приёмник.

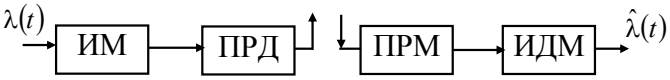
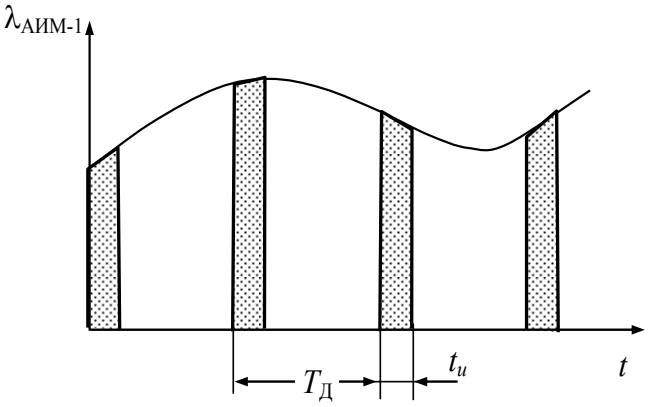


Рис. 1. Упрощённая структурная схема СПИ с импульсной модуляцией

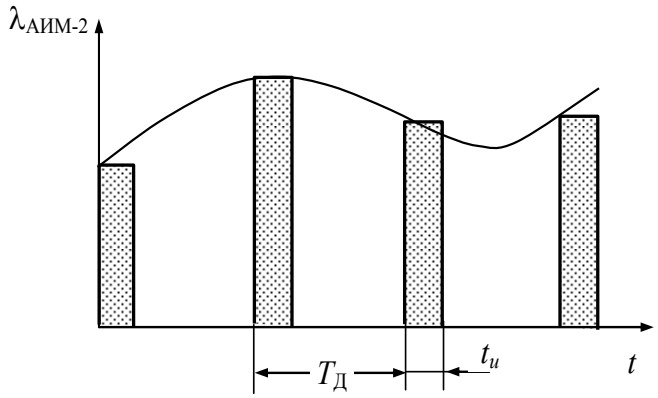
1.2. АМПЛИТУДНО-ИМПУЛЬСНАЯ МОДУЛЯЦИЯ

При АИМ (рис. 2) по закону информационного параметра сообщения изменяется амплитуда излучаемых радиоимпульсов.

Временная диаграмма АИМ-АМ-1 радиосигнала изображена на рис. 3.



а)



б)

Рис. 2. Формирование сигналов при АИМ первого (*а*) и второго (*б*) рода
($T_Д$ – интервал дискретизации непрерывного сообщения;
$t_и$ – длительность импульса)

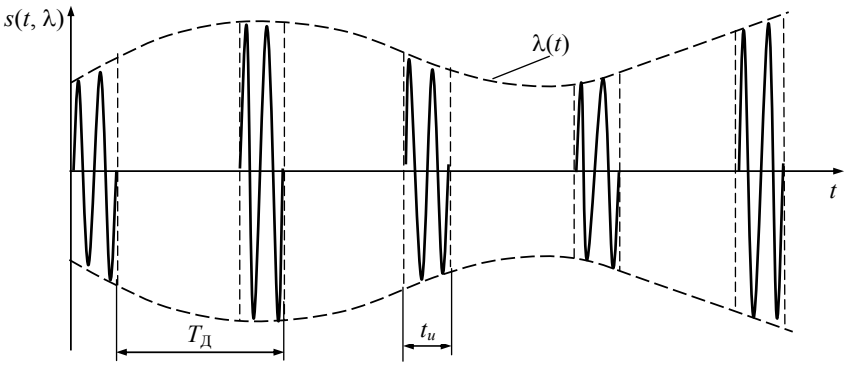


Рис. 3. Временная диаграмма радиосигнала при АИМ-АМ-1

1.3. ШИРОТНО-ИМПУЛЬСНАЯ МОДУЛЯЦИЯ

При ШИМ в соответствии с сообщением изменяется длительность импульса (рис. 4).

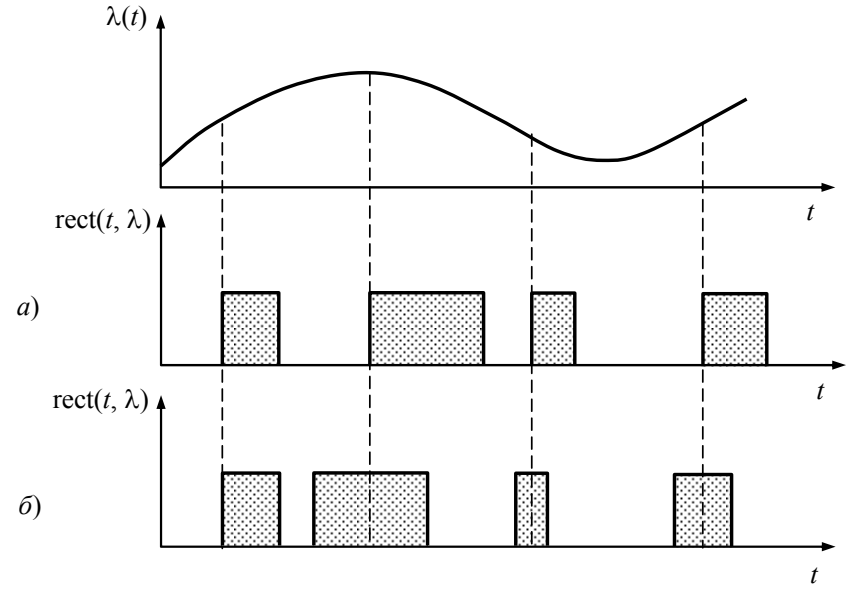


Рис. 4. Формирование сигналов при односторонней ОШИМ-АМ (*а*) и двухсторонней ДШИМ-АМ (*б*) и двусторонней (*б*) ШИМ

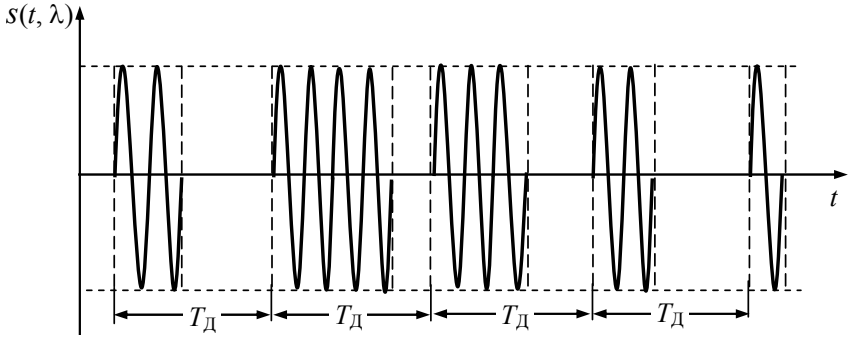


Рис. 5. Временная диаграмма радиосигнала при ОШИМ-АМ

Реализация радиосигнала ОШИМ-АМ приведена на рис. 5.

ШИМ практически используется в многоканальных линиях импульсной радиосвязи.

1.4. ФАЗОИМПУЛЬСНАЯ МОДУЛЯЦИЯ

При этом виде модуляции каждый импульсный сигнал по своему положению во времени перемещается от тактовой точки на время, пропорциональное значению сообщения (рис. 6).

Радиосигнал при односторонней фазоимпульсной модуляцией (ФИМ-АМ) показан на рис. 7.

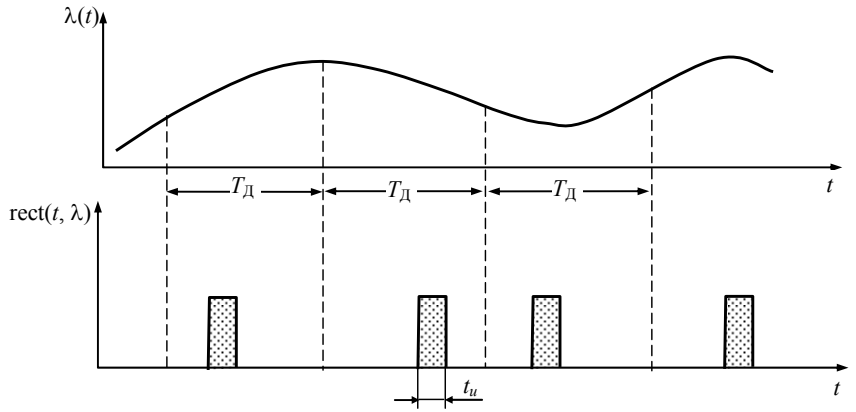


Рис. 6. Формирование сигналов при односторонней ФИМ

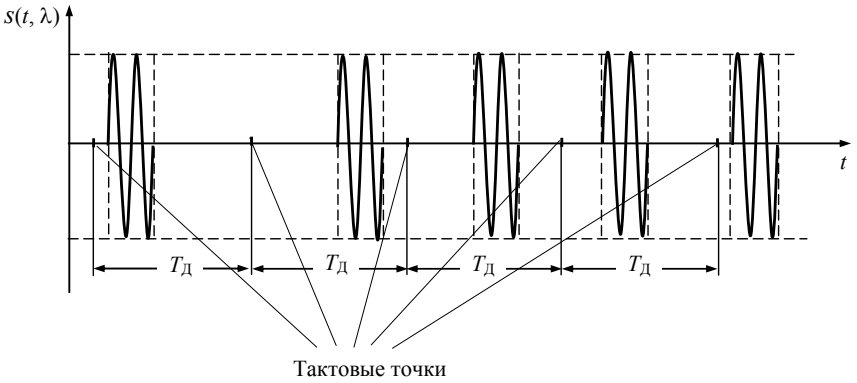


Рис. 7. Временная диаграмма радиосигнала при односторонней ФИМ-АМ

Основным достоинством систем связи с ФИМ-АМ является их наиболее высокая помехоустойчивость среди систем с импульсной модуляцией.

1.5. ОБЩИЕ СВЕДЕНИЯ О ПЕРЕДАЧЕ НЕПРЕРЫВНЫХ СООБЩЕНИЙ ЦИФРОВЫМИ СИГНАЛАМИ

Цифровая передача непрерывных сообщений обычно осуществляется на основе схемы, представленной на рис. 8.

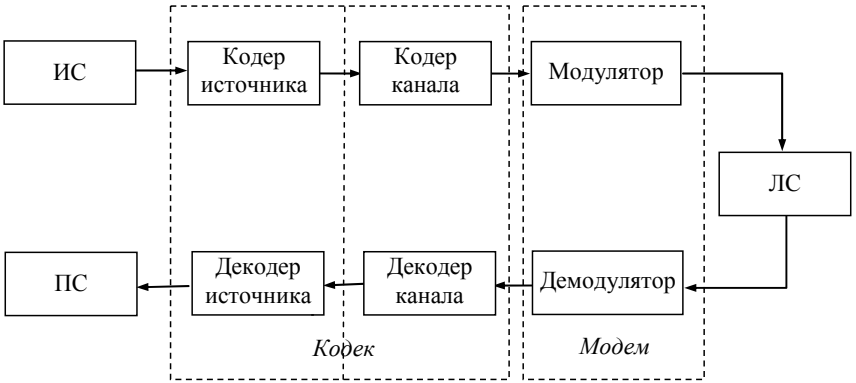


Рис. 8. Структурная схема цифровой системы передачи сообщений

Кодер источника – осуществляет преобразование непрерывного сообщения в последовательность кодовых символов (в кодовые комбинации). *Кодер канала* – добавляет в цифровой сигнал, получаемый с выхода кодера источника, дополнительную (избыточную) информацию, предназначенную для защиты от ошибок передаваемого сигнала по линии связи. Кроме того, в кодере канала может быть предусмотрено перемежение символов кодовых последовательностей, т.е. их перестановка некоторым детерминированным способом. Перемежение позволяет преобразовать групповые ошибки в одиночные, которые легче исправлять с помощью помехоустойчивых кодов.

Модулятор – осуществляет перенос кодированного видеосигнала на несущую частоту. При этом используются различные виды манипуляции: амплитудная, частотная, фазовая и их разновидности.

Приёмное устройство по составу в основном соответствует передающему, но с обратными функциями входящих в него блоков: демодулятора, декодера канала, декодера источника.

Демодулятор – выделяет из модулированного радиосигнала кодированный видеосигнал, несущий информацию.

Декодер канала – проверяет принятый сигнал на наличие ошибок, и выявленные ошибки по возможности исправляются.

Декодер источника – преобразует поступающий на него с декодера канала сигнал (кодовые комбинации) в знаки сообщения, т.е. восстанавливает переданное непрерывное сообщение.

Заметим, что принято различать две группы относительно самостоятельных устройств: кодеки и модемы. *Кодеком* называется конструктивно совмещённая совокупность кодера и декодера, а *модемом* – совокупность модулятора и демодулятора. Системы связи с ИКМ, дифференциальной ИКМ и ДМ различаются между собой способами построения кодеков. Рассмотрим сначала систему связи с ИКМ, в которой кодер представляет собой аналого-цифровой преобразователь (АЦП), а декодер – цифроаналоговый (ЦАП).

1.6. ИМПУЛЬСНО-КОДОВАЯ МОДУЛЯЦИЯ

Преобразование аналогового сигнала в цифровой с использованием АЦП предусматривает выполнение трёх основных операций (рис. 9):

1) дискретизацию аналогового сигнала по времени, в результате чего формируются дискретные отсчёты;

2) квантование дискретных отсчётов по уровню;

3) кодирование квантованных отсчётов.

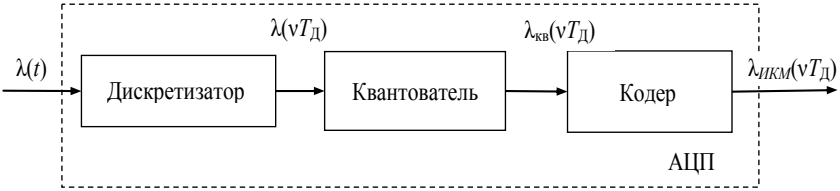


Рис. 9. Схема преобразования аналогового сигнала в ИКМ сигнал

Дискретизация сигналов по времени заключается в замене аналогового сигнала последовательностью его отсчётов, в соответствии с известной теоремой Котельникова. Согласно этой теореме *любой непрерывный сигнал* $\lambda(t)$, *ограниченный по спектру верхней частотой* $F_в$, *полностью определяется последовательностью дискретных отсчётов* $\lambda(\nu T_Д)$, *взятых через интервал времени* $T_Д \le 1/2F_в$, *называемый интервалом* (*периодом*, *шагом*) *дискретизации.*

Процесс преобразования непрерывного сообщения в цифровой сигнал показан на рис. 10.

Квантование по уровню заключается в замене непрерывного множества мгновенных значений входного сигнала дискретным множеством заранее определённых значений, которые называют *уровнями квантования.* В процессе квантования диапазон возможных значений входного сигнала делится на интервалы, и операция квантования сводится к тому, что всем отсчётам входного сигнала, попавшим в некоторый интервал, приписывается одно и то же значение (один и тот же уровень квантования).

Обычно уровни и интервалы квантования выбирают равномерно, т.е.

$$\lambda_i - \lambda_{i-1} = \Delta\,, \tag{1}$$

$$\lambda_{кв_i} - \lambda_{кв_{i-1}} = \Delta\,, \tag{2}$$

где Δ – *шаг квантования*. Если шаг квантования выбирается постоянным, то квантование называют равномерным.

Таким образом, для полного интервала $2\lambda_{max}$ величину шага квантования получим в следующем виде:

$$\Delta = \frac{2\lambda_{max}}{2^n}\,, \tag{3}$$

где λ_{max} – максимальная величина $\lambda(t)$.

Кодирование квантованных отсчётов заключается в присвоении каждому *i*-му уровню квантования соответствующей кодовой комбинации. В большинстве случаев для кодирования квантованных отсчётов используются двоичные коды, состоящие из кодовых комбинаций длиной *n.*

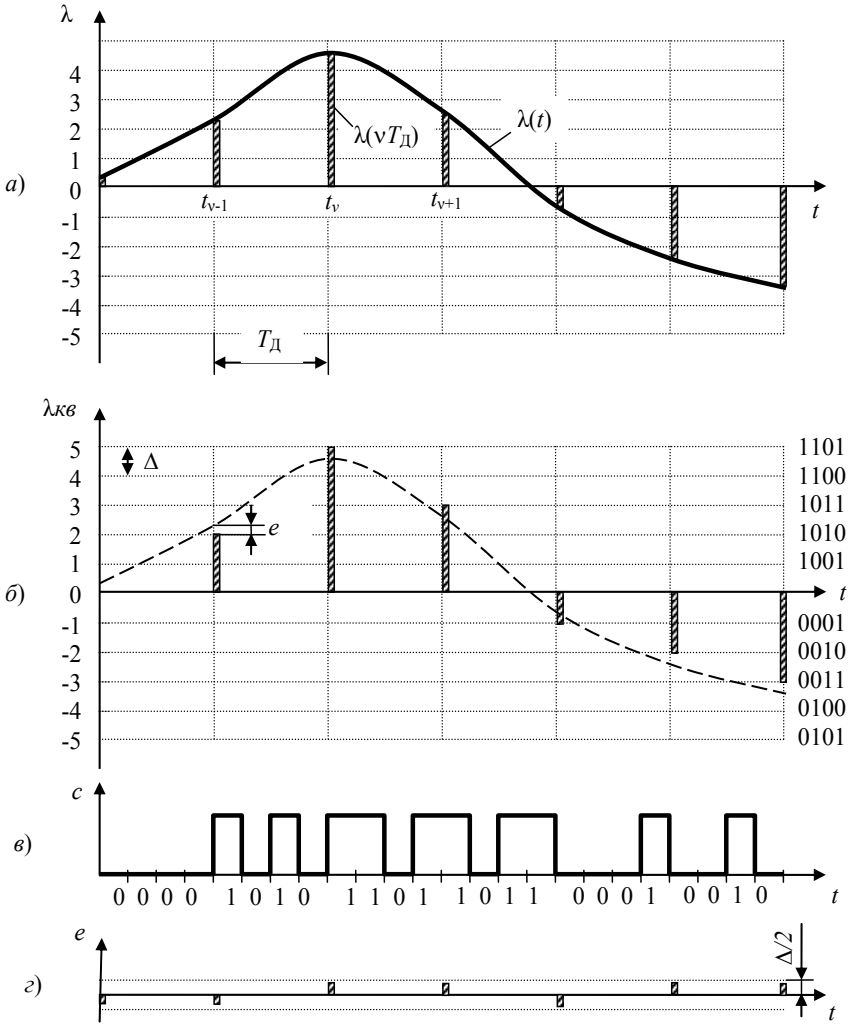


Рис. 10. Преобразование аналогового сигнала в ИКМ сигнал

С помощью таких кодов можно представить 2^n различных уровней квантования. Например, для передачи речевого сигнала обычно используют 8-разрядный двоичный код, который обеспечивает кодирование $N = 2^8 = 256$ уровней квантования. На практике наибольшее применение получили натуральный двоичный код, симметричный двоичный код и код Грея.

Таблица 1

Десятичное число	Натуральный десятичный код	Код Грея	Десятичное число	Натуральный десятичный код	Код Грея
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Натуральный двоичный код – это код, кодовые комбинации которого представляют собой запись натуральных чисел (номеров уровней квантования) в двоичной системе счисления (табл. 1). Например, в натуральном двоичном коде с $n = 4$ числу 13 соответствует кодовая комбинация 1101 ($13 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 1$).

Симметричный двоичный код используется для кодирования положительных и отрицательных квантованных величин. Первый символ кодовой комбинации такого кода несёт информацию о полярности отсчёта, а остальные символы – об его абсолютном значении. Например, если $n = 4$, то квантованным уровням 3 и –3 будут соответствовать кодовые комбинации 1011 и 0011 (рис. 10, *б*). В симметричном двоичном коде старший разряд комбинаций, равный 1, обозначает положительную величину отсчёта, а 0 – отрицательную.

Код Грея – это код, в котором любые две кодовые комбинации, соответствующие соседним уровням квантования, отличаются друг от друга только одним разрядом (табл. 1). Такая особенность построения кода Грея позволяет повысить быстродействие кодирующих устройств. Кодовые комбинации кода Грея можно получить из кодовых комбинаций натурального кода за счёт сложения по модулю два исходной комбинации с той же комбинацией, сдвинутой вправо на один разряд. При этом младший разряд сдвинутой комбинации отбрасывается. В качестве примера получим кодовую комбинацию кода Грея, соответствующую десятичному числу 12 (табл. 1):

$S_n(\omega)$ 1100 – исходная комбинация натурального кода

110 0 – сдвинутая исходная комбинация

$\overline{1010}$ – полученная комбинация кода Грея.

Для передачи двоичных символов ИКМ сигнала по дискретному каналу связи могут использоваться различные виды манипуляции: амплитудная, частотная, фазовая. Скорость передачи двоичных символов (скорость цифрового потока) определяется соотношением

$$B = nF_{\text{Д}} \text{ бит/с}, \tag{4}$$

где n – число бит на отсчёт сигнала.

Преобразование ИКМ сигнала в аналоговый сигнал (рис. 11) предусматривает выполнение двух основных операций

1) декодирование кодовой последовательности;

2) преобразование квантованного АИМ сигнала в аналоговый сигнал.

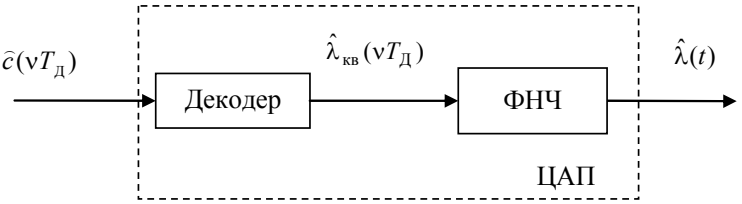


Рис. 11. Схема преобразования ИКМ сигнала в аналоговый сигнал

Временные диаграммы преобразований ИКМ сигнала в аналоговый сигнал, представлены на рис. 12.

Ошибка квантования $e(\nu)$ представляет собой разность между квантованным $\lambda_{\text{кв}}(\nu)$ и исходным сигналом $\lambda(\nu)$ и называется *шумом квантования*, т.е.

$$e(\nu) = \lambda_{\text{кв}}(\nu) - \lambda(\nu). \tag{5}$$

Дисперсия шума квантования, возникающего при квантовании сигналов, попадающих в пределы i-го шага, находится следующим образом

$$\sigma_i^2 = \int_{-\Delta_i/2}^{+\Delta_i/2} (\lambda_{\text{кв}_i} - \lambda)^2 \, p(\lambda)\, d\lambda \cong \frac{\Delta_i^2}{12} P_i, \tag{6}$$

где $p(\lambda)$ – плотность распределения вероятностей мгновенных значений сигнала $\lambda(t)$; P_i – вероятность появления сигнала с уровнем, лежащим в пределах i-го шага квантования Δ_i.

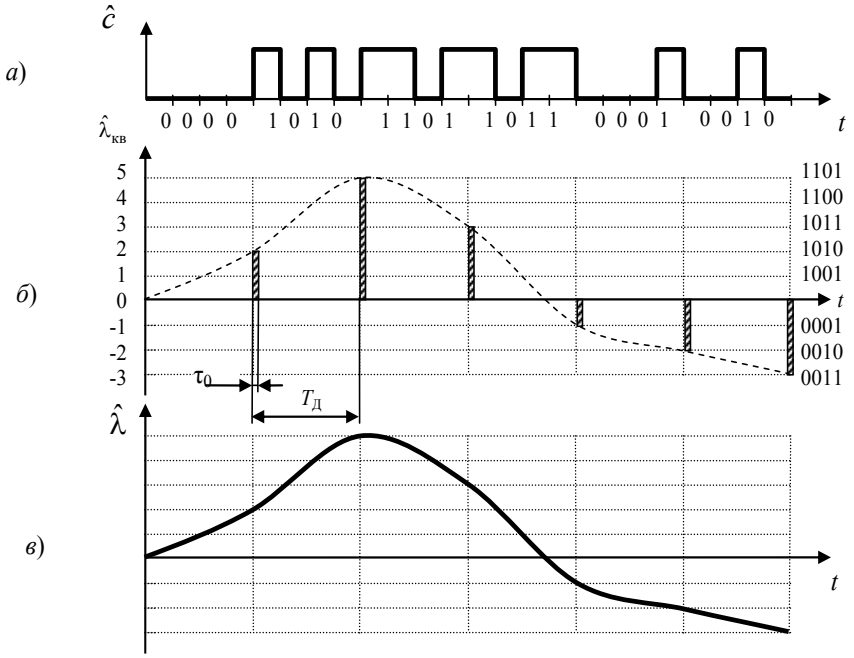


Рис. 12. Преобразование ИКМ сигнала в аналоговый сигнал

Дисперсия полного шума квантования в пределах от $-\lambda_{max}$ до $+\lambda_{max}$ равна сумме составляющих от каждого шага:

$$\sigma_e^2 = \sum_{i=1}^{N} \frac{\Delta_i^2}{12} P_i \,. \tag{7}$$

При равномерном квантовании ($\Delta_i = \Delta = \text{const}$)

$$\sigma_e^2 = \frac{\Delta^2}{12}, \tag{8}$$

так как $\sum_{i=1}^{N} p_i = 1$.

Иная запись выражения для мощности шума квантования

$$\sigma_e^2 = \frac{1}{12}\left(\frac{2\lambda_{max}}{2^n}\right)^2 = \frac{\lambda_{max}^2}{3 \cdot 2^{2n}} \,. \tag{9}$$

Отношение мощности сигнала к мощности шума квантования

$$\rho = \frac{\sigma_\lambda^2}{\sigma_e^2} = 2^{2n} \qquad (10)$$

или в децибелах

$$\rho_{\text{дБ}} = 10\lg(\rho) = 10\lg(2^{2n}) = 20n\lg(2) = 6,02n\,. \qquad (11)$$

1.7. ДИФФЕРЕНЦИАЛЬНАЯ ИМПУЛЬСНО-КОДОВАЯ МОДУЛЯЦИЯ

Сокращение избыточности дифференциальной импульсно-кодовой модуляции (ДИКМ) реализуется путём вычитания предсказанного значения из следующего выборочного значения и передачи значений этой разности в цифре.

Структурная схема, поясняющая принцип построения системы передачи информации с ДИКМ, приведена на рис. 13.

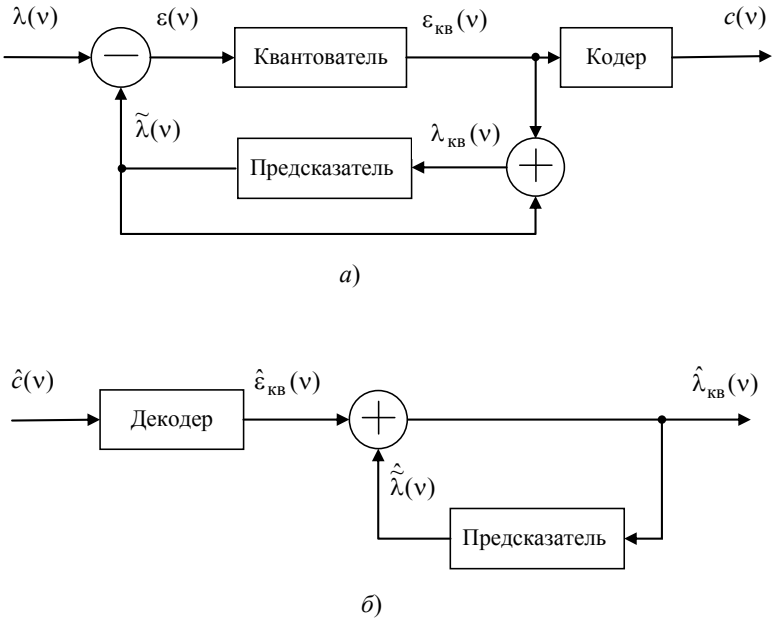


Рис. 13. Структурная схема системы передачи информации с ДИКМ:
а – кодер; *б* – декодер

Последовательность дискретных отсчётов $\lambda(\nu)$ исходного сигнала $\lambda(t)$ поступает на один из входов вычитающего устройства, а на второй вход поступает сигнал предсказания $\tilde{\lambda}(\nu)$, который формируется в предсказателе из предыдущих отсчётов. С выхода вычитающего устройства отсчёты сигнала ошибки предсказания

$$\varepsilon(\nu) = \lambda(\nu) - \tilde{\lambda}(\nu) \tag{12}$$

подвергаются квантованию, кодированию и передаче по каналу связи.

Схема, восстанавливающая переданный сигнал $\lambda(\nu)$ по последовательности кодовых слов $\hat{c}(\nu)$, показана на рис. 13, *б* и содержит декодер, восстанавливающий разностный сигнал $\hat{\varepsilon}_{\text{кв}}(\nu)$, и предсказатель (такой же, как и на передающей стороне). Очевидно, что при совпадении $\hat{c}(\nu)$ и $c(\nu)$ сигнал $\hat{\lambda}_{\text{кв}}(\nu) = \lambda_{\text{кв}}(\nu)$ и отличается от $\lambda(\nu)$ лишь ошибкой квантования $e(\nu)$, вносимой квантованием сигнала $\varepsilon(\nu)$.

Широко используется адаптивная ДИКМ (АДИКМ). Предполагается, что коэффициенты предсказания зависят от времени. В этом случае предсказанное значение имеет вид

$$\tilde{\lambda}(\nu) = \sum_{i=1}^{L} a_i(\nu)\,\lambda_{\text{кв}}(\nu - i)\,. \tag{13}$$

При адаптации коэффициентов предсказателя $a(\nu)$ обычно полагается, что свойства передаваемого сигнала не меняются в течение короткого интервала времени. Коэффициенты предсказания $a(\nu)$ рассчитываются в вычислителе коэффициентов по выборочным значениям входного сигнала и передаются по каналу связи вместе с разностным сигналом $c(\nu)$. Скорость изменения этих адаптивных коэффициентов связана со временем, в течение которого входной сигнал может считаться локально-стационарным.

Например, речь не может изменять свои характеристики быстрее, чем 10 − 20 раз за секунду. Это даёт интервал обновления 50...100 мс. В системах передачи речи обычно для вычисления коэффициентов предсказания принят интервал, равный 20 мс.

В системах АДИКМ адаптация чаще всего распространяется как на предсказатель, так и на квантователь. В этом случае адаптируется ещё и шаг квантования. Основная идея адаптивного квантования состоит в том, что шаг квантования изменяется таким образом, чтобы соответствовать изменяющейся дисперсии входного сигнала. Необходимость адаптивного квантования вызвана тем, что при квантовании сигналов возникают серьёзные трудности. С одной стороны, шаг квантования должен быть достаточно большим для согласования диапазона квантования с размахом сигнала:

$$\Delta \cdot 2^n = 2\lambda_{\max} . \qquad (14)$$

С другой стороны, шаг квантования должен быть малым для уменьшения шума квантования. Как уже отмечалось, один из методов уменьшения шума квантования состоит в применении неравномерного квантования. Другой метод состоит в адаптации свойств квантователя к уровню сигнала.

Адаптивное квантование может осуществляться как по входному сигналу, так и по выходному сигналу. В системах АДИКМ с адаптацией по входному сигналу величина шага $\Delta(\nu)$ выбирается на основании оценок значений $\varepsilon(\nu)$. При этом по каналу связи вместе с кодовыми последовательностями $c(\nu)$ и $a(\nu)$ передаётся информация о значениях $\Delta(\nu)$. В системах АДИКМ с адаптацией по выходу шаг квантования подстраивается по выходному сигналу $\varepsilon_{\text{кв}}(\nu)$ либо по выходной последовательности кодовых символов $c(\nu)$. В этом случае информация о величине $\Delta(\nu)$ в канал связи не передаётся.

1.8. ДЕЛЬТА-МОДУЛЯЦИЯ

Разновидностью ДИКМ является *дельта-модуляция* (ДМ), при которой число уровней квантования разностного сигнала $\varepsilon(\nu)$ равно двум. Уменьшить число уровней квантования до двух и перейти к одноразрядным системам можно за счёт увеличения частоты дискретизации. Действительно, по мере увеличения $F_{\text{Д}}$ сокращается интервал дискретизации, возрастает корреляция между отсчётами и, следовательно, уменьшается динамический диапазон разностного сигнала. В системах ДМ частота дискретизации выбирается много больше чем $2F_{\text{В}}$.

В наиболее простом дельта-модуляторе используется шаг квантования Δ постоянной величины для всех уровней сигнала, поэтому он называется равномерным или *линейным дельта-модулятором.*

Основное достоинство дельта-модуляции состоит в её простоте. Система с ДМ не требует синхронизации по кодовым комбинациям, поскольку передача осуществляется одиночными импульсами, а не кодовыми комбинациями. Требования к выбору шага квантования являются противоречивыми. С одной стороны, шаг квантования должен быть небольшим, чтобы не превысить допустимый уровень шума квантования, а с другой стороны – большим, чтобы исключить искажения, называемые «перегрузкой по крутизне». Такие искажения возникают, когда сигнал предсказания «отстаёт» от исходного сигнала. Для неискажённой передачи необходимо выполнить условие

$$\Delta \geq \left|\frac{d\lambda(t)}{dt}\right|_{\max} \cdot T_{\text{Д}}\,, \qquad (15)$$

где $\left|\frac{d\lambda(t)}{dt}\right|_{\max}$ – максимальное значение крутизны входного сигнала. При постоянном шаге квантования удовлетворить этим требованиям удаётся только при достаточно высокой частоте дискретизации. Так, для получения хорошего качества передачи сигналов, сравнимого с качеством, достигаемым с помощью ИКМ, требуется увеличение скорости цифрового потока в $2-3$ раза по сравнению с ИКМ.

Уменьшить скорость цифрового потока без увеличения шума квантования можно, применяя адаптивные методы в системах с ДМ. Большинство этих методов основано на адаптации шага квантования по выходной последовательности кодовых символов. *Адаптивную дельта-модуляцию* (АДМ), при которой изменяется шаг квантования в зависимости от параметров передаваемого сигнала, также называют АДМ *с компандированием*.

Контрольные вопросы

1. Как изменится полоса пропускания канала передачи данных, если передавать непрерывное сообщение по цифровому каналу связи (ЦКС)?
2. Перечислить достоинства и недостатки при передаче непрерывных сообщений по ЦКС.
3. Пояснить принцип ИКМ, перечислить недостатки этого метода модуляции.

4. Записать и пояснить выражение для интервала дискретизации по времени.

5. Пояснить выбор величины шага квантования. К чему приведёт отклонение шага квантования от оптимального значения?

6. Выполнить сравнительный анализ ИКМ и ДИКМ.

7. Пояснить принцип дельта – модуляции, перечислить достоинства и недостатки.

Задачи для самостоятельного решения

Задача 1. Речевое сообщение передаётся по каналу связи с помощью ИКМ, при этом сообщение имеет равномерное распределение в интервале ±2,0 В. Необходимо определить мощность шума квантования, если АЦП реализует равномерное квантование и имеет разрядность $n = 4, 8, 16$.

Задача 2. Определить отношение мощности сигнала (сообщения) к мощности шума квантования, если при ИКМ используется 8-разрядный АЦП и реализуется равномерное квантование.

Задача 3. На 8- разрядный АЦП поступает аналоговое сообщение с динамическим диапазоном ±5 В ($\Delta\lambda_{\max} = 10$ В). Необходимо определить среднее квадратическое отклонение (СКО) шума квантования.

Задача 4. Определить полосу частот, занимаемую цифровым сигналом ИКМ, если спектральная плотность речевого сообщения ограничена частотой $F_{\text{в}} = 3{,}4$ кГц , а число уровней квантования в АЦП равно 128.

Задача 5. Определить пропускную способность двоичного симметричного канала связи при передаче цифрового ИКМ сигнала с частотой дискретизации $F_{\text{д}} = 8$ кГц и использовании 8-разрядного АЦП, если вероятность ошибочного приёма символа цифрового сигнала $P_e = 0{,}1$.

Задача 6. Определить пропускную способность непрерывного канала связи, если его полоса пропускания равна 3 кГц, а отношение сигнал/шум по мощности в полосе частот канала равно трём.

Задача 7. Для формирования цифрового сигнала при ИКМ используется частота дискретизации $F_{\text{д}} = 8$ кГц , отношение мощности сигнала

(сообщения) к мощности шума квантования обеспечивается равным 48 дБ. Определить: минимальное число уровней квантования, длительность элементарного импульса, скорость цифрового потока.

Задача 8. Построить зависимость отношение мощности сообщения к мощности шума квантования от числа разрядов АЦП: $n = 4, 8, 16$.

Задача 9. Построить зависимость полосы частот занимаемой цифровым сигналом ИКМ от числа уровней квантования, если полоса частот передаваемого речевого сообщения $\Delta F = 3{,}1$ кГц, а количество уровней квантования $N = 64, 128, 256, 512$.

ШИРОКОПОЛОСНЫЕ СИГНАЛЫ В МОБИЛЬНЫХ СИСТЕМАХ ПЕРЕДАЧИ ИНФОРМАЦИИ

Цель: совершенствование теоретических знаний по технологиям мобильных сетей

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: достоинства и недостатки широкополосных систем передачи информации; методику оценки помехоустойчивости приёма широкополосных сигналов.

– *уметь*: оценить потенциальную помехоустойчивость приёма широкополосных сигналов в условиях помех; провести анализ эффективности применения широкополосных сигналов в современных беспроводных сетях передачи информации.

Проведение практического занятия включает три этапа:

1. Предварительная подготовка к занятию – проработка теоретического материала студентом на занятии и в ходе самостоятельной работы.
2. Основная часть – письменный опрос и решение задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

− краткие ответы на поставленные в работе вопросы;

− решение предложенных задач;

− выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 19 – 49].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. ОСНОВНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ ШИРОКОПОЛОСНЫХ СИСТЕМ СВЯЗИ

В ходе разработки систем передачи информации принципиальным является обеспечение надёжной связи в условиях воздействия организованных и непреднамеренных помех, многолучевого распространения радиоволн, а также осуществления многостанционного доступа при работе в беспроводных пакетных сетях.

Теоретической базой для разработки широкополосных систем передачи информации (ШСПИ) стала фундаментальная формула К. Е. Шеннона, которая определяет предельную пропускную способность C (бит/с) канала связи при воздействии на передаваемый сигнал помехи типа ограниченного по полосе приёма аддитивного белого гауссовского шума (БГШ)

$$C = \Delta f_s \log_2\left(1 + \frac{P_s}{P_{\text{п}}}\right), \qquad (1)$$

где Δf_s – ширина полосы частот занимаемой сигналом или полоса пропускания канала связи; P_s и $P_{\text{п}}$ – средние мощности сигнала и помехи в канале, соответственно.

Из (1) следует, что пропускная способность C канала радиосвязи, после того как она задана, в условиях действия аддитивной гауссовской помехи (шума) с ограниченной средней мощностью $P_{\text{п}}$ (Вт) может быть обеспечена либо использованием широкой полосы частот Δf_s (Гц) и малым отношением сигнал–помеха $\frac{P_s}{P_{\text{п}}}$, либо узкой полосы частот Δf_s с более высоким отношением сигнал – помеха $\frac{P_s}{P_{\text{п}}}$. Следовательно, между полосой пропускания канала (шириной спектра передаваемого сигнала) Δf_s и отношением сигнал–помеха $\frac{P_s}{P_{\text{п}}}$ в этом канале возможен взаимообмен.

Основные свойства и характеристики широкополосных систем связи. Вводимая в сигнал частотная избыточность характеризуется *базой сигнала*

$$B = \frac{\Delta f_s}{\Delta F_\lambda}, \qquad (2)$$

где Δf_s – ширина спектра сигнала; ΔF_λ – ширина спектра сообщения.

В обычных узкополосных системах $B \geq 1$. Если $B >> 1$, то имеет место широкополосная система передачи информации.

К основным свойствам ШСПИ относятся:

1. Высокая устойчивость к воздействию сосредоточенных по спектру организованных помех (повышенная помехоустойчивость).

2. Скрытность передачи (малая вероятность обнаружения и перехвата сообщений).

3. Связь многих абонентов в общей полосе частот (обеспечение многостанционного доступа на основе кодового разделения каналов).

4. Высокая разрешающая способность по времени прихода сигнала (борьба с многолучевостью, возможность точной синхронизации).

Повышенная помехоустойчивость ШСПИ определяется тем, что при ограниченной мощности помехи $P_{\text{п}}$ и ширине её спектра $\Delta f_{\text{п}}$, меньшей или равной ширине спектра сигнала Δf_s, применение ШПС позволяет существенно увеличить отношение сигнал–шум относительно узкополосных сигналов. В частности, если $P_{\text{п}} = N_{\text{п}} \Delta f_{\text{п}} / 2$ и $\Delta f_{\text{п}} = \Delta f_{\text{s}}$, то отношение сигнал–шум на выходе корреляционного приёмника или согласованного фильтра (СФ), в случае оптимального приёма, может быть представлено в виде

$$Q_0 = \frac{2E_s}{N_{\text{п}}} = \frac{P_s T}{P_{\text{п}} / \Delta f_s} = \frac{P_s \Delta f_s}{P_{\text{п}} \Delta F_{\lambda}} = \frac{P_s}{P_{\text{п}}} B . \qquad (3)$$

Здесь E_s, P_s и T – соответственно энергия, мощность и длительность принимаемого сигнала; $\frac{N_{\text{п}}}{2}$ – интенсивность спектральной плотности помехи.

Из (3) следует, что отношение сигнал–шум в ШПСС увеличивается пропорционально базе сигнала. Подобный выигрыш в отношении сигнал–шум имеет место только при сосредоточенных по спектру помехах, когда $\Delta f_{\text{п}} \le \Delta f_{\text{s}}$. Помехоустойчивость приёма сигналов на фоне широкополосной помехи типа БГШ определяется только отношением энергии сигнала Å$_s$ к спектральной плотности помехи (шума) $N_{\text{п}}$ и не зависит от вида используемого сигнала.

Скрытность передачи в ШПСС связана с уменьшением уровня спектральной плотности сигнала в результате увеличения его базы. При заданной мощности сигнала $P_s = N_s \Delta f_s / 2$ спектральная плотность ШПС

$$\frac{N_s}{2} = \frac{P_s}{\Delta f_s} = \frac{P_s T}{\Delta f_s T} = \frac{P_s T}{B} \qquad (4)$$

в B раз меньше, чем у узкополосного сигнала (УПС) при равных мощностях и скорости передачи информации. Поэтому в точке приёма при неизвестной структуре ШПС вероятность его обнаружения на фоне помехи (шума) низка (рис. 1).

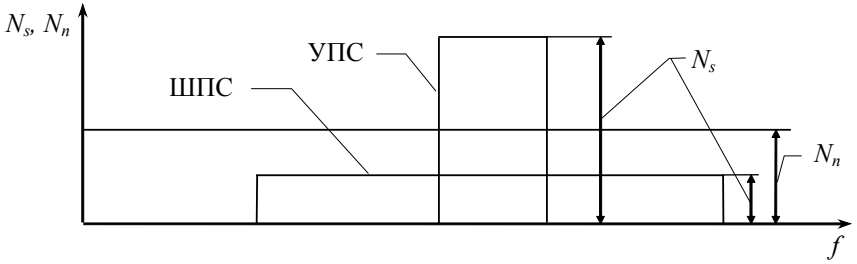


Рис. 1. Спектральные плотности сигналов с различными базами и помехи (шума)

Уменьшение уровня спектральной плотности сигнала в B раз сказывается не только на повышении скрытности передачи информации, но и в ряде случаев позволяет обеспечить лучшую электромагнитную совместимость.

Многостанционный доступ различных абонентов в общей полосе частот на основе использования ШПС может быть обеспечен при кодовом разделении каналов. В этом случае отдельные абоненты имеют разные формы ШПС (расширение спектра частот осуществляется на основе способов модуляции или кодирования, различных для каждого канала). Сигналы разных абонентов выбирают таким образом, чтобы они как можно больше отличались друг от друга, т.е. были приблизительно ортогональны:

$$\int S_k(t)\, S_j(t)\, dt, \;\; k \neq l.$$

Высокая разрешающая способность по времени прихода сигнала основана на том, что ШПС имеют корреляционную функцию с узким пиком, длительность которого обратно пропорциональна ширине спектра сигнала Δf_s.

При оптимальной обработке ШПС приёмные устройства, построенные на основе корреляторов или СФ, дают выходной сигнал, совпадающий с корреляционной функцией ШПС. Ширина пика корреляционной функции обратно пропорциональна ширине спектра сигнала. Поэтому отклики СФ при приёме сигналов в многолучевом канале связи (рис. 2, *а*) для узкополосных сигналов (УПС) (рис. 2, *б*) и ШПС (рис. 2, *в*) определяют возможность разделения при приёме запаздывающих лучей. Когерентное суммирование узких пиков корреляционной функции ШПС позволяет устранить возникающую в результате многолучевости интерференцию принимаемых сигналов и связанных с ней потерь в мощности.

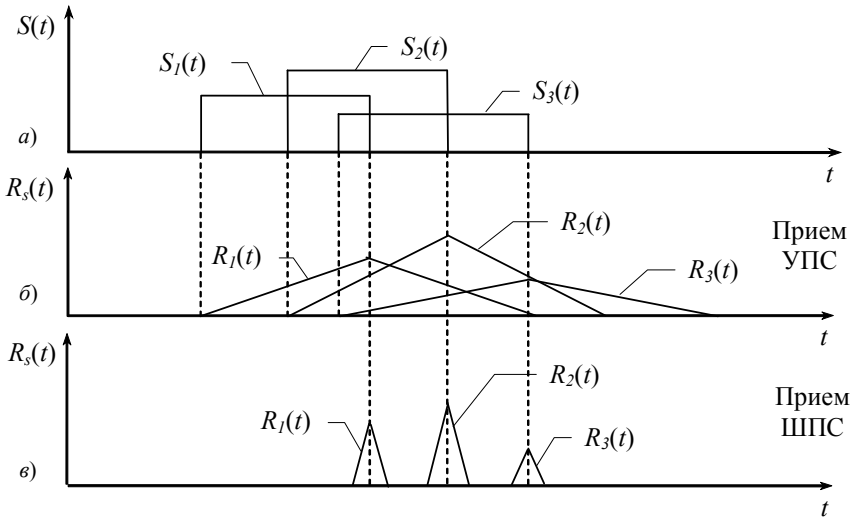


Рис. 2. Сигналы на выходе согласованного фильтра

Возможность разделения отдельных лучей и, как следствие, определение времени прихода запаздывающих сигналов позволяют повысить *точность синхронизации* при использовании ШПС.

К основным характеристикам ШПС с дискретным изменением параметров, относятся:

1) закон (правило) формирования ШПС;

2) число различных вариантов ШПС (ансамбль сигналов) m при заданной базе B;

3) вид корреляционной функции сигнала $R_s(\tau)$;

4) вид спектра (спектральной плотности) сигнала $S_s(\omega)$;

5) взаимокорреляционные характеристики ансамбля сигналов и число квазиортогональных сигналов.

Параллельный ШПС представляет собой совокупность разнесённых по частоте радиоимпульсов, передаваемых одновременно в течение интервала времени T. Этот вид ШПС называется *многочастотным* (МЧ) сигналом.

На рисунке 3 приведены временные диаграммы, поясняющие принцип формирования параллельного ШПС.

На рисунке 4, *а* и *б* приведён примерный вид спектральной плотности $S_s(f)$ и огибающей корреляционной функции $R(\tau)$ параллельного ШПС при количестве элементов $L = 20$.

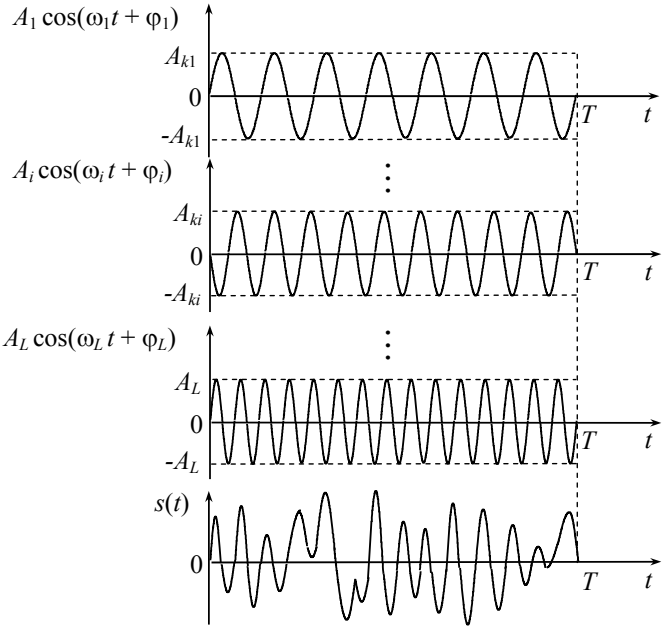
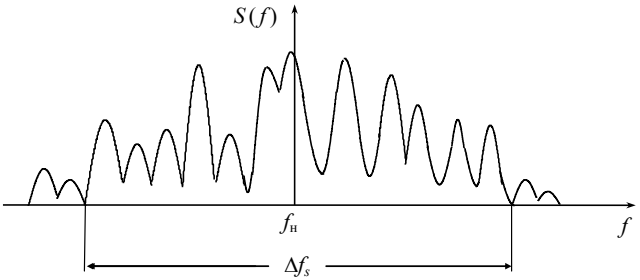


Рис. 3. Формирование параллельного ШПС

а)



б)

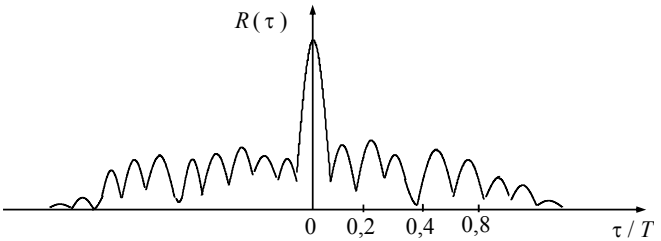


Рис. 4. Спектральная плотность и корреляционная функция параллельного ШПС

Из рисунка видно, что основная энергия сигнала, заключена в области частот шириной Δf_s вокруг некоторой несущей $f_н$ (средней частоты в спектре). Спектральная плотность имеет точки, равные нулю или близкие к нулю. Положение этих точек определяется фазовым кодом. Огибающая корреляционной функции имеет основной максимум в окрестности точки $\tau = 0$. Форма основного максимума и побочных выбросов существенно зависит от выбранного кода.

Процесс формирования последовательного ШПС с двоичной внутриимпульсной ФМн по закону расширяющей последовательности $p(t)$ понятен из рис. 5.

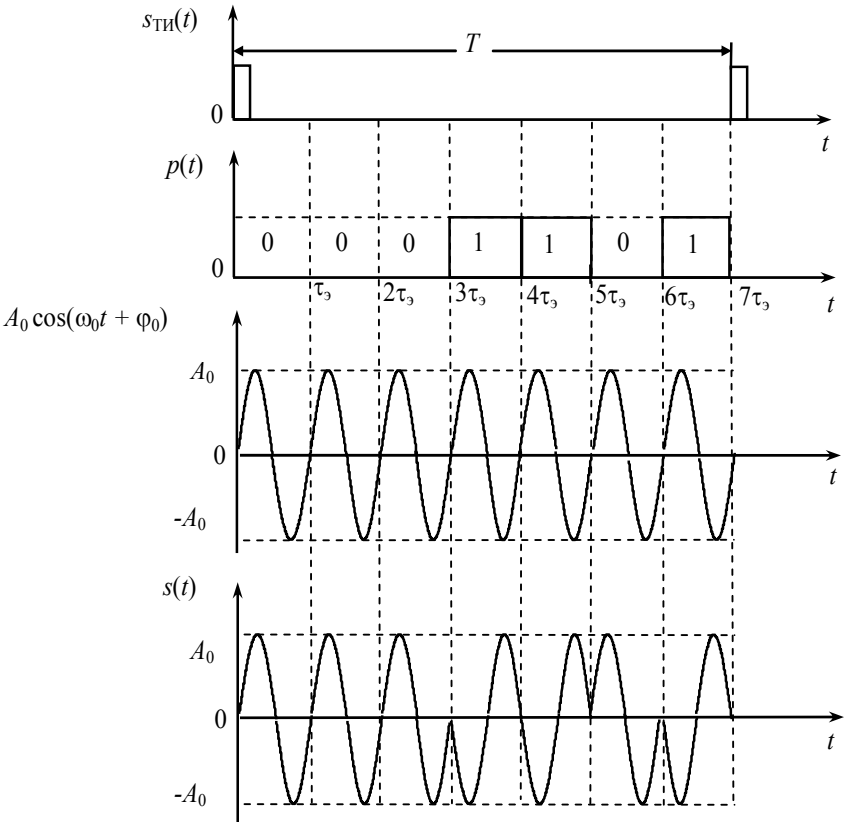
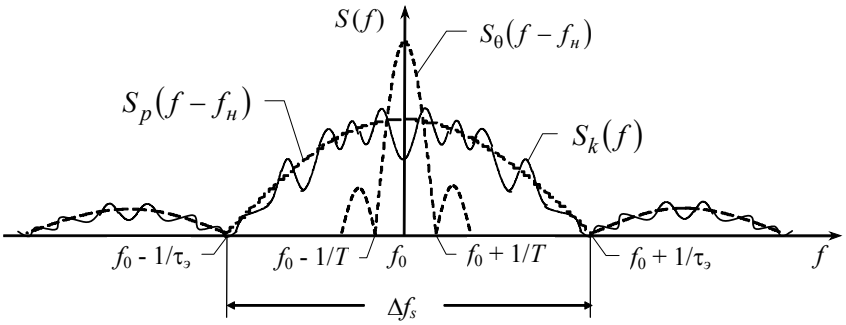


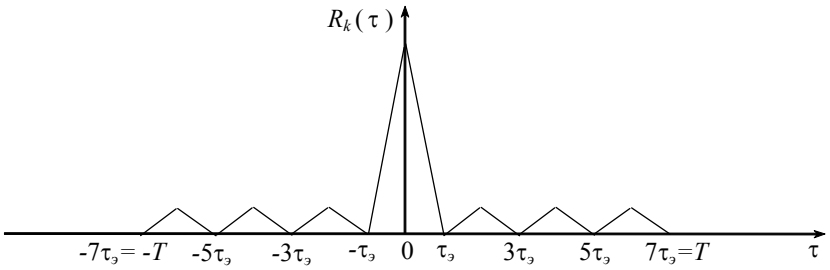
Рис. 5. Формирование последовательного ШПС с двоичной внутриимпульсной ФМн

Спектральная плотность и огибающая корреляционной функции последовательного ШПС, соответствующего представленному на рис. 6, *а* наряду со спектральной плотностью ШПС, приведены изображенные пунктиром осредненные спектральные плотности информационного сигнала θ(*t*) и расширяющей информационной последовательности *p*(*t*), перенесённые на некоторую несущую частоту f_0. Спектральная плотность последовательного ШПС $S_k(f)$ по существу является результатом суперпозиции спектральных плотностей $S_\theta(f - f_0)$ и $S_\text{р}(f - f_0)$.

Представленная на рис. 6, *б* огибающая корреляционной функции имеет узкий основной пик и относительно малый уровень боковых лепестков. Это свидетельствует о хороших корреляционных свойствах последовательного одночастотного ШПС с двоичной внутриимпульсной ФМн.



а)



б)

Рис. 6. Спектральная плотность и корреляционная функция *k*-го варианта последовательного ШПС с двоичной ФМн

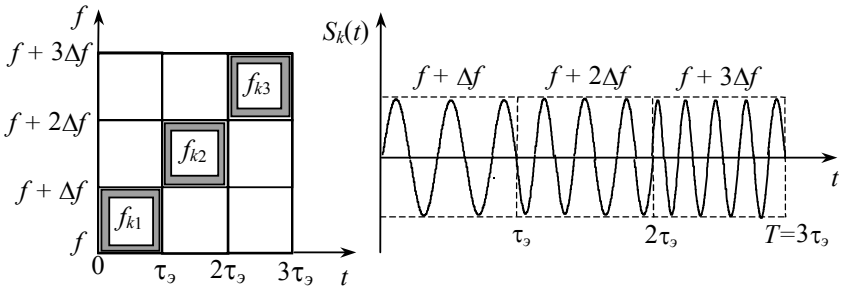


Рис. 7. Частотно-временная матрица и временная диаграмма трёхэлементного последовательно-параллельного ШПС с быстрой ППРЧ

Широкое применение последовательных ШПС с двоичной внутриимпульсной ФМн объясняется их хорошими корреляционными свойствами. В качестве расширяющей последовательности используются коды Баркера, последовательности Голда, последовательности Хаффмена (или *M*-последовательности). Эти последовательности позволяют получать хорошие спектральные и корреляционные характеристики сигнала.

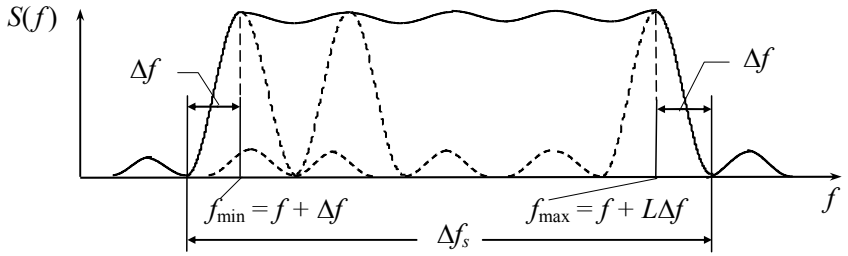
К широкополосным сигналам также часто относят сигналы со скачкообразным изменением несущей частоты по закону псевдослучайной последовательности (ПСП). Такие сигналы называют сигналами с *псевдослучайной*, *или программно*, *перестройкой рабочей частоты* (ППРЧ*)*.

В сигналах с ППРЧ рабочая частота может изменяться несколько раз за время передачи одного бита информации (быстрая ППРЧ) или один раз за время передачи нескольких бит (медленная ППРЧ).

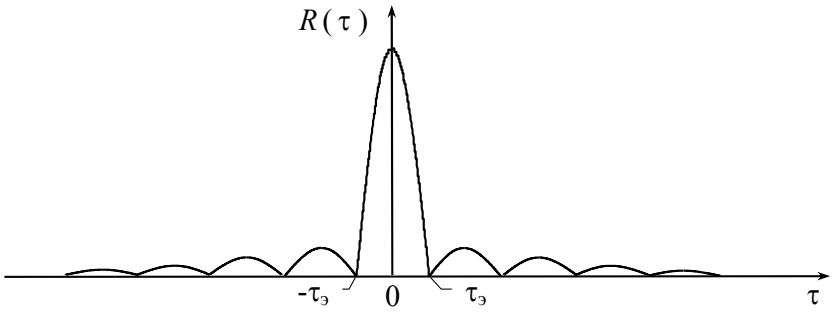
Пример частотно-временной матрицы (ЧВМ) и соответствующей ей временной диаграммы трёхэлементного последовательно-параллельного ШПС с быстрой ППРЧ приведён на рис. 7.

На рисунке 8, *а* и *б* соответственно приведены спектральная плотность и корреляционная функция последовательно-параллельного ШПС, когда *i* последовательно принимает значения 1, 2, ... , *L*, а частотный сдвиг $\Delta f = \Delta\omega/2\pi = 1/\tau_э$.

В общем случае спектр последовательно – параллельного ШПС имеет сложный характер, так как определяется не только амплитудными, но и фазовыми спектрами элементарных импульсов. Наличие временного и частотного сдвигов между составляющими отдельных импульсов может приводить к провалам вплоть до нуля в амплитудном спектре результирующего сигнала. Приведённая на рис. 8, *а* спектральная плотность соответствует случаю, когда осуществляется дискретная частотная манипуляция (ДЧМн) без разрыва фазы (φ_{ki} кратна 2π) и провалов до нуля в амплитудном спектре.



а)



б)

Рис. 8. Спектральная плотность и корреляционная функция последовательно-параллельного ШПС с ДЧМн

1.2. МЕТОДЫ ОБРАБОТКИ ШПС

Устройства обработки широкополосных сигналов строятся либо на основе корреляторов, либо с использованием СФ.

Корреляционная обработка ШПС осуществляется в два этапа:

1. Устраняется широкополосная модуляция (манипуляция при помощи ПСП или скачкообразное изменение частоты). При этом ШПС преобразуется в узкополосный сигнал, ширина спектра которого определяется модулирующим информационным сообщением.

2. Выделяется содержащаяся в сигнале информация. При этом используются обычные способы демодуляции УПС (ЧМн, ФМн и т.п.).

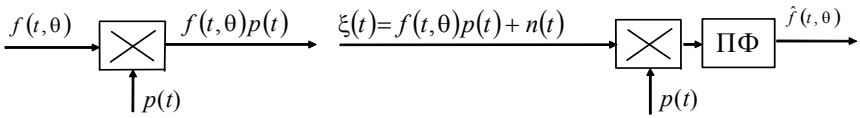
На первом этапе широкополосная модуляция снимается при помощи умножения принимаемого ШПС на опорный сигнал, формируемый в приёмнике и имеющий ту же структуру, что и обрабатываемый сигнал. При этом опорный и принимаемый сигналы должны быть синхронизированы. В процессе перемножения осуществляется преобразование спектра принимаемого ШПС, приводящее к появлению спектральных составляющих УПС, которые в последующем и выделяются.

На втором этапе осуществляется демодуляция УПС.

По способу построения корреляторы, реализующие первый этап обработки, делятся на *корреляторы прямого преобразования* и *корреляторы с преобразованием частоты*. На рисунке 9 приведена схема модема на основе корреляционной обработки сигнала. На рисунке 9, *а* представлен фазовый модулятор, формирующий ШПС. На рисунке 9, *б* – коррелятор прямого преобразования приёмника. При передаче ШПС формируется путём умножения функции $f(t,\theta)$, описывающей УПС и зависящей от времени t и информационного параметра $\theta(t)$, на некоторую расширяющую ПСП $p(t)$. При приёме на вход демодулятора подаётся наблюдение $\xi(t)$, включающее в себя непосредственно сигнальную составляющую $f(t,\theta)\,p(t)$ (ШПС) и помеху (шум) $n(t)$, которое в корреляторе перемножается на ПСП $p(t)$, идентичную использовавшейся в модуляторе ШПСС. На выходе полосового фильтра (ПФ), согласованного по полосе пропускания с шириной полосы частот спектра УПС, формируется оценка УПС $\hat{f}(t,\theta)$.

На рисунке 10 приведён пример процесса демодуляции при условии, что уровень помехи (шума) $n(t)$ мал и не оказывает существенного влияния на качество приёма, а проблема синхронизации $p(t)$ на передающей и приёмной стороне решена.

Характер преобразований спектральных плотностей сигнала и помехи в ШПСС, модулятор и демодулятор которой приведены на рис. 9, поясняется на рис. 11. Демодуляция ШПС приводит к модуляции помеховой составляющей $n(t)$ наблюдения $\xi(t)$ и, как следствие, расширению спектральной плотности помехи. При неизменной мощности помехи это приводит к уменьшению уровня её спектральной плотности $N_{\text{п}}$ относительно уровня спектральной плотности сигнала. Подобное преобразование объясняет увеличение отношения сигнал–шум (помеха) в полосе частот УПС при обработке ШПС, принимаемого на фоне сосредоточенной по спектру помехи.



а) *б*)

Рис. 9. Преобразование сигнала в ШПСС с коррелятором прямого преобразования

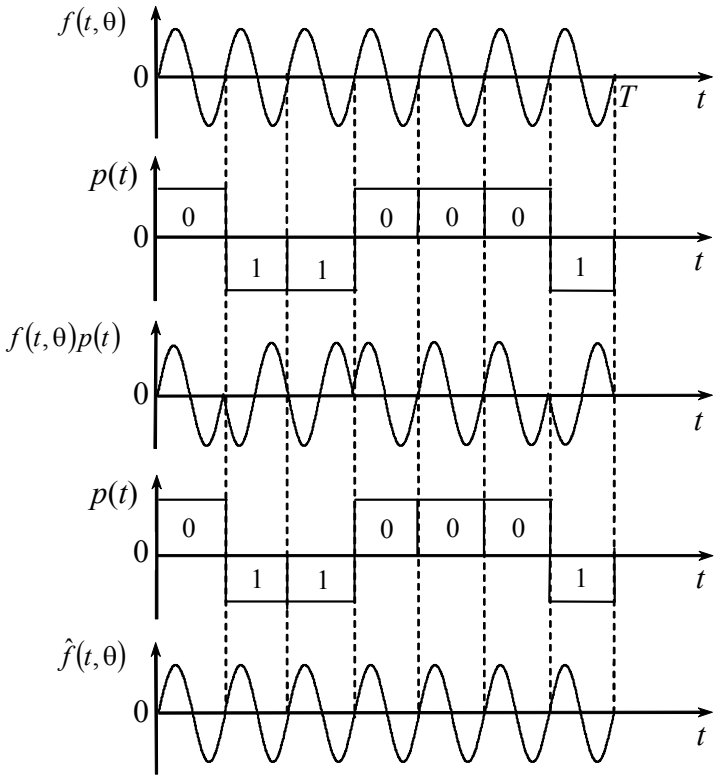


Рис. 10. Демодуляция последовательного ШПС с внутриимпульсной ФМн в корреляторе прямого преобразования

Из рисунка 11 понятно, что при корреляционной обработке ШПС осуществляется сжатие спектра сигнала.

Согласованный фильтр – это линейный фильтр, который полностью согласован с параметрами принимаемого сигнала и обеспечивает на выходе максимально возможное отношение сигнал–шум.

В отличие от корреляционной обработки, при согласованной фильтрации осуществляется сжатие сигнала не по спектру, а по времени. Сигнал на выходе приёмного устройства на СФ имеет примерно ту же ширину спектра, но длительность его уменьшается примерно на величину базы B (отклик СФ соответствует корреляционной функции ШПС, а ширина её главного лепестка в B раз уже, чем у корреляционной функции исходного УПС). При этом остаются справедливыми все отмеченные выше эффекты повышенной помехозащищённости, хотя они и объясняются через явление сжатия спектра, т.е. с позиций корреляционной обработки.

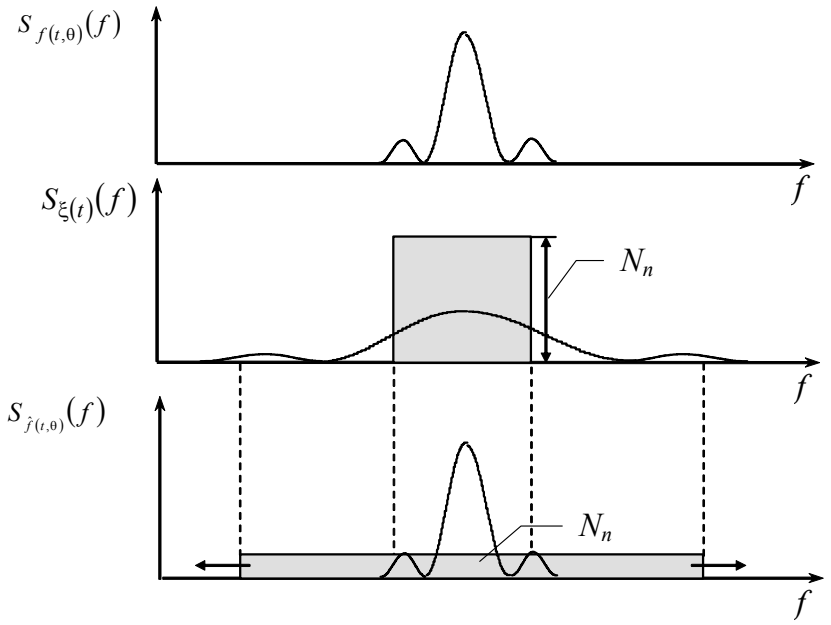


Рис. 11. Преобразование спектральных плотностей сигнала и помехи в ШПСС при корреляционной обработке

На рисунке 12 приведён вариант СФ для сигнала, приведённого на рис. 10.

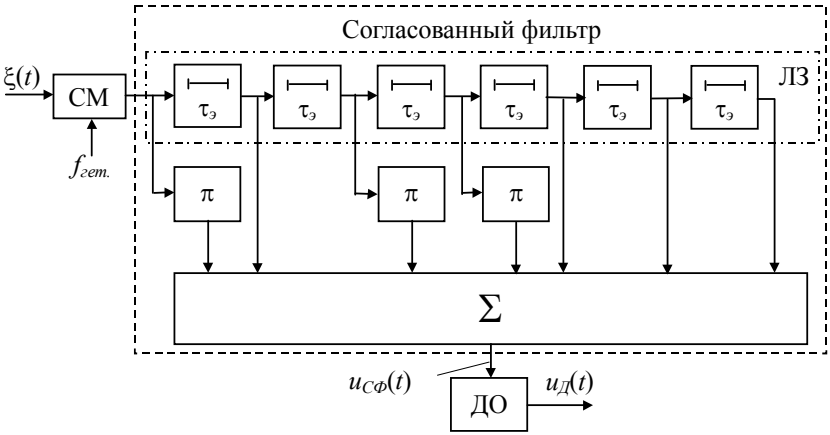


Рис. 12. Демодулятор последовательного ШПС с двоичной внутриимпульсной ФМн на основе согласованного фильтра

Смеситель при помощи частоты гетеродина $f_{\text{гет}}$ обеспечивает понижение частоты принимаемого сигнала до некоторого промежуточного значения, позволяющего существенно облегчить реализацию СФ. Включение элементов линии задержки (ЛЗ) и фазовращателей на π соответствует зеркальному отражению ШПС. Характер напряжений на выходе СФ $u_{\text{СФ}}(t)$ и детектора огибающей (ДО) $u_{\text{Д}}(t)$ приведены на рис. 13, *а* и *б*, соответственно. Напряжение $u_{\text{СФ}}(t)$ повторяет в масштабе реального времени корреляционную функцию принимаемого ШПС, а $u_{\text{Д}}(t)$ её огибающую. В момент $t = T$ напряжение на выходе ДО максимально.

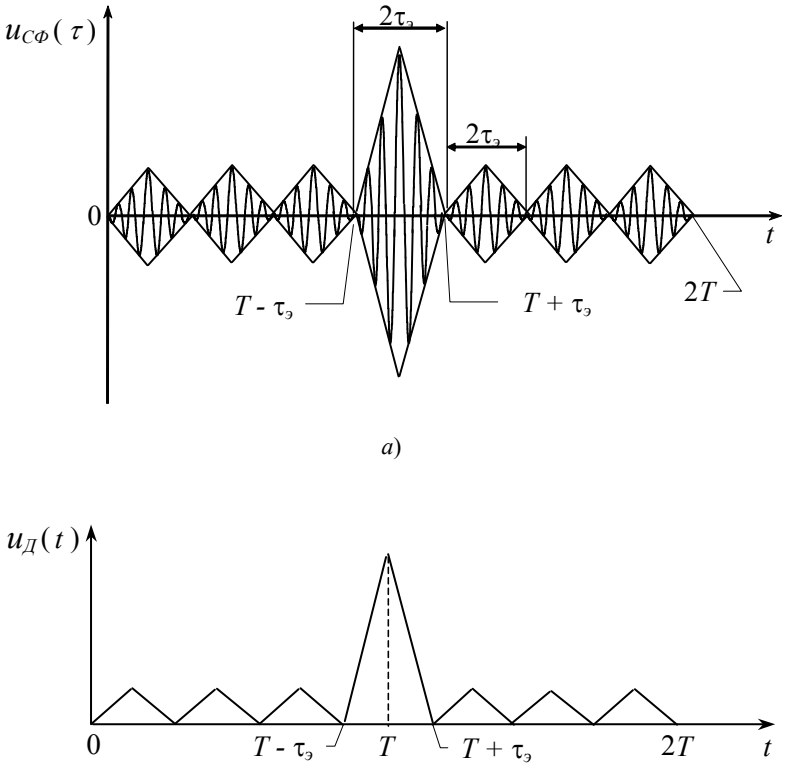


Рис. 13. Напряжения на выходе согласованного фильтра (*а*) и детектора огибающей (*б*)

Инвариантность СФ к времени прихода сигнала: во-первых, облегчает решение проблемы синхронизации при приёме ШПС; во-вторых, упрощает реализацию приёма многопозиционных сигналов (например, при использовании в канале связи кода с основанием $m > 2$). Эти обстоятельства во многом определяют эффективность применения согласованной фильтрации при обработке ШПС.

Контрольные вопросы

1. Дать определение широкополосного сигнала.
2. Дать определение широкополосной системы передачи информации.
3. Каким показателем характеризуется частотная избыточность сигнала?
4. Длительность элемента исходного узкополосного цифрового сигнала T = 5 мк·с. База широкополосного сигнала, сформированного из исходного, B = 100. Найти ширину спектра широкополосного сигнала $\Delta f_{\text{ш.пс}}$.
5. Используемая схема приёма ШПС – на основе коррелятора. Ширина спектра ШПС $\Delta f_{\text{ш.пс}}$ = 5 МГц. Ширина спектра помехи на входе $\Delta f_{\text{п}}$ = 2 МГц. Найти $\Delta f_{\text{п}}$ после снятия расширяющей последовательности.
6. Используемая схема приёма ШПС – на основе коррелятора. Ширина спектра ШПС $\Delta f_{\text{ш.пс}}$ = 5 МГц, длительность T = 1 мк·с. Ширина спектра помехи на входе умножителя $\Delta f_{\text{п}}$ = 2 МГц, отношение мощности сигнала к мощности помехи $\frac{P_{\text{с}}}{P_{\text{п}}} = 10$. Найти $\frac{P_{\text{с}}}{P_{\text{п}}}$ на выходе умножителя.

Задачи для самостоятельного решения

Задача 1. В таблице 1 приведены три расширяющие последовательности Хаффмена. На основе этих последовательностей сформировать широкополосные сигналы. Изобразить согласованный фильтр для одного из трёх сигналов. Изобразить отклики на выходе этого согласованного фильтра (СФ) для всех трёх сигналов. Сделать выводы о применимости ШПС для решения задачи кодового разделения каналов.

Таблица 1

№ п/п	p_1	p_2	p_3	p_4	p_5	p_6	p_7
1	–1	–1	1	1	1	1	1
2	1	–1	–1	1	1	1	–1
3	–1	1	1	1	–1	1	–1

Задача 2. Используя, полученный в задаче 1, СФ исследовать применимость ШПС в условиях многолучевого характера распространения сигнала в мобильных системах. Необходимо изобразить отклик на выходе СФ для сигнала на входе вида

$$S(t) = \sum_{0}^{3} S_i^{\text{ш.пс}}(t - iT,\ \gamma_i \times A)\ ,$$

где T – длительность ШПС; γ – коэффициент затухания по амплитуде в каждом луче; A – амплитуда излученного сигнала. Во сколько раз увеличится амплитуда отклика на выходе СФ, если $\gamma_0 = 0{,}1$; $\gamma_1 = 0{,}07$; $\gamma_2 = \gamma_3 = 0{,}05$?

Задача 3. Используя, полученный в задаче 1, СФ исследовать применимость ШПС в условиях многолучевого характера распространения сигнала в мобильных системах. Необходимо изобразить отклик на выходе СФ для сигнала на входе вида

$$S(t) = \sum_{0}^{3} S_i^{\text{ш.пс}}(t - iT,\ \gamma_i \times A)\ ,$$

где $\tau_э$ – длительность расширяющей последовательности ШПС; γ – коэффициент затухания по амплитуде в каждом луче; A – амплитуда излученного сигнала. Во сколько раз увеличится амплитуда отклика на выходе СФ, если $\gamma_0 = \gamma_1 = \gamma_2 = \gamma_3 = 0{,}7$?

МНОГОПОЗИЦИОННЫЕ СИГНАЛЫ И ЭФФЕКТИВНОСТЬ МОБИЛЬНЫХ СИСТЕМ

Цель: совершенствование теоретических знаний по основным технологиям мобильных сетей.

В результате выполнения практического занятия обучаемые *должны*:

– *знать* принципы применения многопозиционных сигналов в мобильных сетях;

– *уметь*: оценить потенциальную помехоустойчивость приёма многопозиционных сигналов в условиях помех; провести анализ эффективности применения многопозиционных сигналов в современных беспроводных сетях.

Проведение практического занятия включает три этапа:

1. Предварительная подготовка к занятию – проработка теоретического материала студентом на занятии и в ходе самостоятельной работы.
2. Основная часть – письменный опрос и решение задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные в работе вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [4, с. 21 – 31].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1. ОБЩИЕ СВЕДЕНИЯ О МНОГОПОЗИЦИОННЫХ СИГНАЛАХ

Наряду с бинарными сигналами в современных цифровых системах связи всё более востребованными становятся многопозиционные сигналы, использование которых позволяет:

1) повысить скорость передачи информации, по сравнению с двоичными сигналами, при заданной полосе частот. Платой за это является снижение помехоустойчивости приёма. Компромисс достигается применением помехоустойчивого кодирования;

2) обеспечить при одинаковой скорости передачи информации значительный выигрыш в энергетике по сравнению с двоичными сигналами. Платой за это является увеличение занимаемой полосы частот при той же скорости передачи информации;

3) реализовать многоканальные асинхронно-адресные системы связи.

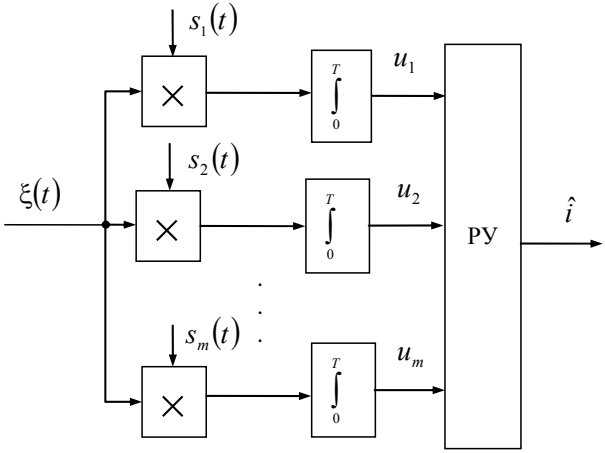


Рис. 1. Схема оптимального приемника *m* детерминированных сигналов

Оптимальный алгоритм приёма детерминированного многопозиционного сигнала $s_i(t),\ i = 1, 2, ..., m$ в условиях аддитивного БГШ, в предположении, что $m > 2$ и появление каждого из сигналов равновероятно, сводится к неравенству

$$\int_0^T \xi(t)\, s_i(t)\, dt - \frac{E_i}{2} > \int_0^T \xi(t)\, s_j(t)\, dt - \frac{E_j}{2},\ i \neq j. \qquad (1)$$

Схема оптимального корреляционного приёмника детерминированного многопозиционного сигнала приведена на рис. 1.

Помехоустойчивость приёма многопозиционных сигналов зависит от минимального расстояния между сигналами

$$d(s_i,\, s_j) = \left\{ \int_0^T \left[s_i(t) - s_j(t) \right]^2 dt \right\}^{\frac{1}{2}},\ i,\ j = 1, ..., m. \qquad (2)$$

Если сигналы имеют одинаковые энергии $E_i = E$, то

$$d(s_i,\, s_j) = \left[2E(1 - r_{ij}) \right]^{\frac{1}{2}},$$

где $r_{ij} = \frac{1}{E} \int_0^T s_i(t)\, s_j(t)\, dt$ – коэффициент взаимной корреляции сигналов $s_i(t)$ и $s_j(t)$. Для обеспечения одинаковой вероятности ошибки для

любого $s_i(t)$ необходимо, чтобы $r_{ij} = r_0$ для всех i и j, кроме $i \neq j$. Выражение для коэффициента взаимной корреляции найдём, исходя из неравенства

$$\int_0^T \left[\sum_{i=1}^{m} s_i(t) \right]^2 dt = \int_0^T \sum_{i=1}^{m} \sum_{j=1}^{m} s_i(t)\, s_j(t)\, dt = mE + r_0 E(m^2 - m) \geq 0 ,$$

решая которое, получим, что

$$r_0 \geq -\frac{1}{m-1}. \tag{3}$$

Для оптимальной системы

$$r_0 = -\frac{1}{m-1}. \tag{4}$$

Исходя из коэффициента взаимной корреляции r_{ij}, $i \neq j$, многопозиционные сигналы подразделяются на симплексные и ортогональные.

Симплексные сигналы – это сигналы, для которых

$$r_{ij} = \begin{cases} 1 \text{ при } i = j, \\ -1/(m-1), \text{ если } i \neq j. \end{cases} \tag{5}$$

Такие сигналы являются *эквидистантными*, т.е. для всех пар сигналов $s_i(t)$ и $s_j(t)$ расстояние $d(s_i, s_j)$ одинаково.

Ортогональные сигналы наиболее часто применяют на практике и для них

$$r_{ij} = \begin{cases} 1 \quad i = j, \\ 0, \text{ если } i \neq j. \end{cases} \tag{6}$$

Если все ортогональные сигналы имеют равную энергию, то они также эквидистантны. При больших значениях m помехоустойчивость приёма ортогональных сигналов близка к помехоустойчивости симплексных сигналов.

Помехоустойчивость приёма многопозиционных ортогональных сигналов. Вероятность правильного приёма i-го сигнала

$$P_{\text{пр}}(s_i) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left[-\frac{1}{2} \left(x - \sqrt{\frac{2E}{N_0}} \right)^2 \right] \Phi^{m-1}(x)\, dx , \tag{7}$$

где $\Phi(x)$ – интеграл вероятности.

Полная вероятность ошибки при различении m равновероятных сигналов, т.е. $P(s_1) = P(s_2) = ... = P(s_m) = 1/m$, определяется выражением

$$P_e = \left[1 - P_{\text{пр}}(s_1)\right]P(s_1) + ... + \left[1 - P_{\text{пр}}(s_m)\right]P(s_m) =$$

$$= (1/m)\left[m - \sum_{i=1}^{m} P_{\text{пр}}(s_i)\right] = 1 - P_{\text{пр}}(s_i). \qquad (8)$$

При сравнении систем необходимо иметь в виду, что при фиксированной длительности элемента сигнала T каждый равновероятный m–ичный сигнал несёт в $\log_2 m$ раз больше количества информации, чем двоичный сигнал. На рисунке 2 приведены кривые помехоустойчивости когерентного приёма ортогональных сигналов при m = 2, 4, 32 и 256. Для сравнения приведена кривая, характеризующая помехоустойчивость когерентного приёма детерминированного бинарного ФМн радиосигнала. Здесь $E_{\text{В}} = E/\log_2 m$ – энергия, затрачиваемая на один бит информации.

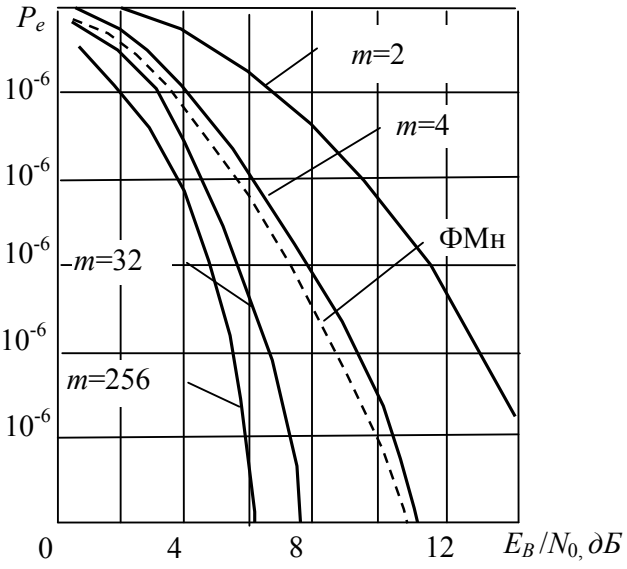


Рис. 2. Зависимость вероятности ошибки от отношения E_B/N_0 при оптимальном приёме m детерминированных ортогональных сигналов

1.2. ВЫБОР И ФОРМИРОВАНИЕ МНОГОПОЗИЦИОННЫХ СИГНАЛОВ

Многопозиционные сигналы могут отличаться по амплитуде, фазе, частоте или могут зависеть от двух и более параметров сигнала. *Многопозиционный с амплитудной манипуляцией* (АМн) сигнал $s_i(t)$ отличается от бинарного количеством дискретных уровней амплитуды сигнала A_i:

$$s_i(t) = A_i(t)\cos\left[\omega_0 t + \varphi(t),\ i = 1,\ 2,\ ...,\ m.\right]$$

При демодуляции каждому уровню A_i сопоставляется кодовая комбинация (слово) двоичного цифрового сигнала длиной k, и, следовательно, целесообразно выбирать $m = 2^k$.

На рисунке 3 даны соответствующие пространственные диаграммы сигналов для $m = 2$, $m = 4$ и $m = 8$.

Отображение возможных амплитуд целесообразно выполнять таким образом, чтобы соседние кодовые комбинации отличались только в одном разряде. Это важно при демодуляции сигнала, поскольку наиболее вероятные ошибки вызывает ошибочный выбор амплитуды, соседней по отношению к той, которая действительно передана. В этом случае, в k-битовой информационной последовательности возникает ошибка только в одном бите. Такое отображение называется кодом Грея.

Многопозиционные АМн сигналы используются редко, что объясняется их низкой помехоустойчивостью.

Чаще применяют *многопозиционную частотную манипуляцию* (ЧМн). Сигнал в этом случае может быть записан в виде

$$s_i(t) = A_0\cos\left[\omega_i(t)\,t + \varphi(t)\right],\ i = 1,\ 2,\ ...,\ m.$$

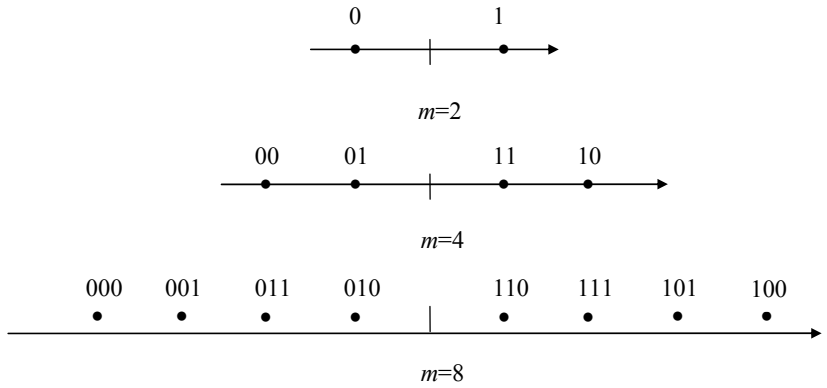


Рис. 3. Пространственная диаграмма многопозиционных цифровых сигналов с амплитудной манипуляцией

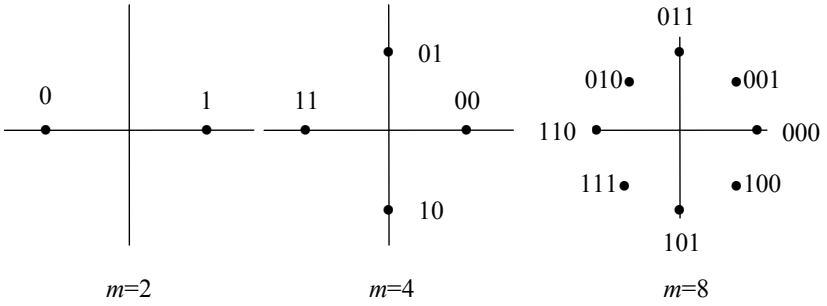


Рис. 4. Пространственная диаграмма сигналов с многопозиционной цифровой фазовой манипуляцией

Ещё более широкое применение находят сигналы с *многопозиционной фазовой манипуляцией* (ФМн)

$$s_i(t) = A_0 \cos\left[\omega_0 t + 2\pi(i-1)/m\right].$$

Пространственные диаграммы сигналов с фазовой манипуляцией для $m = 2, 4, 8$ приведены на рис. 4.

Как и в случае АМн, отображение или задание k информационных бит можно выполнить различными путями, предпочтительнее пользоваться кодом Грея.

Квадратурная амплитудная манипуляция (КАМ) обеспечивает хорошую частотную эффективность. Сигнал может быть записан в виде

$$s_i(t) = A_{ic}(t)\cos\omega_0 t - A_{is}(t)\sin\omega_0 t,$$

где A_{ic} и A_{is} – квадратурные информационные составляющие.

Или же

$$s_i(t) = U_i(t)\cos(\omega_0 t + \psi_i),$$

где $U_i(t) = \sqrt{A_{ic}^2 + A_{is}^2}$ и $\psi_i = \operatorname{arctg}(A_{is} / A_{ic})$, т.е. сигнал с КАМ можно рассматривать как комбинацию амплитудной и фазовой манипуляций. Образуем комбинацию m_1 уровневой АМн и m_2 позиционной ФМн таким образом, чтобы сигнальное созвездие имело $m = m_1 m_2$ точек пространства. Если $m_1 = 2^n$ и $m_2 = 2^i$, то сигнальное созвездие комбинированной АМн–ФМн сводится к мгновенной передаче $\log m_1 m_2$ двоичных символов, возникающих со скоростью $R/\log m_1 m_2$, где $R = \log_2 k$; k – количество элементов в кодовой комбинации первичного кода. Примеры сигнальных пространственных диаграмм КАМ для $m = 8$ и $m = 16$ показаны на рис. 5.

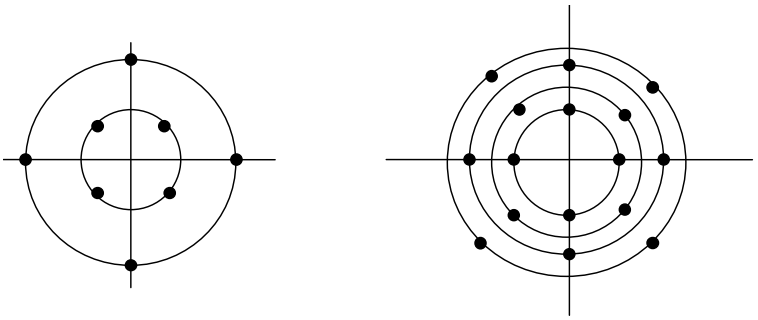


Рис. 5. Примеры пространственных диаграмм для КАМ

При *выборе ансамбля сигналов* необходимо иметь в виду, что многопозиционные сигналы можно разделить на два класса. К одному из них принадлежат сигналы, для которых характерно то, что с увеличением объёма ансамбля m растёт энергетическая эффективность, но при этом расширяется полоса частот, занимаемая сигналом (снижается частотная эффективность). К этому классу относятся ортогональные, биортогональные и симплексные сигналы. При $m >> 1$ они обеспечивают практически одинаковую помехоустойчивость и являются наилучшими. В то же время их полосы частот, по сравнению с двоичными сигналами, шире соответственно в $m/\log_2 m$ раз при той же скорости передачи информации.

К другому классу принадлежат сигналы, для которых характерно то, что с увеличением объёма ансамбля m расстояние между сигналами уменьшается (снижается энергетическая эффективность), а полоса частот, занимаемая сигналами, не увеличивается. К этому классу относятся сигналы с КАМ.

1.3. ЭФФЕКТИВНОСТЬ СИСТЕМ ПЕРЕДАЧИ ИНФОРМАЦИИ

Эффективность может быть повышена на основе наиболее совершенных способов передачи (кодирования и модуляции (в том числе и выбор сигнала)) и приёма (демодуляции и декодирования) сообщений, а также сокращения избыточности источника (сжатия данных).

Наиболее известны следующие виды эффективности систем передачи информации:

1) *информационная эффективность*

$$\eta = R / C ; \tag{9}$$

2) *частотная эффективность*

$$\gamma = R / \Delta f_s ; \tag{10}$$

3) *энергетическая эффективность*

$$\beta = R/(P_s / N_0). \tag{11}$$

Здесь R – скорость передачи информации; C – пропускная способность канала; Δf_s – полоса частот канала; P_s – мощность сигнала; N_0 – интенсивность спектральной плотности помехи в канале.

В принципе достаточно ограничиться γ и β – показателями. Зависимость между характеристиками эффективности β и γ :

$$\beta = \gamma/\left(2^{\gamma} - 1\right). \tag{12}$$

Следует заметить, что γ – эффективность может изменяться от 0 до $2\log m$ при передаче цифры, в то время как β – эффективность ограничена сверху при $\gamma \to 0$ $\beta_{\max} = 1/\ln 2$.

В реальных системах ошибка всегда имеет конечное значение $\eta < 1$. В этих случаях при заданной допустимой вероятности ошибки можно определить отдельно γ и β и построить кривые $\beta = f(\gamma)$ при P_e = const. На рисунке 6 представлены кривые для ортогональных (ОС), биортогональных (БС) и ФМн сигналов для различного числа сигналов M и вероятности ошибки $P_e = 10^{-5}$. Ход этих кривых зависит от вида сигнала, кода и способа обработки сигнала.

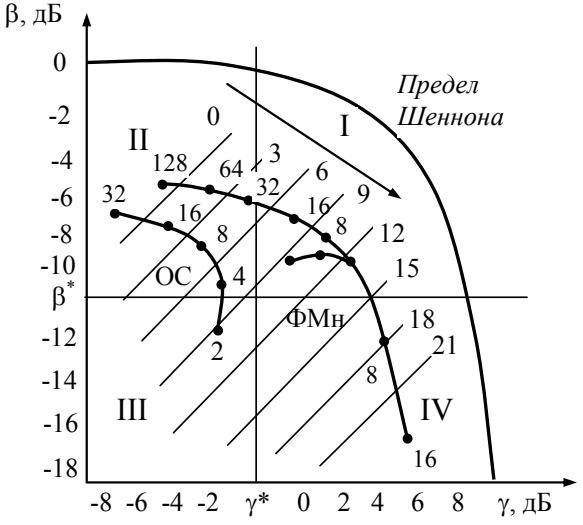


Рис. 6. Зависимости $\beta = f(\gamma)$ для ОС, БС и ФМн сигналов для различного числа сигналов M и вероятности ошибки $P_e = 10^{-5}$

Полученные таким образом результаты позволяют выбрать системы, удовлетворяющие заданным требованиям. Например, заданы скорость передачи информации R^*, полоса частот канала Δf_s и отношение сигнал – шум q. Тогда область возможных значений β и γ можно разбить на четыре квадранта. Системы, расположенные в квадранте I, удовлетворяют требованиям по обоим показателям: $\beta > \beta^*$ и $\gamma > \gamma^*$. Системы, расположенные в квадранте II удовлетворяют требованиям только по β, а системы квадранта IV удовлетворяют требованиям только по γ. Системы квадранта III не удовлетворяют требованиям по обоим показателям: $\beta < \beta^*$ и $\gamma < \gamma^*$.

Возможные СПИ можно условно разделить на две группы: системы с высокой β-эффективностью (но малой γ) и системы с высокой γ-эффективностью (и соответственно малой β). К первой группе относятся системы, в которых первостепенное значение имеют энергетические показатели, в частности, космические и спутниковые системы связи. В системах проводной связи важнейшим показателем является γ-эффективность. Полезными могут оказаться сравнения с идеальной системой, в нашем случае с пределом Шеннона:

$$\Delta\beta = \beta - \beta_и, \quad \Delta\gamma = \gamma - \gamma_и.$$

В системах с помехоустойчивым кодированием β и γ характеризуют не только модем, но и кодек. Поясним это, для чего запишем выражения (12) и (13) следующим образом:

$$\beta = \frac{R}{E/N_0} = \frac{R_s R_k}{E_s/N_0}, \quad \gamma = \frac{R_s R_k}{TF}, \tag{13}$$

где $E = P_s\tau$ – энергия, необходимая для передачи одной двоичной единицы (бита) информации; $R_s = RT$ – скорость передачи информации в битах на один сигнал длительностью T. Эта скорость полностью определяется системой модуляции. Скорость (относительная) кода

$$R_k = \log M / N = (1 - \chi_k)\log m$$

определяется видом выбранного кода и его избыточностью χ_k (при двоичном кодировании $M = 2^k$, $R_k = K/N = 1 - \chi_k$, где K – объём алфавита

дискретного источника; N – длина кодовой комбинации). Энергия сигнала при использовании кода со скоростью R_k при фиксированных R и P_s, очевидно, будет равна $E_s = P_s T R_k = E \log M$.

Связь между полосой пропускания линий и её *максимально возможной пропускной способностью*, вне зависимости от принятого способа физического кодирования, установил Клод Шеннон:

$$C = \Delta f_{\text{пплс}} \log_2 \left(1 + \frac{P_\text{с}}{P_\text{ш}} \right), \qquad (14)$$

где C – максимальная пропускная способность линии в битах в секунду (бит/с); $\Delta f_{\text{пплс}}$ – полоса пропускания линии в герцах; $P_\text{с}$ – мощность сигнала; $P_\text{ш}$ – мощность шума (помехи).

Близким, по сути к формуле Шеннона является соотношение, полученное Найквистом, которое определяет максимально возможную пропускную способность линии связи, по без учёта шума на линии:

$$C = 2\Delta f_{\text{пплс}} \log_2 m \,, \qquad (15)$$

где m – количество различных состояний информационного параметра сигнала (основание кода в канале).

Контрольные вопросы

1. Дать определение многопозиционного сигнала.
2. Назвать области применения многопозиционных сигналов.
3. Записать выражения для многопозиционных АМн, ЧМн и ФМн сигналов.
4. Почему многопозиционные АМн сигналы не нашли широкого применения?
5. Чем обусловлено применение кода Грея в многопозиционных сигналах.
6. Достоинство и недостатки многопозиционного сигнала с КАМ?
7. Достоинства и недостатки многопозиционного ФМн?
8. Перечислить основные показатели эффективности системы передачи информации, пояснить необходимость их оценки.
9. Какую роль играет оценка эффективности при обосновании выбора сигнала в проводных и беспроводных сетях?
10. Пояснить, почему многопозиционные сигналы позволяют повысить скорость передачи информации, по сравнению с двоичными сигналами, при заданной полосе частот.

Задачи для самостоятельного решения

Задача 1. Выполнить анализ радиоконфигурации прямого канала трафика cdma 2000, приведённого в табл. 1.

Таблица 1

RC	SR	Вид модуляции и скорость кода R	Поддерживаемые скорости передачи данных, кбит/с
RC1	1	BPSK; $R = 1/2$	1.2, 2.4, 4.8, 9.6
RC2	1	BPSK; $R = 1/2$	1.8, 3.6, 7.2, 14.4
RC3	1	QPSK; $R = 1/4$	1.5, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6
RC4	1	QPSK; $R = 1/2$	1.5, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6, 307.2
RC5	1	QPSK; $R = 1/4$	1.8, 3.6, 7.2, 14.4, 28.8, 57.6, 115.2, 230.4
RC6	3	QPSK; $R = 1/6$	1.5, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6, 307.2
RC7	3	QPSK; $R = 1/3$	1.5, 2.7, 4.8, 9.6, 19.2, 38.4, 76.8, 153.6, 307.2, 614.4
RC8	3	QPSK; $R = 1/4, 1/3$	1.8, 3.6, 7.2, 14.4, 28.8, 57.6, 115.2, 230.4, 460.8
RC9	3	QPSK; $R = 1/2, 1/3$	1.8, 3.6, 7.2, 14.4, 28.8, 57.6, 115.2, 230.4, 460.8, 1036.8
RC10	1	QPSK; 8- PSK;16-QAM; $R = 1/5$	43.2, 81.6, ……..,931.2, ……..,3091.2

Здесь RC (Radio Configuration) – набор параметров модуляции, расширения спектра, схемы кодирования, формата кадра и чипсовой скорости); SR (Spreading Rate) – форма чиповой скорости (число несущих); BPSK (Binary Phase Shift Keying) – двоичная фазовая манипуляция, QPSK (Quadrature Phase Shift Keying) – квадратурная фазовая манипуляция, PSK (Phase Shift Keying) – фазовая манипуляция, QAM (Quadrature – Amplitude Modulation) – квадратурная амплитудная модуляция.

Часть II
ОСНОВЫ МАРШРУТИЗАЦИИ И КОММУТАЦИИ

ИСПОЛЬЗУЕМЫЕ В ПОСОБИИ ПИКТОГРАММЫ

Маршрутизатор

Коммутатор

Сервер

Ноутбук

ПК

Сеть

Последовательный канал

Линия связи

СОГЛАШЕНИЕ О СИНТАКСИСЕ КОМАНД

Синтаксис приводимых в пособии команд конфигурации идентичен обозначениям, используемым в «Справочнике по командам Cisco IOS 15.2» («Cisco IOS 15.2 Command Reference»).

Команды и ключевые слова, которые должны вводиться без изменений, выделены **полужирным**.

Аргументы, значения которых изменяются, выделены *курсивом*.

Вертикальная черта | используется для разделения альтернативных взаимоисключающих значений команд.

Квадратные скобки [ ] обозначают необязательные элементы команды.

Фигурные скобки { } указывают на необходимость выбора из указанных в скобках параметров.

Фигурные скобки внутри квадратных [{ }] означают необходимость выбора среди возможных значений необязательных элементов.

Практическое занятие 1

МОДЕЛЬ OSI

Цель: изучить процесс передачи данных в сети и работу протоколов уровней модели OSI.

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: характеристики, функции и назначение уровней модели OSI; особенности процесса инкапсуляции и деинкапсуляции; особенности процесса обмена данными между узлами; структуры заголовков PDU уровней модели OSI; принципы работы сетевой утилиты Scapy;

– *уметь* работать с сетевой утилитой Scapy.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 58 – 86].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Эталонная модель Open Systems Interconnection (OSI) описывает передачу данных по сети. Модель имеет семь уровней (рис. 1). Каждый уровень модели OSI выполняет определённые функции и поддерживает соответствующее программное обеспечение и устройства.

Уровень 1: физический уровень. На физическом уровне определены электрические, механические, процедурные и функциональные характеристики активации, поддержания и отключения физического канала между конечными системами. Технические характеристики физического уровня определяют такие параметры, как уровни напряжения, синхронизацию изменений напряжения, физическую скорость передачи данных, максимальное расстояние передачи данных, физические подключения и другие аналогичные характеристики.

Уровень 2: канальный уровень. На канальном уровне определяется формат данных для передачи и методы контроля доступа к физическим средам. Этот уровень также включает функции обнаружения и коррекции ошибок для обеспечения надёжной передачи данных.

Уровень 3: сетевой уровень. Сетевой уровень обеспечивает связь и выбор пути между двумя хостами, которые могут находиться в сетях, географически удалённых друг от друга.

Уровень 4: транспортный уровень. На транспортном уровне выполнятся сегментация данных от передающего хоста и реорганизация данных в поток данных в принимающем хосте. Транспортный уровень скрывает детали передачи данных от верхних уровней. В частности, на транспортном уровне решаются задачи, связанные с надёжностью передачи данных между двумя хостами. В рамках реализации службы обмена данными транспортный уровень создаёт, поддерживает и корректно завершает виртуальные каналы. Функции обнаружения и коррекции ошибок, а также управление потоками данных, обеспечивают надёжность служб.

Рис. 1. Эталонная модель OSI

Уровень 5: сеансовый уровень. На сеансовом уровне выполняется создание, управление и завершение сеансов между двумя хостами, обменивающимися данными. Кроме того, на сеансовом уровне выполняется синхронизация диалога между представительскими уровнями двух хостов и управление обменом данными между ними. Кроме управления сеансами сеансовый уровень предлагает дополнительные средства для эффективной передачи данных – классы обслуживания (Class of Service) и событийная отчётность о проблемах сеансового, прикладного и представительского уровней.

Уровень 6: уровень представления. Уровень представления гарантирует, что сведения, передаваемые на прикладном уровне одной системы, могут быть распознаны на прикладном уровне другой системы. При необходимости на уровне представления выполняется перевод из одного формата данных в другой.

Уровень 7: прикладной уровень. Прикладной уровень OSI находится ближе всего к пользователю. На этом уровне предоставляются сетевые услуги для пользовательских приложений. Прикладной

уровень отличается от остальных уровней тем, что он не предоставляет услуг другим уровням OSI, а только приложениям вне модели OSI. Прикладной уровень обеспечивает доступность целевых партнёров по соединению, а также синхронизирует и формирует соглашения по процедурам коррекции ошибок и контроля целостности данных.

Инкапсуляция и деинкапсуляция. Данные, передаваемые по сети, должны пройти процесс преобразования как на передающей, так и на принимающей системе. Этот процесс преобразования называется *инкапсуляцией* и *деинкапсуляцией* данных.

Информация, передаваемая по сети, называется данными. При передаче данных с одного компьютера на другой необходимо упаковать данные в фрагменты. Этот процесс называется инкапсуляцией. Во время инкапсуляции данные разбиваются на фрагменты с необходимыми данными протокола перед передачей по сети. При перемещении данных через уровни модели OSI на каждом уровне к данным добавляется *заголовок* (и *концевая метка*, если необходимо), пока данные не доходят до самого низкого уровня (рис. 2).

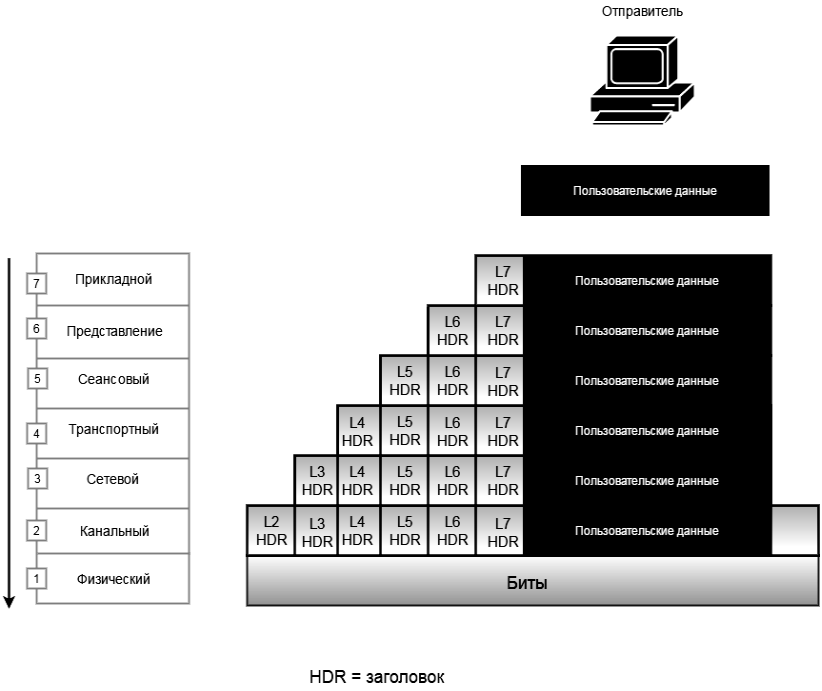


Рис. 2. Инкапсуляция данных

Заголовки и концевые метки содержат контрольную информацию для сетевых устройств, которая обеспечивает корректную доставку данных и их правильную интерпретацию на приёмнике.

Когда удалённое устройство получает последовательность бит, его *физический уровень* передаёт биты данных на вышележащие уровни для обработки. Этот процесс называется *деинкапсуляцией*.

Чтобы пакеты данных могли передаваться от источника к месту назначения, каждый уровень модели OSI источника должен обмениваться данными с соответствующим уровнем получателя (рис. 3).

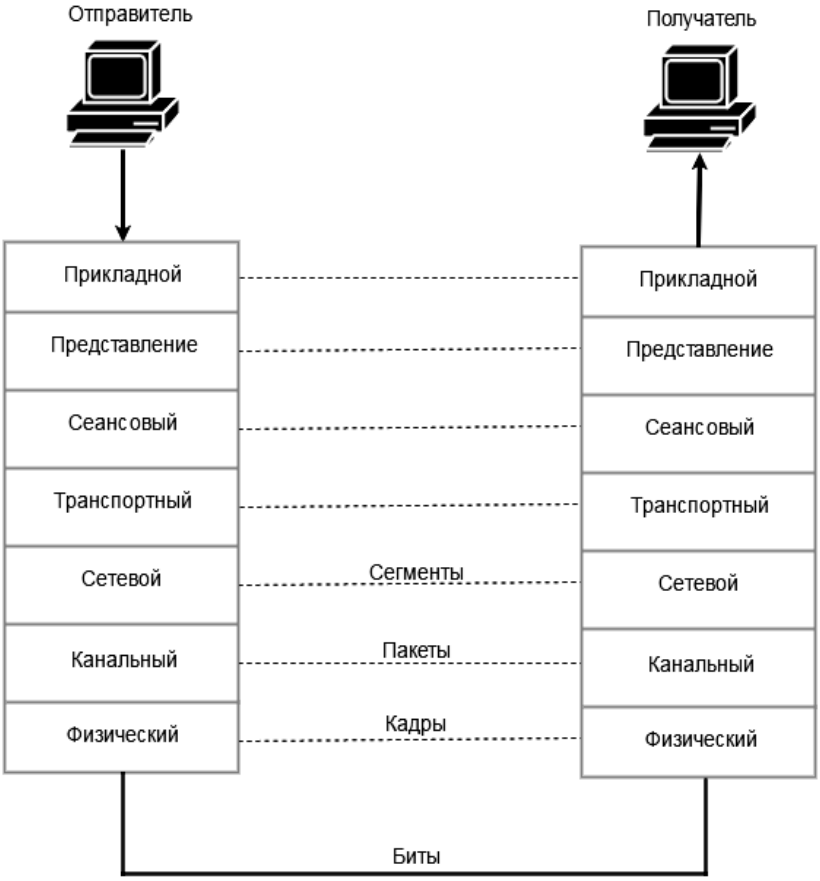


Рис. 3. Типы PDU

При обмене данными между узлами *протоколы* на каждом уровне обмениваются пакетами данных, которые называются «элементами информации протокола» (*Protocol Data Unit*, *PDU*). Эти фрагменты данных создаются на источнике в сети, а затем передаются к месту назначения. Предоставление услуг на каждом уровне зависит от низлежащего уровня OSI. Для выполнения своих функций более низкий уровень использует инкапсуляцию для размещения элементов PDU верхнего уровня в поле данных нижнего уровня. После этого на каждом уровне добавляются заголовки, которые необходимы этому уровню для выполнения своих функций. На каждом уровне модели OSI PDU имеют собственное название: на транспортном – сегменты (для протокола UDP – датаграммы), на сетевом – пакеты, на канальном – кадры.

TCP-заголовок. Структура TCP-заголовка представлена на рис. 4.

Порт источника **–** в этом поле указывается номер порта отправителя. Предполагается, что это значение задаёт порт, на который при необходимости будет посылаться ответ. В противном же случае, значение должно быть равным 0.

Порт получателя **–** это поле обязательно и содержит порт получателя.

Номер последовательности – номер последовательности первого байта в сегменте, обеспечивает правильную последовательность поступающих данных (32 бита).

Номер подтверждения – следующий ожидаемый байт по протоколу TCP (32 бита).

Длина заголовка – количество 32-битных слов в заголовке (4 бита).

Октет	**0…3**	**4…9**	**10…15**	**16…31**
0	Порт источника			Порт получателя
4	Номер последовательности			
8	Номер подтверждения			
12	Длина заголовка	Зарезер-вировано	Флаги	Размер Окна
16	Контрольная сумма			Указатель важности
20	Параметры			
24+	Данные			

Рис. 4. Структура TCP-заголовка

Флаги – битовые поля, управляют функциями настройки, контролем перегрузки в сети и прекращением сеанса (6 бит):

1) *URG* – флаг указателя важности;

2) *ACK* – флаг номера подтверждения;

3) *PSH* – флаг инструктирует получателя протолкнуть данные, накопившиеся в приёмном буфере, в приложение пользователя;

4) *RST* – флаг очистки буфера;

5) *SYN* – синхронизация номеров последовательности;

6) *FIN* – флаг завершения соединения.

Размер окна – число байт данных, которые устройство ожидает принять (16 бит).

Контрольная сумма – рассчитанная контрольная сумма заголовка и данных (16 бит).

Указатель важности – это поле указывает на конец срочных данных (16 бит).

Параметры – могут применяться в некоторых случаях для расширения протокола. Иногда используются для тестирования.

Данные – данные протокола верхнего уровня (размер не фиксирован).

Заголовок UDP-датаграммы. Заголовок UDP-датаграммы имеет вид, представленный на рис. 5.

Октет	**0…15**	**16…31**
0	Порт отправителя	Порт получателя
4	Длина датаграммы	Контрольная сумма
8+	Данные	

Рис. 5. Структура заголовка UDP-датаграммы

Порт отправителя **–** в этом поле указывается номер порта отправителя. Предполагается, что это значение задаёт порт, на который при необходимости будет посылаться ответ. В противном же случае значение должно быть равным 0 (16 бит).

Порт получателя **–** это поле обязательно и содержит порт получателя (16 бит).

Длина датаграммы **–** поле, задающее длину всей датаграммы (заголовка и данных) в байтах. Минимальная длина равна длине заголовка – 8 байт. Теоретически, максимальный размер поля – 65 535 байт для UDP-датаграммы (8 байт на заголовок и 65 527 на данные). Факти-

ческий предел для длины данных при использовании IPv4 – 65 507 (помимо 8 байт на UDP-заголовок требуется ещё 20 на IP-заголовок).

Заголовок IPv4. Заголовок IP-пакета представлен на рис. 6.

Октет	**0…3**	**4…7**	**8…15**	**16…18**	**19…31**
0	Версия	IHL	Тип обслуживания	Длина пакета	
4	Идентификатор			Флаги	Смещение фрагмента
8	Время жизни (TTL)		Протокол	Контрольная сумма заголовка	
12	IP-адрес отправителя				
16	IP-адрес получателя				
20	Параметры (от 0 до 10 32-битных слов)				
24+	Данные				

Рис. 6. Структура заголовка IP-пакета

Версия – для IPv4 значение поля должно быть равно 4.

Internet Header Length (*IHL*) – длина заголовка IP-пакета в 32-битных словах (dword). Именно это поле указывает на начало блока данных (англ. *payload* – полезный груз) в пакете. Минимальное корректное значение для этого поля равно 5.

Длина пакета – длина пакета в октетах, включая заголовок и данные. Минимальное корректное значение для этого поля равно 20, максимальное – 65 535.

Идентификатор – значение, назначаемое отправителем пакета и предназначенное для определения корректной последовательности фрагментов при сборке пакета. Для фрагментированного пакета все фрагменты имеют одинаковый идентификатор.

3 бита флагов. Первый бит должен быть всегда равен нулю, второй бит DF (Don’t Fragment) определяет возможность фрагментации пакета и третий бит MF (More Fragments) показывает, не является ли этот пакет последним в цепочке пакетов.

Смещение фрагмента – значение, определяющее позицию фрагмента в потоке данных. Смещение задаётся количеством 8-байтовых блоков, поэтому это значение требует умножения на 8 для перевода в байты.

Время жизни (*Time To Life*, *TTL*) – число маршрутизаторов, которые может пройти этот пакет. При прохождении маршрутизатора это число уменьшается на единицу. Если значение этого поля равно нулю, то пакет должен быть отброшен, и отправителю пакета может быть послано сообщение *Time Exceeded* (ICMP тип 11 код 0).

Протокол – идентификатор интернет-протокола следующего уровня указывает, данные какого протокола содержит пакет, например, TCP или ICMP.

Контрольная сумма заголовка – вычисляется в соответствии с RFC 1071.

IP-адрес отправителя – имеет размер 32 бита.

IP-адрес получателя – имеет размер 32 бита.

Параметры – является необязательным и используется обычно только при отладке сети.

Заголовок Ethernet. Заголовок кадра Ethernet представлен на рис. 7.

Октет	**7**	**1**	**6**	**6**	**2**	**…**	**4**	**1**
	Преамбула	Разделитель начала кадры	Адрес назначения	Адрес отправителя	Протокол или тип	Данные	Контрольная сумма	Разделитель конца кадра

Рис. 7. Структура кадра Ethernet

Преамбула состоит из семи байтов синхронизирующих данных. Каждый байт содержит одну и ту же последовательность битов – 10101010.

Разделитель начала кадра состоит из одного байта с набором битов 10101011. Появление этой комбинации является указанием на предстоящий приём кадра.

Поле данных может содержать от 46 до 1500 байт данных. Если длина поля меньше 46 байт, то используется заполнение. Это обеспечивает корректную работу механизма обнаружения коллизий.

Адрес назначения – 6-байтовое поле, содержащее MAC-адрес станции получателя.

Адрес отправителя – 6-байтовое поле, содержащее MAC-адрес станции отправителя.

Контрольная последовательность кадра (*контрольная сумма пакета CRC*) – 4 байта, содержащие значение, которое вычисляется по определённому алгоритму (полиному CRC-32)

Разделитель конца кадра – появление этой комбинации является указанием на предстоящее окончание кадра.

Scapy. Scapy – сетевая утилита, написанная на языке Python, которая позволяет посылать, просматривать и анализировать любые сетевые пакеты.

Этапы установки утилиты:

1. Установить интерпретатор языка Python версии 2.6.3. После установки необходимо добавить переменную окружения PATH для текущего пользователя, указав в качестве её значения рабочую директорию языка Python и директорию, в которой расположены рабочие скрипты: C:\Python26;C:\Python26\Scripts.

2. Установить стабильную версию утилиты Scapy, распаковать архив, в директории Scapy, вызвать командную строку и выполнить установку командой «python setup.py install».

3. Установить библиотеки pywin32, WinPcap, pypcap, libdnet и pyreadline.

Работа с утилитой. Для просмотра списка поддерживаемых протоколов используется функция *ls*().

Утилита поддерживает более 300 разнообразных протоколов, включая прикладные, например, HTTP, транспортные TCP и UDP, сетевого уровня IPv4 и IPv6 и канального уровня Ethernet. Важно обращать внимание на регистр: большинство протоколов пишутся в Scapy с использованием заглавных букв.

Для того чтобы посмотреть поля определённого протокола, необходимо вызвать функцию *ls*() с указанием протокола, например, *ls*(*TCP*):

```
>>> ls(TCP)
sport      : ShortEnumField       = (20)
dport      : ShortEnumField       = (80)
seq        : IntField             = (0)
ack        : IntField             = (0)
dataofs    : BitField             = (None)
reserved   : BitField             = (0)
flags      : FlagsField           = (2)
window     : ShortField           = (8192)
chksum     : XShortField          = (None)
urgptr     : ShortField           = (0)
options    : TCPOptionsField      = ({})
>>>
```

Обратите внимание на то, что порядок флагов в заголовке TCP-фрагмента в Scapy иной, нежели общепринятый:

```
>>> flags = {
    'F': 'FIN',
    'S': 'SYN',
    'R': 'RST',
    'P': 'PSH',
    'A': 'ACK',
    'U': 'URG',
    'E': 'ECE',
    'C': 'CWR',
}

>>> ls(UDP)
sport      : ShortEnumField       = (53)
dport      : ShortEnumField       = (53)
len        : ShortField           = (None)
chksum     : XShortField          = (None)
>>>

>>> ls(IP)
version    : BitField             = (4)
ihl        : BitField             = (None)
tos        : XByteField           = (0)
len        : ShortField           = (None)
id         : ShortField           = (1)
flags      : FlagsField           = (0)
frag       : BitField             = (0)
ttl        : ByteField            = (64)
proto      : ByteEnumField        = (0)
chksum     : XShortField          = (None)
src        : Emph                 = (None)
dst        : Emph                 = ('127.0.0.1')
options    : PacketListField      = ([])
>>>

>>> ls(Ether)
dst        : DestMACField         = (None)
src        : SourceMACField       = (None)
type       : XShortEnumField      = (0)
>>>
```

В результате будет выведена информация о полях, которые можно модифицировать в процессе создания пакетов. В скобках показаны значения, которые используются по умолчанию, например, для TCP-сегмента порт отправителя 20, а порт получателя – 80, установлен флаг SYN (flags = 2)

Для того чтобы получить более подробную информацию о каждой функции, можно использовать *help*(*имя_функции*), например:

```
    >>> help(send)
    Help on function send in module scapy.sendrecv:

    send(x, inter=0, loop=0, count=None,
verbose=None, realtime=None, *args, **kargs
    )
    Send packets at layer 3
    send(packets, [inter=0], [loop=0],
[verbose=conf.verb]) -> None

    >>>
```

В утилите Scapy можно создавать сразу фрагменты высоких уровней, используя возможность автоматического дополнения нижележащих уровней, а можно вручную собрать, начиная с канального уровня. Разделяются уровни в модели OSI символом прямого слэша (/). Следует обратить внимание на то, что Scapy читает данные фрагмента слева направо, от нижнего до более высокого. В терминологии Scapy сетевой фрагмент разделяется на слои, и каждый слой представляется как экземпляр объекта. Собранный фрагмент в упрощённом виде может выглядеть как

```
    Ether()/IP()/TCP()/"App Data"
```

В большинстве случаев используется только уровень L3, предоставляя Scapy возможность самостоятельно заполнять канальный уровень на основе информации из операционной системы.

Пример создания простого пакета:

```
    packet=IP(dst="192.168.1.1")/TCP(dport=22)/"TEST"
```

Посмотреть содержимое пакета можно функцией ls:

```
>>> ls(packet)
version   : BitField      = 4               (4)
ihl       : BitField      = None            (None)
tos       : XByteField    = 0               (0)
```

```
len       : ShortField    = None             (None)
id        : ShortField    = 1                (1)
flags     : FlagsField    = 0                (0)
frag      : BitField      = 0                (0)
ttl       : ByteField     = 64               (64)
proto     : ByteEnumField = 6                (0)
chksum    : XShortField   = None             (None)
src       : Emph          = '192.168.1.114' (None)
dst       : Emph          = '192.168.1.1'   ('127.0.0.1')
options   : PacketListField  = []            ([])
--
sport     : ShortEnumField   = 20            (20)
dport     : ShortEnumField   = 22            (80)
seq       : IntField         = 0             (0)
ack       : IntField         = 0             (0)
dataofs   : BitField         = None          (None)
reserved  : BitField         = 0             (0)
flags     : FlagsField       = 2             (2)
window    : ShortField       = 8192          (8192)
chksum    : XShortField      = None          (None)
urgptr    : ShortField       = 0             (0)
options   : TCPOptionsField  = {}            ({})
--
load      : StrField         = 'TEST'           ('')
>>>
```

Уровни модели OSI разделяются символами «--». Пакет можно создавать по частям:

```
part1=IP(dst="192.168.1.1")
part2=TCP(dport=22)
part3="TEST"
```

В этом примере мы создали переменные для каждого уровня модели OSI. Далее необходимо собрать части в один пакет:

```
packet=part1/part2/part3
```

Краткую сводку о переменной (фрагменте) можно получить, указав её имя:

```
>>> packet
<IP frag=0 proto=tcp dst=192.168.1.1 |<TCP
dport=ssh |<Raw load='TEST' |>>>
>>>
```

Для получения детальной информации необходимо использовать метод show():

```
>>> packet.show()
###[ IP ]###
   version= 4
   ihl= None
   tos= 0x0
   len= None
   id= 1
   flags=
   frag= 0
   ttl= 64
   proto= tcp
   chksum= None
   src= 192.168.1.114
   dst= 192.168.1.1
   \options\
###[ TCP ]###
     sport= ftp_data
     dport= ssh
     seq= 0
     ack= 0
     dataofs= None
     reserved= 0
     flags= S
     window= 8192
     chksum= None
     urgptr= 0
     options= {}
###[ Raw ]###
        load= 'TEST'
>>>
```

Можно просмотреть любое поле, указав его название:

```
>>> packet.dst
'192.168.1.1'
>>> packet.dport
22
>>>
```

В любой момент можно поменять значение любого поля:

```
    >>> packet.sport=443
    >>> packet
    <IP frag=0 proto=tcp dst=192.168.1.1 |<TCP
sport=https dport=ssh |<Raw load='TEST' |>>>
    >>>
```

В случае если поле не является уникальным, необходимо указать протокол:

```
    >>> packet[TCP].flags="SA"
    >>> packet
    <IP frag=0 proto=tcp dst=192.168.1.1 |<TCP
sport=https dport=ssh flags=SA |<Raw load='TEST'
|>>>
    >>>
```

Множество адресов можно задать, разделяя значения запятой:

```
    >>>
packet=IP(dst=["192.168.1.1","192.168.1.2","192.168.
1.3"])
```

Можно указать диапазон:

```
    >>> packet=IP(dst="192.168.1.1")/TCP(dport=
(1,1000))
        >>> packet
        <IP frag=0 proto=tcp dst=192.168.1.1 |<TCP
dport=(1, 1000) |>>
        >>> packet=IP(dst="192.168.1.1")/TCP
(dport=(22,23,80))
        >>> packet
        <IP frag=0 proto=tcp dst=192.168.1.1 |<TCP
dport=(22, 23, 80) |>>
        >>>
```

В случае указания диапазона используются круглые скобки, а в случае множества – квадратные.

Функции для отправки пакетов:

send() – отправляет фрагменты, используя сетевой уровень, не обрабатывая ответы;

sendp() – отправляет фрагменты, используя канальный уровень, учитываются указанные параметры и заголовки Ethernet кадров, ответы не обрабатываются;

sr() – является аналогичной send(), но ожидаются ответные пакеты;

srp() – отправляет и принимает кадры на канальном уровне;

sr1() – отправляет пакет третьего уровня и получает только первый ответ, множество ответов не предусматривается;

srp1() – аналогично sr1(), только для канального уровня.

Каждую из этих функций можно вызвать без дополнительных параметров, просто указав имя переменной, содержащей фрагмент.

```
>>> send(packet)
.
Sent 1 packets.
>>>
```

Существует много дополнительных опций для функций отправки. Например, timeout – укажет, сколько времени (в секундах) нужно ждать до получения ответа, retry – сколько раз нужно повторно слать фрагмент, если ответ не был получен. Одна из самых полезных опций – filter, синтаксис опции похож на синтаксис утилиты tcpdump. В качестве наглядного примера отправим пакет в сеть:

```
>>> sr(packet,timeout=1,filter="host 192.168.0.1")
Begin emission:
Finished to send 1 packets.
.*
Received 2 packets, got 1 answers, remaining 0
packets
(<Results: TCP:1 UDP:0 ICMP:0 Other:0>,
<Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>
)
>>>
```

Была использована функция, которая после отправки ожидает ответ, таймаут был установлен в значение 1 секунда, фильтровались ответы, которые подпадали под указанное правило.

Можно назначить переменную, которая будет содержать ответ:

```
>>> response=sr(packet)
Begin emission:
Finished to send 1 packets.
..............................*
Received 32 packets, got 1 answers, remaining 0
packets
>>>
```

Содержимое ответа можно посмотреть, вызвав переменную response. Видно, что ответ сохранился в двух вариантах – Results и Unanswered, результаты ответов и без ответа, соответственно. Указывая смещение response[0] и response [0][0], можно вывести только необходимую часть ответа или подробную информацию:

```
    >>> response [0][0]
    (<IP frag=0 proto=tcp dst=192.168.0.1 |<TCP
dport=http |<Raw load='TEST' |>>>
    , <IP version=4L ihl=5L tos=0x0 len=44 id=0
flags=DF frag=0L ttl=64 proto=tcp chksum=0xb915
src=192.168.0.1 dst=192.168.0.101 options=[] |<TCP
sport=http dport=ftp_data seq=4082761583L ack=1
dataofs=6L reserved=0L flags=SA window=5840 chksum=
0xc61 urgptr=0 options=[('MSS', 1460)] |<Padding
load='\x00\x00' |>>>)
    >>>
```

Если же пакет был отправлен в сеть без указания переменной (например, просто функцией *sr*()), то по умолчанию ответы будут попадать в переменную «_» (символ подчёркивания). Чтобы получить ответы, можно использовать конструкцию res,unans=_:

```
    >>> sr(packet,timeout=1,filter="host 192.168.0.1")
    Begin emission:
    Finished to send 1 packets.
    ....*
    Received 5 packets, got 1 answers, remaining 0
packets
    (<Results: TCP:1 UDP:0 ICMP:0 Other:0>,
<Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>
    )
    >>> res,unans=_
    >>> res
    <Results: TCP:1 UDP:0 ICMP:0 Other:0>
    >>> unans
    <Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>
    >>>
```

При этом разные результаты сохранятся в двух разных переменных (res и unans). Более подробный вывод достигается путём указания смещения. Просмотр первого пришедшего ответа:

```
    >>> res[0]
    (<IP frag=0 proto=tcp dst=192.168.0.1 |<TCP
dport=http |<Raw load='TEST' |>>>
    , <IP version=4L ihl=5L tos=0x0 len=44 id=0
flags=DF frag=0L ttl=64 proto=tcp chksum=0xb915
src=192.168.0.1 dst=192.168.0.101 options=[] |<TCP
sport=http dport=ftp_data seq=92812631 ack=1
dataofs=6L reserved=0L flags=SA window=5840
chksum=0xc44b urgptr=0 options=[('MSS', 1460)]
|<Padding load='\x00\x00' |>>>)
    >>>
```

Теперь рассмотрим ситуацию, когда пакетов в ответ приходит много.

```
    >>>
packet=IP(dst="192.168.0.1")/TCP(dport=(1,100),flag
s= "S")
    >>> res,unans=sr(packet,timeout=10)
    Begin emission:
    Finished to send 100 packets.
    ...*.*.*.*.*.*.*.*.*..*..*.*..*.*.*.*.*.*.*.*.*
...*...*...*.*.*..*.*.*.*.*.*.*..
    .*..*.*...*..*.*.*.*.*.*.*.*.*.*...*.*.*.*..*.*
.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.*.
    *.*.*....*.*.*.*.*.*..*.*.*..*.*..*...*...*.*.*
.*.*.*.*..*.*..*...*.*.*..........
    Received 1122 packets, got 98 answers,
remaining 2 packets
    >>>
```

Код, представленный выше, есть ни что иное, как сканирование портов. С открытых портов будут приходить сегменты с флагами SYN/ACK, например:

```
    >>> res[21]
    (<IP frag=0 proto=tcp dst=192.168.0.1 |<TCP
dport=telnet flags=S |>>, <IP version=4L ihl=5L tos=
0x0 len=44 id=0 flags=DF frag=0L ttl=64 proto=tcp
chksum=0xb915 src=192.168.0.1 dst=192.168.0.101
options=[] |<TCP sport=telnet dport=ftp_data seq=
555566504 ack=1 dataofs=6L reserved=0L flags=SA
window=5840 chksum=0x989e urgptr=0 options=[('MSS',
1460)] |<Padding load='\x00\x00' |>>>)
    >>>
```

Можно посмотреть содержимое пакета, обратившись к нему по его номеру, начиная с нуля:

```
    >>> sent21,rec21=res[21]
    >>> sent21
    <IP frag=0 proto=tcp dst=192.168.0.1 |<TCP
dport=telnet flags=S |>>
    >>> rec21
    <IP version=4L ihl=5L tos=0x0 len=44 id=0
flags=DF frag=0L ttl=64 proto=tcp chksum=0xb915
src=192.168.0.1 dst=192.168.0.101 options=[] |<TCP
sport=telnet dport=ftp_data seq=555566504 ack=1
dataofs=6L reserved=0L flags=SA window=5840 chksum=
0x989e urgptr=0 options=[('MSS', 1460)] |<Padding
load='\x00\x00' |>>>
    >>>
```

Контрольные вопросы

1. Характеристики, функции и назначение уровней модели OSI.
2. Особенности процесса инкапсуляции и деинкапсуляции.
3. Особенности процесса обмена данными между узлами.

Задача для самостоятельного решения

Используя возможности утилиты Scapy, необходимо создать процесс трёхстороннего рукопожатия протокола TCP. Для этого необходимо отправить сегмент для установки соединения, получить ответ с флагами SYN/ACK, извлечь из ответа номер последовательности, увеличить его значение на единицу и, поместив полученное значение в поле подтверждения TCP-сегмента, отправить фрагмент в сеть.

Практическое занятие 2

ПРИНЦИПЫ МАРШРУТИЗАЦИИ

Цель: изучение статической маршрутизации в Cisco IOS.

В результате выполения практического занятия обучаемые *должны*:

– *знать*: функции и принципы маршрутизации; структуру таблицы маршрутизации; принципы настройки и проверки статической маршрутизации; принципы поиска и устранения ошибок конфигурации статической маршрутизации;

– *уметь* настраивать и проверять статические маршруты.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 134 – 161, с. 466 – 499], [4], [8], [9].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Маршрутизация – это процесс пересылки пакетов данных между сетями или подсетями с помощью устройства 3-го уровня. Процесс маршрутизации использует таблицы, протоколы и алгоритмы маршрутизации, чтобы определить наиболее эффективный путь для пересылки IP-пакета. Маршрутизаторы значительно увеличивают масштабируемость сетей.

Маршрутизатор – это сетевое устройство, которое определяет оптимальный путь для передачи данных из одной сети в другую. Маршрутизаторы выполняет две основные функции.

1. *Определение пути*. Маршрутизатор использует таблицу маршрутизации, чтобы найти оптимальный путь для пересылки пакетов.

Когда маршрутизатор получает пакет, он проверяет адрес назначения пакета и использует таблицу маршрутизации для поиска оптимального пути к нужной сети.

2. *Пересылка пакетов*. В таблице маршрутизации содержится информация о том, какой интерфейс следует использовать для пересылки пакетов в каждую известную сеть. Если оптимальный маршрут найден, маршрутизатор инкапсулирует пакет в кадр канала передачи данных выходного интерфейса и пересылает пакет до пункта назначения.

Для выполнения своих задач маршрутизаторы ведут локальные таблицы маршрутизации. На этапе определения пути передачи данных маршрутизатор анализирует доступные пути к удалённым сетям, хранящиеся в таблице маршрутизации.

Записи в таблицах маршрутизации бывают следующих типов.

1. *Сеть с прямым подключением.* Этот тип записи означает, что некоторая сеть имеет прямое подключением к одному из интерфейсов маршрутизатора.

2. *Статический маршрут*. Этот тип записи требует ручного конфигурирования информации о маршруте в таблице маршрутизации.

3. *Динамический маршрут*. При использовании динамической маршрутизации таблица маршрутизации заполняется на основе данных, полученных по протоколам маршрутизации.

4. *Маршрут по умолчанию.* Этот тип записи исключает необходимость установки явно заданного маршрута к каждой сети. Запись маршрута по умолчанию может быть задан статически или приниматься по протоколу динамической маршрутизации.

В таблице маршрутизации сохраняется только одна запись для каждой сети. При наличии нескольких источников информации о пути к одной и той же сети назначения процесс маршрутизации должен иметь возможность выбора источника информации для использования в таблице маршрутизации. Несколько источников появляются при использовании нескольких протоколов динамической маршрутизации, а также статических маршрутов и маршрутов по умолчанию.

В операционной системе Cisco IOS для выбора источника маршрута и занесения его в таблицу маршрутизации применяется административное расстояние (Administrative Distance, AD). Административное расстояние – это мера «надёжности» маршрута: чем меньше значение, тем более надёжным является источник маршрута. В таблице представлены административные расстояния, которые применяются в маршрутизаторах Cisco для различных протоколов маршрутизации.

Источник маршрута	Административное расстояние
Прямое соединение	0
Статический маршрут	1
Суммарный маршрут EIGRP	5
Внешний BGP	20
Внутренний EIGRP	90
IGRP	100
OSPF	110
IS–IS	115
RIP	120
Внешний EIGRP	170
Внутренний BGP	200
Неизвестный	255

Определение оптимального пути подразумевает оценку нескольких путей в одну и ту же сеть назначения и выбор лучшего. Когда существует несколько путей до одной сети, каждый путь использует различный выходной интерфейс маршрутизатора для достижения сети. Протокол маршрутизации выбирает наилучший путь, исходя из метрики, используемой для определения расстояния до сети назначения. Метрика – это числовое значение, используемое для измерения расстояния до заданной сети. Наиболее оптимальным путём к сети является путь с наименьшей метрикой.

На маршрутизаторе Cisco команда **show ip route** может быть использована для отображения таблицы маршрутизации. Маршрутизатор предоставляет дополнительную информацию о маршруте, включая способ получения маршрута, длительность пребывания маршрута в таблице, а также сведения о конкретном интерфейсе, который следует использовать для достижения сети назначения.

Пример вывода команды **show ip route**:

```
Codes: L - local, C - connected, S - static,
R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF,
IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 -
OSPF NSSA external type 2
```

```
       E1 – OSPF external type 1, E2 – OSPF
external type 2
       i – IS-IS, su – IS-IS summary, L1 – IS-IS
level-1, L2 – IS-IS level-2
       ia – IS-IS inter area, * – candidate
default, U – per-user static route
       o – ODR, P – periodic downloaded static
route, H – NHRP, l – LISP
       + – replicated route, % – next hop override

     Gateway of last resort is not set

           10.0.0.0/8 is variably subnetted,
5 subnets, 2 masks
     C 10.1.1.0/30 is directly connected,
Serial0/0/0
     L 10.1.1.1/32 is directly connected,
Serial0/0/0
     D 10.2.2.0/30 [90/2681856] via 10.1.1.2,
00:03:09, Serial0/0/0
     C 10.3.3.0/30 is directly connected,
Serial0/0/1
     L 10.3.3.1/32 is directly connected,
Serial0/0/1
       192.168.1.0/24 is variably subnetted,
2 subnets, 2 masks
     C 192.168.1.0/24 is directly connected,
GigabitEthernet0/0
     L 192.168.1.1/32 is directly connected,
GigabitEthernet0/0
     D 192.168.2.0/24 [90/1794560] via 10.1.1.2,
00:03:09, Serial0/0/0
     D 192.168.3.0/24 [90/2684416] via 10.1.1.2,
00:03:08, Serial0/0/0
```

Каждая запись таблицы маршрутизации содержит следующие сведения: код источника маршрута, адрес сети, административное расстояние/метрика, адрес следующего роутера (next hop), длительность пребывания маршрута в таблице, выходной интерфейс.

Пример:

```
   D 10.89.1.0/26 [90/5639936] via 10.93.1.2,
00:05:15, Serial1
```

D	Код источника маршрута
10.89.1.0/26	Адрес сети
[90/5639936]	[Административное расстояние/метрика],
10.93.1.2	Адрес следующего роутера
00:05:15	Длительность пребывания маршрута в таблице
Serial1	Выходной интерфейс

Источники записей таблицы маршрутизации идентифицируются с помощью кода:

Код источника	Описание
L	Указывает адрес, назначенный интерфейсу маршрутизатора. Данный код позволяет маршрутизатору быстро определить, что полученный пакет предназначен для интерфейса, а не для пересылки
C	Сеть с прямым подключением
S	Статический маршрут
D	Динамически полученный маршрут от другого маршрутизатора с помощью протокола EIGRP
O	Динамически полученный маршрут от другого маршрутизатора с помощью протокола OSPF

Интерфейсы локального маршрута добавляются, когда интерфейс настроен и активен. Эта запись отображается только в Cisco IOS 15 или более поздних версиях для IPv4-маршрутов и во всех версиях IOS для IPv6-маршрутов. Интерфейсы с прямым подключением добавляются в таблицу маршрутизации, когда интерфейс настроен и активен. Статические маршруты добавляются, когда маршрут настроен вручную и активен выходной интерфейс. Протокол динамической маршрутизации добавляется, когда определены сети и реализуются протоколы маршрутизации, которые получают информацию о сети динамически, например EIGRP или OSPF.

Сети с прямым подключением. Такие маршруты создаются на основе информации интерфейсов маршрутизатора, которые непосредственно подключены к сегментам сети. В случае сбоя или административного отключения интерфейса маршрут к этой сети удаляется из таблицы маршрутизации. Административное расстояние таких

маршрутов равно 0, поэтому этот маршрут имеет наивысший приоритет перед всеми остальными маршрутами к этой сети назначения. При получении пакета, адресованного в сеть с прямым подключением, маршрутизатор сразу отправляет пакет на локальный интерфейс, к которому она подключена, не используя протоколы маршрутизации.

Корректно настроенный активный интерфейс с прямым подключением фактически создаёт две записи таблицы маршрутизации.

Например:

```
    C 192.168.10.0/24 is directly connected,
GigabitEthernet0/0
    L 192.168.10.1/32 is directly connected,
GigabitEthernet0/0
```

До версии IOS 15 записи таблицы маршрутизации локальных маршрутов (L) не отображались в таблице маршрутизации IPv4.

Статические маршруты – вводятся системным администратором вручную непосредственно в конфигурацию маршрутизатора. Административное расстояние по умолчанию для статического маршрута равно 1. Поэтому статические маршруты включаются в таблицу маршрутизации, если нет прямого подключения к некоторой сети. Статические маршруты эффективны для сетей небольшого размера, топологии которых изменяются редко.

Динамические маршруты. При использовании протоколов динамической маршрутизации, администратор сети конфигурирует выбранный протокол на каждом маршрутизаторе в сети. После этого маршрутизаторы начинают обмен информацией об известных им сетях. Маршрутизаторы обмениваются информацией с теми маршрутизаторами, где запущен тот же протокол динамической маршрутизации. В случае изменения топологии сети, информация об изменениях автоматически распространяется на все маршрутизаторам, и каждый маршрутизатор вносит необходимые изменения в свою таблицу маршрутизации. Однако между изменением сети и получением сообщений об этом изменении всеми маршрутизаторами всегда существует некоторая задержка. Задержка согласования данных маршрутизатора с изменениями сети называется временем сходимости. Чем меньше время сходимости, тем лучше. Для различных протоколов маршрутизации время сходимости отличается. Динамическую маршрутизацию рекомендуется использовать в крупных сетях и сетях с частыми изменениями топологий.

Маршрут по умолчанию – необязательный маршрут, используемый в случае, если в таблице маршрутизации не найден явно заданный

маршрут к сети назначения. Маршрут по умолчанию может конфигурироваться вручную или приниматься по протоколу динамической маршрутизации.

Метрики маршрутов в таблице маршрутизации могут основываться на одной или нескольких характеристиках пути следования пакетов до сети назначения, указанной в этом маршруте. В протоколах маршрутизации для подсчёта метрики наиболее часто используются следующие характеристики.

Полоса пропускания – пропускная способность канала (соединения) между двумя сетевыми устройствами.

Задержка – интервал времени, который требуется для перемещения пакета по каналу связи между устройствами. Задержка зависит от размера очередей в портах маршрутизатора, загрузки сети и физического расстояния.

Загрузка – средняя загруженность канала связи в единицу времени.

Надёжность – относительное количество ошибок на канале связи.

Количество переходов – количество маршрутизаторов, которые должен пройти пакет, прежде чем достигнет сети назначения.

Стоимость – произвольное значение, назначаемое сетевым администратором, которое обычно основано на полосе пропускания, предпочтениях администратора или других показателях.

2. СТАТИЧЕСКАЯ МАРШРУТИЗАЦИЯ

Преимущества статической маршрутизации.

– статические маршруты не распространяются по сети, поэтому их использование является более безопасным процессом;

– исключение служебного трафика, связанного с поддержкой и корректировкой таблиц маршрутизации;

– снижение нагрузки на маршрутизатор;

– путь, используемый статическим маршрутом для отправки данных, известен.

У статической маршрутизации также имеются недостатки:

– исходная настройка и дальнейшее обслуживание требуют временных затрат;

– для внесения изменений в данные маршрутов требуется вмешательство администратора;

– недостаточные возможности масштабирования для растущих сетей, обслуживание при этом становится довольно трудоёмким;

– для эффективного использования статической маршрутизации требуется подробное знание топологии сети.

Статические маршруты рекомендуется использовать в небольших сетях, для которых задан только один путь к внешней сети. Важно понимать, что статическая и динамическая маршрутизация не являются взаимоисключающими. В большинстве сетей используется комбинация протоколов динамической маршрутизации и статических маршрутов. Это может привести к тому, что для маршрутизатора задаётся несколько путей к сети назначения посредством статических маршрутов и динамически получаемых маршрутов. Однако административное расстояние (AD) статического маршрута равно 1. Поэтому статический маршрут имеет приоритет по сравнению со всеми динамически получаемыми маршрутами.

Статический маршрут определяет IP-адрес следующего соседнего маршрутизатора или локальный выходной интерфейс, который используется для направления трафика к определённой сети получателю. Как следует из названия, статический маршрут не может быть автоматически адаптирован к изменениям в топологии сети. Если определённый в маршруте маршрутизатор или интерфейс становятся недоступными, то маршрут к сети получателю становиться недоступным.

Статическая маршрутизация может быть использована в следующих ситуациях:

– когда администратор нуждается в полном контроле маршрутов используемых маршрутизатором;

– когда необходимо резервирование динамических маршрутов;

– когда есть сети, достижимые единственно возможным путём (тупиковые сети);

– когда нежелательно иметь служебный трафик, необходимый для обновления таблиц маршрутизации (например, при использовании коммутируемых каналов связи);

– когда используются устаревшие маршрутизаторы, не имеющие необходимого уровня вычислительных возможностей для поддержания динамических протоколов маршрутизации.

Настройка статических маршрутов. В таблице маршрутизации представлены два распространённых типа статических маршрутов:

1) в конкретную сеть;
2) по умолчанию (или маршруты по умолчанию).

Для конфигурации статического маршрута используется команда **ip route**:

```
Router(config)# ip route network-address
subnet-mask { ip-address | exit-intf [ip-address]}
[distance] [name name] [permanent] [tag tag]
```

Параметры команды	Описание
`network-address`	Адрес удалённой сети назначения, который необходимо добавить в таблицу маршрутизации (данный параметр часто называют префиксом)
`subnet-mask`	Маска удалённой сети, которую необходимо добавить в таблицу маршрутизации
`ip-address`	IP-адрес соседнего маршрутизатора, который используется для пересылки пакетов в сеть назначения. Этот IP-адрес чаще всего называют следующим переходом или следующим узлом
`exit-intf`	Исходящий интерфейс, который используется для передачи пакета на следующий переход
`distance`	Заданное значение административного расстояния
`name`	Назначение имени указанному маршруту
`permanent`	Указание того, что маршрут не может быть удалён из таблицы маршрутизации, если интерфейс, на который он указывает, становится недоступным
`tag`	Метка, использующаяся при перераспределения маршрутов

Чаще всего используется упрощённый синтаксис команды:

```
Router(config)# ip route network-address subnet-mask {ip-address | exit-intf}
```

Статические маршруты должны быть заданы *на обоих концах канала связи* между маршрутизаторами, иначе удалённый маршрутизатор не будет знать маршрута, по которому нужно отправлять ответные пакеты и будет построена лишь односторонняя связь.

В качестве выходного адреса с маршрутизатора для статического маршрута может применяться IP-адрес входного интерфейса соседнего маршрутизатора или указываться выходной интерфейс маршрутизато-

ра. Единственным различием между этими двумя видами записи команды будет являться административное расстояние маршрута при его помещении в таблицу маршрутизации. Стандартно при использовании адреса следующего перехода административное расстояние устанавливается равным 1. При задании выходного интерфейса для административного расстояния устанавливается значение 0.

Пример:

```
    Router(config)# ip route 172.16.1.0
255.255.255.0 172.16.2.2
    Router(config)# ip route 172.16.3.0
255.255.255.0 Serial0/1
```

Настройка статического маршрута с выходным интерфейсом позволяет маршрутизатору определить выходной интерфейс в ходе одного процесса поиска маршрута в таблице маршрутизации вместо двух.

В полностью заданном статическом маршруте указываются и выходной интерфейс, и IP-адрес следующего перехода. Такой тип статического маршрута используется тогда, когда выходной интерфейс представляет собой интерфейс, подключённый к сети с множественным доступом и есть необходимость явно определить следующий переход.

Пример:

```
    Router(config)# ip route 172.16.1.0
255.255.255.0 GigabitEthernet 0/1 172.16.2.2
```

Иногда статические маршруты могут использоваться в качестве резервных. Согласно административному расстоянию маршрутизатор в большей степени доверяет статическим маршрутам. Когда существует необходимость сконфигурировать резервный статический маршрут для динамического маршрута, то в такой ситуации статический маршрут не должен использоваться, пока доступен динамический маршрут.

С помощью опции **distance** можно сделать статический маршрут менее предпочтительным, чем динамический маршрут, или один статический маршрут сделать более предпочтительным, чем другой статический маршрут. Такие маршруты называют *плавающими.*

Пример:

```
    Router(config)# ip route 10.1.2.0 255.255.255.0
172.16.2.2
    Router(config)# ip route 10.1.2.0 255.255.255.0
172.16.1.2 100
```

Пока будет доступен основной канал связи, статический маршрут с AD = 100 не будет занесён в таблицу маршрутизации, потому что его административное расстояние больше чем у стандартного статического маршрута. Как только основной канал связи станет недоступным, плавающий маршрут будет внесён в таблицу маршрутизации.

Статический маршрут по умолчанию – это маршрут, которому соответствуют все пакеты. Вместо хранения всех маршрутов ко всем сетям в таблице маршрутизации маршрутизатор может хранить один маршрут по умолчанию, представляющий любую сеть, отсутствующую в таблице маршрутизации.

Маршрут по умолчанию может быть настроен администратором или получен от другого маршрутизатора с помощью протокола динамической маршрутизации. Маршрут по умолчанию используется только тогда, если IP-адрес сети назначения пакета не совпадает ни с одним маршрутом в таблице маршрутизации.

Статические маршруты по умолчанию обычно используются при подключении:

– пограничного маршрутизатора компании к сети интернет-провайдера;

– «тупикового» маршрутизатора (маршрутизатора только с одним соседним маршрутизатором в исходящем направлении).

Синтаксис команды для задания статического маршрута по умолчанию аналогичен синтаксису команды для любого другого статического маршрута за исключением того, что адрес сети указывается как 0.0.0.0, а маска подсети – 0.0.0.0. Синтаксис основной команды статического маршрутизатора по умолчанию следующий:

```
Router(config)# ip route 0.0.0.0 0.0.0.0
{ip-address | exit-intf}
```

Статический маршрут IPv4 по умолчанию обычно называют маршрутом с четырьмя нулями (quad-zero).

Если маршрут по умолчанию задан статически, в таблице маршрутизации он обозначается кодом S*.

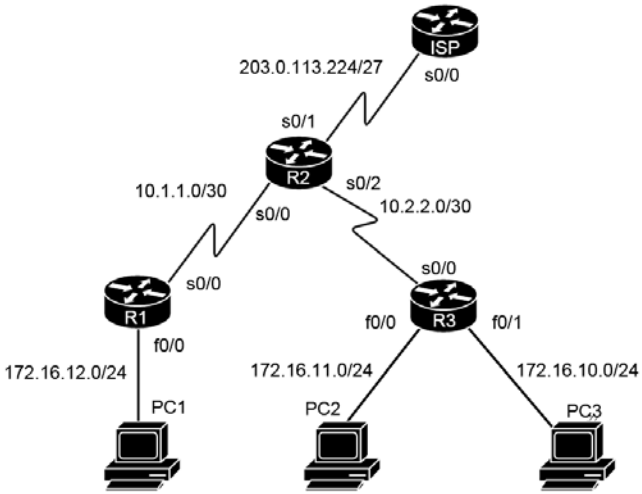
Команды для проверки статических маршрутов: **show ip route, show ip route static, show ip route** *network*.

Контрольные вопросы

1. Функции и принципы маршрутизации.
2. Структура таблицы маршрутизации.
3. Принципы настройки и проверки статической маршрутизации.
4. Принципы поиска и устранения ошибок конфигурации статической маршрутизации.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить адресацию устройств согласно таблице:

Устройство	Интерфейс	IP-адрес	Маска	Шлюз
R1	S0/0	10.1.1.12	255.255.255.252	–
	F0/0	172.16.12.1	255.255.255.0	–
R2	S0/0	10.1.1.	255.255.255.252	–
	Lo0	203.0.113.225	255.255.255.255	–
	S0/2	10.2.2.1	255.255.255.252	–
R3	S0/0	10.2.2.2	255.255.255.252	–
	Fa0/0	172.16.10.1	255.255.255.0	–
	Fa0/1	172.16.11.1	255.255.255.0	–
PC1	NIC	172.16.12.10	255.255.255.0	172.16.12.1
PC2	NIC	172.16.11.10	255.255.255.0	172.16.11.1
PC3	NIC	172.16.10.10	255.255.255.0	172.16.10.1

3. Настроить статическую маршрутизацию следующим образом:

1) маршруты должны быть кратчайшими;

2) с каждого узла должен быть доступен любой узел любой подсети;

3) суммировать маршруты, где существует такая возможность.

Практическое занятие 3

ПРОТОКОЛ EIGRP

Цель: изучение особенностей функционирования и настройки протокола EIGRP.

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: принципы настройки и проверки протокола EIGRP, распределения нагрузки в протоколе EIGRP, аутентификации в протоколе EIGRP; принципы поиска и устранения ошибок конфигурации протокола EIGRP;

– *уметь* настраивать и устранять неполадки в работе протокола EIGRP в сетях IPv4 и IPv6.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.
2. Основная часть – устный или письменный опрос, решение предложенных задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [3, с. 315 – 373], [6].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

EIGRP (Enhanced Interior Gateway Routing) – это проприетарный протокол маршрутизации Cisco на базе векторов расстояния, включающий возможности протоколов маршрутизации с учётом состояния канала. Маршруты, полученные по протоколу EIGRP, в таблице маршрутизации помечаются буквой D. Административное расстояние: суммарные маршруты – 5, внутренние маршруты – 90, внешние маршруты – 170.

Алгоритм диффузионного обновления (*DUAL*). EIGRP использует алгоритм DUAL (Diffusing Update Algoritm) для ускорения сходимости. Маршрутизатор под управлением EIGRP сохраняет все доступные маршруты к местам назначения и может быстро адаптироваться к аль-

тернативным маршрутам. Если в таблице маршрутизации нет подходящих маршрутов или резервных маршрутов, EIGRP запрашивает соседние узлы, чтобы обнаружить альтернативный маршрут.

Бесклассовая маршрутизация. Поскольку EIGRP является бесклассовым протоколом маршрутизации, он передаёт маску маршрутов для каждой сети назначения. Маска маршрутов позволяет EIGRP работать с несмежными подсетями и масками подсети переменной длины (VLSM).

Установление отношений смежности. EIGRP устанавливает отношения с напрямую подключёнными маршрутизаторами, на которых также включена поддержка EIGRP. Отношения смежности с соседними устройствами используются для отслеживания статуса соседних устройств.

Надёжный транспортный протокол (*Reliable Transport Protocol*, *RTP*). Надёжный транспортный протокол (RTP) является уникальным для EIGRP, обеспечивая доставку пакетов EIGRP соседним маршрутизаторам и отслеживание отношений смежности с соседними устройствами.

*Частичные и ограниченные обновления***.** Протокол EIGRP не отправляет периодических обновлений, и записи маршрутов не устаревают. Термин «частичное» означает, что обновление содержит только данные об изменениях маршрутов, например о новом канале или о канале, ставшем недоступным. Термин «ограниченное» относится к распространению частичных обновлений, которые отправляются только тем маршрутизаторам, на работу которых влияют эти изменения. Это снижает требования к пропускной способности, необходимой для передачи обновлений EIGRP.

Рассылка меньшего объёма служебных данных. EIGRP использует многоадресную и одноадресную, а не широковещательную рассылку. В результате обновления маршрутизации и запросы данных топологии не затрагивают конечные станции.

Балансировка нагрузки. EIGRP поддерживает балансировку нагрузки по маршрутам с неравными метриками, что позволяет администраторам более эффективно распределять потоки трафика в сети.

Простое суммирование. EIGRP позволяет администраторам создавать суммарные маршруты в любой точке сети, не ограничиваясь традиционным классовым суммированием дистанционно-векторных протоколов, которое можно использовать только на границах основной сети.

Протокол EIGRP определяет свой домен маршрутизации, который включает все маршрутизаторы, поддерживающие EIGRP, а также

сети внутри домена с помощью автономной системы (AS). Маршрутизаторы EIGRP могут обмениваться информацией, только если они имеют один и тот же номер AS, т.е. они рассматриваются как члены одного домена маршрутизации. Автономные системы EIGRP, имеющие различные номера, не могут обмениваться маршрутной информацией. Номер AS назначается произвольно при настройке и запуске EIGRP на первом маршрутизаторе домена. После назначения номера AS все другие маршрутизаторы внутри автономной системы должны иметь тот же номер автономной системы.

Маршрутизаторы EIGRP внутри автономной системы должны в первую очередь распознать соседние маршрутизаторы, непосредственно подключённые к их собственным интерфейсам. Идентифицируя своих соседей, маршрутизаторы будут способны определить, какие из них недоступны, и тем самым обнаружить неисправность в сети. Это позволяет им быстро отреагировать на неисправность и откорректировать маршруты в таблице маршрутизации.

База данных протокола EIGRP. Все маршрутизаторы EIGRP внутри одной и той же автономной системы должны создать и поддерживать базу данных, содержащую две таблицы.

1. *Таблица соседей.* Все маршрутизаторы EIGRP ведут таблицу соседства, в которой хранится список соседних маршрутизаторов. Эта таблица, как и в протоколах по состоянию канала служит для обеспечения дуплексного обмена данными между соседями, имеющими непосредственное соединение.

2. *Таблица топологии.* Маршрутизатор EIGRP ведёт таблицу топологии для всех сетей, на которые настроен протокол. В таблице хранятся все известные маршруты ко всем известным сетям, а также вся необходимая информация о маршруте. Таблица изменяется каждый раз при любых изменениях в топологии сети. Каждый маршрутизатор EIGRP передаёт свою таблицу топологии всем своим соседям из таблицы соседства. Сосед, получивший таблицу топологии соседнего маршрутизатора записывает её в свою базу данных топологии сети. Маршрутизатор EIGRP изучает базу данных топологии с целью нахождения лучшего маршрута до каждой сети адресата. По нахождению лучшего маршрута до сети получателя этот маршрут передаётся в таблицу маршрутизации.

Таблица соседей. Чтобы начать обмен маршрутной информацией, маршрутизаторы EIGRP, находящиеся в одном и том же сегменте в пределах одного домена маршрутизации, должны сформировать соседские взаимоотношения (отношения смежности). Маршрутизаторы становятся соседями после того, как они обменяются приветствен-

ными пакетами. Этот процесс называется процессом обнаружения соседей. Каждый маршрутизатор в результате обмена приветственными сообщениями (Hello-пакетами) создаёт локальную таблицу соседей, отслеживая всех соседей и их состояние. Hello-пакеты рассылаются каждый 5 секунд в сетях, скорость которых больше 1,5 Мб/с, и каждые 60 секунд в сетях, скорость которых меньше 1,5 Мб/с. Если маршрутизатор не получает приветственного сообщения от соседнего маршрутизатора в течение трёх временных интервалов (15 и 180 секунд соответственно), то он считает его неработоспособным и удаляет из своей таблицы.

Вывод команды *show ip eigrp neighbor:*

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	20.0.0.1	Se0/0/0	10	00:00:26	27	162	0	19
1	20.0.0.6	Se0/0/1	10	00:00:25	20	120	0	15

Ниже описаны поля, содержащиеся в таблице соседства.

Номер автономной системы (*process 1*), заданный с помощью команды *router eigrp.*

Номер соседнего маршрутизатора (*H*) – порядковый номер соседнего маршрутизатора.

Адрес соседнего маршрутизатора (*Address*) – IP-адрес соседнего маршрутизатора.

Интерфейс (*Interface*) – локальный интерфейс, получающий пакеты приветствия EIGRP.

Время удержания (*Hold*) – интервал времени, по истечении которого, в случае отсутствия сообщений от соседнего маршрутизатора, сосед рассматривается как неработоспособный. По умолчанию интервал равен трём интервалам отправки пакетов приветствия. Первоначально в качестве ожидаемых пакетов рассматривались только пакеты приветствия, однако, в текущих версиях IOS любой пакет протокола EIGRP сбрасывает таймер на нулевое значение.

Время существования соседских отношений (*Uptime*) – временной интервал, прошедший с момента установки соседских отношений с данным маршрутизатором.

Счётчик очереди (*Q Count*) – число пакетов, которые находятся в очереди и ожидают передачи. Если это значение постоянно больше нуля, то маршрутизатор испытывает переполнение. Значение 0 означает, что пакетов EIGRP в очереди нет.

Последовательный номер (*Seq Num*) – номер последнего пакета, полученного от данного соседнего маршрутизатора. Протокол EIGRP использует это поле для подтверждения приема пакета, переданного соседним маршрутизатором и для идентификации пакетов, которые переданы с нарушением порядка.

Таймер цикла обмена сообщениями (*SRTT*) – среднее время, которое требуется для того, чтобы отправить пакет соседнему маршрутизатору и получить от него ответный пакет. Этот таймер определяет интервал повторной передачи пакета.

Таймер повторной передачи (*RTO*) – время, которое маршрутизатор ожидает прихода ответного сообщения от соседа после посылки ему пакета. После истечения таймера происходит повторная отправка неподтверждённого пакета.

Таблица топологии. Все маршрутизаторы EIGRP должны создавать и поддерживать в актуальном состоянии таблицу топологии. Эта таблица представляет собой карту всей автономной системы, в которой указаны все сети, подсети и метрики маршрутов ко всем получателям. Процесс создания и поддержки таблицы топологии является результатом обмена маршрутной информацией. Обмен маршрутной информацией начинается после завершения установки соседских отношений между смежными маршрутизаторами в домене EIGRP. Маршрутизатор EIGRP во время инициализации получают полные копии таблиц топологии своих соседей и на их основе создают свою таблицу топологии домена маршрутизации. С этой целью маршрутизаторы EIGRP используют запросы о сетях получателях, ответы на эти запросы и обновления маршрутной информации. Маршрутизаторы EIGRP используют данные из таблицы топологии для создания и поддержания в актуальном состоянии таблицы маршрутизации, с помощью алгоритма DUAL вычисляя маршруты до сетей получателей.

Вывод команды **show ip eigrp topology***:*

```
    IP-EIGRP Topology Table for AS 1

    Codes: P - Passive, A - Active, U - Update,
Q - Query, R - Reply,
    r - Reply status

    P 192.168.0.0/24, 2 successors, FD is 2684416
    via 20.0.0.6 (2684416/2172416), Serial0/0/1
    via 20.0.0.1 (2684416/2172416), Serial0/0/0
```

Ниже описаны поля, содержащиеся в таблице топологии.

Статус маршрута (*Codes*) – значения Passive и Active обозначают состояния EIGRP с точки зрения места назначения. Пассивные

маршруты (P), под которыми понимаются устойчивые и готовые к использованию маршруты, и активные (A), в отношении которых алгоритм DUAL не закончил процесс расчета маршрута. Значения Update, Query и Reply обозначают тип отправленного пакета: U – отправлен пакет обновления, Q – отправлен пакет запроса, R – отправлен пакет ответа. r – флаг, который устанавливается после того, как маршрутизатор отправляет запрос и начинает ждать ответа.

192.168.0.0/24 – сеть назначения, которая также находится в таблице маршрутизации.

Число приёмников (*successors*) – число маршрутов с равной стоимостью до сети назначения, или другими словами число маршрутизаторов которым могут быть переданы далее пакеты при маршрутизации.

Источник маршрута (*via*) – адрес маршрутизатора анонсировавшего маршрут. Если маршрут анонсирован несколькими маршрутизаторами, то первые *n* строк являются маршрутизаторами приёмниками (*n* – число приёмников), а остальные маршруты выступают либо вероятными приёмниками, либо просто резервными маршрутами. Строка connected означает, что сеть является непосредственно подключённой к маршрутизаторам.

Выполнимое/*Заявленное расстояние* (*Feasible*/*Advertised distance*). Выполнимое расстояние это полная метрика маршрута равная заявленному расстоянию от соседнего маршрутизатора до сети адресата плюс метрика маршрута до заявившего его соседнего маршрутизатора. Заявленное расстояние это метрика маршрута от соседа до сети назначения.

Выходной интерфейс – интерфейс маршрутизатора, через который доступна сеть получатель.

В описании полей таблицы топологии приводилось четыре важнейших определения для протокола EIGRP.

Приёмник (*Successor*) – это лучший маршрут, который используется для достижения получателя. Приёмники передаются в таблицу маршрутизации.

Вероятный приёмник (*Feasible Successor*) – это сосед, который находится на пути к получателю. Его нельзя назвать оптимальным, поэтому он не используется для пересылки данных, это резервный маршрут к получателю. Эти маршруты определяются одновременно с приёмниками и сохраняются в таблице топологии. Таблица топологии может хранить множество вероятных приёмников.

Заявленное расстояние (*Advertised Distance или Reported Distance*) – метрика маршрута, полученная от соседа, заявившего его до сети адресата.

Выполнимое расстояние (*Feasible Distance*) – заявленное расстояние от соседа до сети адресата плюс метрика маршрута до соседа.

На рисунке 8 дано представление понятий заявленного и выполнимого расстояний.

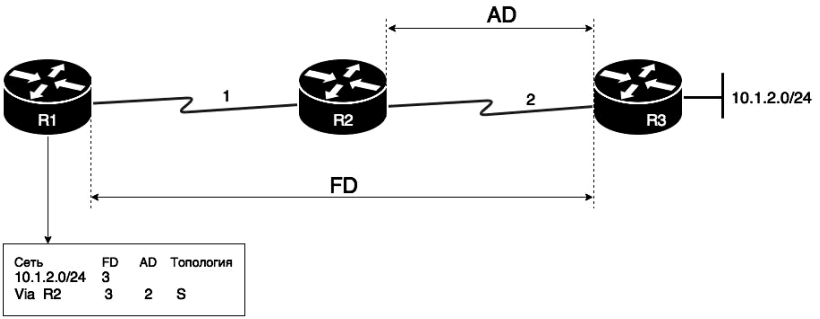


Рис. 8. Пример содержимого таблицы топологии

Метрика протокола. Протокол EIGRP для оценки маршрутов использует комбинированную метрику, состоящую из суммы нескольких метрик умноженных на их весовые коэффициенты.

Ширина полосы пропускания (*BW*). Наименьшая пропускная способность между отправителем и получателем.

Время задержки (*D*) – суммарная задержка по всему маршруту.

Надёжность (*R*) – самая низкая надёжность канала между отправителем и получателем.

Нагрузка (*L*) – максимальная нагрузка, имеющаяся в канале между отправителем и получателем; измеряется в битах в секунду.

Метрика протокола EIGRP рассчитывается по формуле

$$Metric = \left(K1\,BW + \frac{K2\,BW}{256 - L} + K3\,D \right)\left(\frac{K5}{R + K4} \right),$$

где $K1$, $K2$, $K3$, $K4$, $K5$ – весовые коэффициенты.

По умолчанию константы принимают следующие значения: $K1 = K3 = 1$, $K2 = K4 = K5 = 0$.

Следовательно, по умолчанию формула для расчёта метрики EIGRP имеет вид

$$Metric = BWD.$$

Весовые коэффициенты передаются в пакетах приветствия (Hello-пакеты). Несоответствие весовых коэффициентов не позволит установить соседские отношения между маршрутизаторами. Весовые коэф-

фициенты могут модифицироваться только после тщательного планирования, изменение этих значений может препятствовать сходимости сети. Для определения значений, требуемых при вычислении метрики, протокол EIGRP использует формулы

$$BW = \frac{10^7}{bandwitch} \cdot 256,$$

где *bandwitch* – полоса пропускания канала связи заданная на интерфейсе;

$$D = \frac{delay}{10} \cdot 256,$$

где *delay* – задержка на канале связи, рассчитанная маршрутизатором.

Итоговая формула для расчёта метрики имеет вид

$$BW = \left(\frac{10^7}{bandwitch} + \frac{delay}{10} \right) \cdot 256.$$

Используя команду show interfaces, можно получить параметры, используемые для расчёта метрики EIGRP.

Функционирование протокола. Протокол EIGRP использует в своей работе пять типов служебных пакетов, описание которых приводится в таблице:

Тип	Назначение пакета
Update (1)	Пакеты обновления рассылаются для обмена данными о маршрутах до сетей получателей. При изменении топологии сети маршрутизатор рассылает пакеты обновления всем маршрутизаторам из его таблицы соседства
Query (3)	Пакеты запросы рассылаются всем соседям с целью нахождения маршрута до сети получателя, когда приёмник маршрута становиться недоступен
Reply (4)	Пакет отсылается в ответ на запрос
Hello (5)	Пакеты приветствия используются для поиска соседей и дальнейшего подтверждения работоспособности соседних маршрутизаторов. Они рассылаются по групповому адресу 224.0.0.10 и не требуют подтверждения
ACK (5)	Пакет подтверждения используется для подтверждения получения пакетов обновлений, запросов н ответов. Пакет ACK представляет собой пустой Hello пакет, в котором указан номер пакета, получение которого подтверждается

После запуска протокола EIGRP на маршрутизаторе он начинает рассылку Hello пакетов со всех активных интерфейсов по групповому адресу 224.0.0.10. Когда маршрутизатор получает на свой интерфейс Hello пакет от другого маршрутизатора, содержащий такой же номер автономной системы, между маршрутизаторами запускается процесс установки соседских отношений. Соседские отношения не устанавливаются, если не совпадают номера автономных систем или в полученных Hello пакетах содержатся отличные от настроенных на маршрутизаторе весовые коэффициенты.

Запуск протокола EIGRP. Для запуска протокола EIGRP используется команда

```
Router(config)# router eigrp as-number
```

Параметр *as-number* представляет собой номер автономной системы, который используется для идентификации маршрутизаторов, принадлежащих домену маршрутизации. Это значение должно совпадать у всех маршрутизаторов в пределах домена маршрутизации.

Для описания сетей, участвующих в процессе маршрутизации, используется команда **network**:

```
Router(config-router)# network ip-address
[wildcard-mask]
```

Параметры	Описание
ip-address	Адрес сети, участвующей в процессе маршрутизации EIGRP
wildcard-mask	Обратная маска для сети, участвующей в процессе маршрутизации EIGRP

По значению *network* маршрутизатор определяет, какие сети будут участвовать в процессе маршрутизации EIGRP, и через какие интерфейсы производить рассылку служебных пакетов протокола EIGRP. По умолчанию рассылка служебных пакетов производится со всех интерфейсов, попадающих в параметр *network,* поэтому рекомендуется использовать команду *passive-interface* для контроля интерфейсов, с которых не должна производиться рассылка служебной информации.

Пример:

```
Router(config)# router eigrp 200
Router(config-router)# network 10.1.0.0
0.0.255.255
Router(config-router)# network 10.4.0.0
0.0.255.255
Router(config-router)# passive-interface fa 0/0
```

Алгоритм DUAL – использует метрики маршрутов для определения лучшего маршрута до сети получателя. Метрика маршрута до сети получателя (выполнимое расстояние – FD) рассчитывается суммированием метрики заявленной соседом (заявленное расстояние – AD) и метрики маршрута до соседа, заявившего этот маршрут.

Маршрутизатор, объявивший маршрут с самой низкой метрикой, становится приёмником (Successor) – соседом, на которого будут передаваться пакеты до сети получателя. Может быть несколько приёмников, если они имеют одинаковые FD до сети получателя. Все приёмники помещаются в таблицу маршрутизации.

Алгоритм DUAL может вычислить резервный маршрут через вероятного приёмника. Маршрутизатор может быть выбран алгоритмом DUAL в качестве вероятного приёмника (Feasible Successor), если заявленное им расстояние до сети получателя меньше, чем выполнимое расстояние до сети получателя через маршрутизатор-приёмник. Вероятные приёмники не заносятся в таблицу маршрутизации, а хранятся в таблице топологии. В ней могут присутствовать более одного вероятного приёмника.

Если маршрутизатор-приёмник становится недоступным, а для данного маршрута есть вероятный приёмник, то он заносится в таблицу маршрутизации на место приёмника, при этом не производятся дополнительные перерасчёты маршрутов.

Если маршрутизатор-приёмник становится недоступным, а в таблице топологии отсутствуют вероятные приёмники, то алгоритму DUAL необходимо произвести выборы нового приёмника, если это возможно, что потребует некоторого времени, в течение которого пакеты до сети получателя отправляться не будут.

Рассмотрим пример на рис. 9.

Для сегмента сети передачи данных, изображённой на рисунке, маршрутизатор R2 объявляет маршрутизатору R3 сеть 10.1.1.0/24 с заявленным расстоянием (AD) 1000. Маршрутизатор R3 заносит в свою таблицу топологии заявленное расстояние маршрутизатором R2 до сети 10.1.1.0/24 и вычисляет выполнимое расстояние (FD) для этой сети через маршрутизатор R2, которое равняется 2000. Маршрутизатор R4 также объявляет маршрутизатору R3 сеть 10.1.1.0/24, но с заявленным расстоянием (AD) 1500. Маршрутизатор R3 также заносит в таблицу топологии AD от маршрутизатора R4 и вычисляет FD через него, равное 3000. Исходя из полученных данных, маршрутизатор R3 назначает приёмником для сети 10.1.1.0/24 маршрутизатор R2, так как FD через маршрутизатор R2 до сети 10.1.1.0/24 наименьшее.

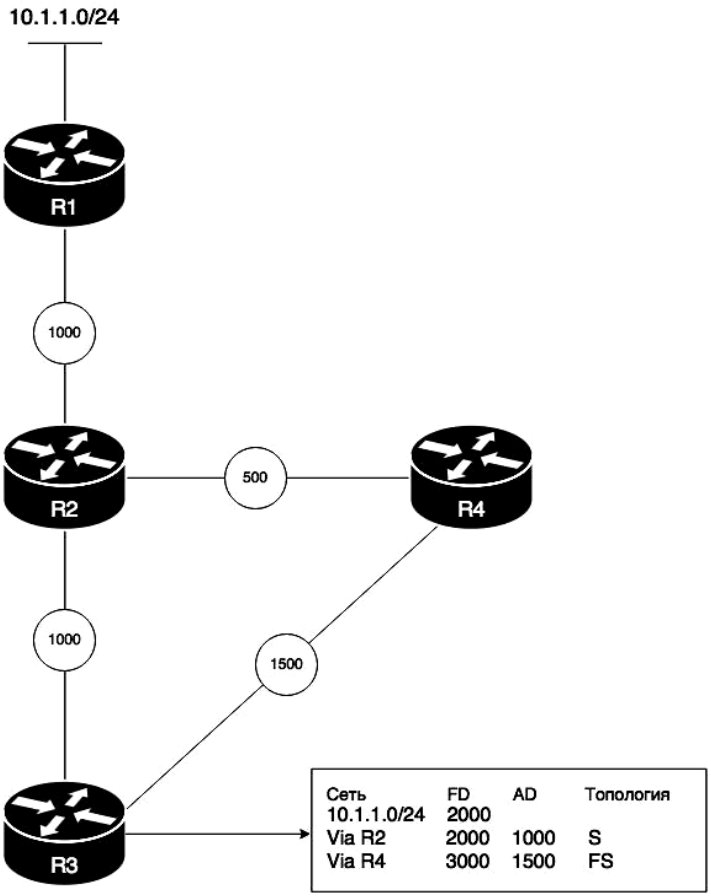


Рис. 9. Выбор приёмника и возможного приёмника

Однако AD, заявленное маршрутизатором R4 до сети 10.1.1.0/24, меньше FD приёмника, назначенного маршрутизатором R3, следовательно, маршрутизатор R3 имеет право назначить маршрутизатор R4 вероятным приёмником для сети 10.1.1.0/24. Если канал связи между маршрутизаторами R2 и R3 выходит из строя, маршрутизатор R3 вычёркивает из таблицы топологии запись о маршрутизаторе R3, приёмником для сети 10.1.1.0/24 становится маршрутизатор R4 и этот маршрут записывается в таблицу маршрутизации. Других действий маршрутизатор R3 не производит, при этом передача пользовательского трафика не прерывается.

Балансировка нагрузки. Балансировка нагрузки по маршрутам с одинаковой стоимостью – это способность маршрутизатора распределять трафик по сетевым портам, обеспечивающим маршруты с одинаковой стоимостью к адресу назначения. Балансировка нагрузки повышает коэффициент использования сетевых сегментов и эффективную полосу пропускания сети.

Cisco IOS реализует балансировку нагрузки по четырём (значение по умолчанию) маршрутам равной стоимости. С помощью команды конфигурации интерфейса **maximum-paths** *number* можно увеличить максимальное число маршрутов с равной стоимостью до 16. Чтобы отключить балансировку нагрузки, необходимо установить значение 1 для параметра *number*. При использовании процессорной коммутации (process switching) балансировка нагрузки по маршрутам с одинаковой стоимостью выполняется для отдельных пакетов. При использовании быстрой коммутации (fast switching) балансировка нагрузки по маршрутам с одинаковой стоимостью выполняется по получателям.

EIGRP также может балансировать трафик по нескольким маршрутам с разными метриками. Эта функция называется балансировкой нагрузки по маршрутам с неравной стоимостью. Уровень балансировки нагрузки EIGRP управляется командой **variance**:

```
Router(config-router)# variance multiplier
```

Параметр *multiplier* принимает значение от 1 до 128. По умолчанию значение равно 1, что означает распределение нагрузки по маршрутам с равной стоимостью. Множитель отражает диапазон значений метрик маршрутов, которые будут приниматься в расчёт для распределения нагрузки.

Рассмотрим пример на рисунке 10.

На рисунке диапазон метрик, для маршрутов от маршрутизатора R5 до сети Z, составляет от 20 до 45. Этот диапазон значений используется в процедуре определения потенциального маршрута. Маршрут считается приемлемым, если следующий маршрутизатор, лежащий на пути, будет ближе к получателю, чем текущий, и метрика всего маршрута лежит в пределах вариации.

Если эти условия соблюдены, то такой маршрут будет считаться приемлемым, и он будет записан в таблицу маршрутизации. Для распределения нагрузки могут быть использованы только приемлемые маршруты.

На рисунке имеются три маршрута к сети Z. Метрики для этих маршрутов:

30 – верхний маршрут;

20 – средний маршрут;

45 – нижний маршрут.

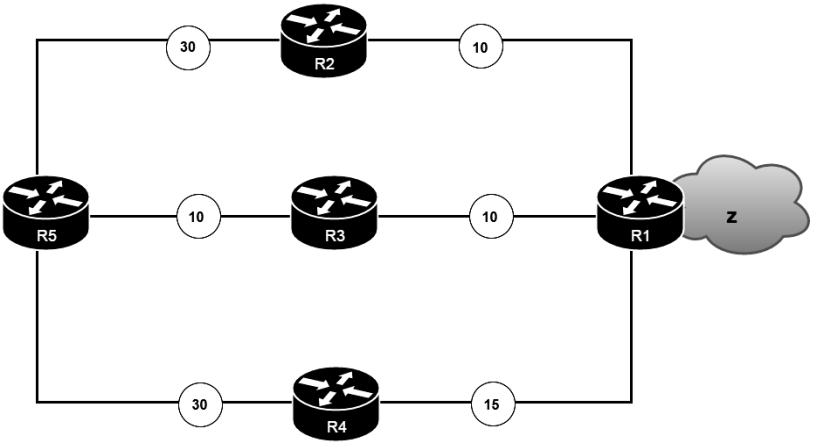


Рис. 10. Пример топологии

Для включения балансировки между средним и верхним маршрутами необходимо использовать variance = 2, так как 20·2 = 40, а это больше метрики верхнего маршрута. Точно так же, чтобы добавить нижний маршрут, необходимо использовать variance = 3.

При включении процесса распределения нагрузки по маршрутам с разными метриками, маршрутизатор производит распределение пакетов в зависимости от величины variance. По маршруту, соответствующему variance = 2, будет отправлено в два раз меньше пакетов относительно количества пакетов, отправленных по наилучшему маршруту.

Аутентификация в протоколе EIGRP. Аутентификация соседних узлов EIGRP позволяет маршрутизаторам принимать участие в процессе маршрутизации только при наличии пароля. По умолчанию, аутентификация для пакетов EIGRP не используется. Для протокола EIGRP можно настроить аутентификацию Message Digest 5 (MD5).

Маршрутизатор, на котором настроена аутентификация соседних узлов, аутентифицирует источник каждого полученного пакета обновления маршрутизации. Для аутентификации MD5 необходимо настроить ключ аутентификации и идентификатор ключа на маршрутизаторе-получателе и маршрутизаторе-отправителе.

Для создания цепочки ключей на маршрутизаторе используется команда **key-chain**:

```
Router(config)# key chain name-of-chain
Router(config-keychain)# key id-key
Router(config-keychain-key)# key-string key
```

Параметры	Описание
name-of-chain	Имя цепочки ключей аутентификации, из которой необходимо извлечь ключ
id-key	Идентификационный номер ключа аутентификации в цепочке ключей. Диапазон номеров ключей: 0 – 2147483647. Идентификационные номера не обязательно должны быть последовательными
key	Строка, используемая для аутентификации принимаемых и отправляемых пакетов EIGRP. Строка может содержать от 1 до 80 буквенно-цифровых символов в верхнем или нижнем регистре. Первый символ не может быть цифрой, строка вводится с учетом регистра

Для настройки аутентификации используются следующие команды. Команда **ip authentication key-chain eigrp** *as-number name-of-chain* указывает на цепочку ключей, которая будет использована для аутентификации. Команда **ip authentication mode eigrp** *as-number* **md5** указывает, что в качестве метода аутентификации будет использоваться MD5.

Синтаксис команды ip eigrp authentication key-chain:

```
Router(config-if)# ip authentication key-chain eigrp as-number name-of-chain
```

Параметры	Описание
as-number	Номер автономной системы EIGRP, для которой будет использоваться аутентификация
name-of-chain	Имя цепочки ключей аутентификации, из которой необходимо извлечь ключ

Синтаксис команды **ip authentication mode eigrp md5**:

```
Router (config-if)# ip authentication mode eigrp as-number md5
```

Пример настройки аутентификации:

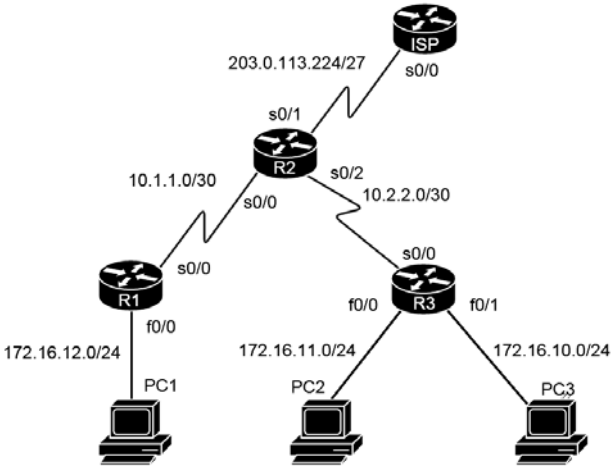
```
    Router(config)# key chain Router-chain
    Router(config-keychain)# key 1
    Router(config-keychain-key)# key-string
firstkey
    Router(config-keychain)# key 2
    Router(config-keychain-key)# key-string
secondkey
    …
    Router(config)# interface Serial0/0/1
    Router(config-if)# ip authentication mode eigrp
100 md5
    Router(config-if)# ip authentication key-chain
eigrp 100 Router-chain
```

Контрольные вопросы

1. Принципы настройки и проверки протокола EIGRP.
2. Принципы настройки распределения нагрузки в протоколе EIGRP.
3. Принципы настройки и проверки аутентификации EIGRP.
4. Принципы поиска и устранения ошибок конфигурации протокола EIGRP.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить адресацию устройств согласно таблице:

Устройство	Интерфейс	IP адрес	Маска	Шлюз
R1	S0/0	10.1.1.12	255.255.255.252	–
	F0/0	172.16.12.1	255.255.255.0	–
R2	S0/0	10.1.1.	255.255.255.252	–
	Lo0	203.0.113.225	255.255.255.255	–
	S0/2	10.2.2.1	255.255.255.252	–
R3	S0/0	10.2.2.2	255.255.255.252	–
	Fa0/0	172.16.10.1	255.255.255.0	–
	Fa0/1	172.16.11.1	255.255.255.0	–
PC1	NIC	172.16.12.10	255.255.255.0	172.16.12.1
PC2	NIC	172.16.11.10	255.255.255.0	172.16.11.1
PC3	NIC	172.16.10.10	255.255.255.0	172.16.10.1

3. Настроить протокол динамической маршрутизации EIGRP:

– объявить все сети;

– проверить доступность всех адресов;

– на маршрутизаторе R2 настроить маршрут по умолчанию и объявить его как маршрут по умолчанию для всех маршрутизаторов в сети.

Практическое занятие 4

ПРОТОКОЛ OSPF

Цель: изучение особенностей функционирования и настройки протокола OSPF.

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: принципы настройки и проверки протокола OSPF, распределения нагрузки в протоколе OSPF, настройки и проверки аутентификации OSPF; принципы поиска и устранения ошибок конфигурации протокола OSPF;

– *уметь* настраивать и устранять неполадки в работе протокола OSPF в сетях IPv4 и IPv6.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.
2. Основная часть – устный или письменный опрос, решение предложенных задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 500 – 535], [5], [7].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Протокол OSPF (Open Shortest Path First) представляет собой протокол маршрутизации по состоянию канала. Является открытым. Протокол OSPF имеет ряд значительных преимуществ, обеспечивая быструю сходимость и возможность масштабирования в целях реализации сетей большего размера. Существует несколько версий протокола OSPF, в настоящее время широкое распространение получила вторая версия протокола – OSPF v2. Маршруты, полученные по протоколу OSPF, в таблице маршрутизации помечаются буквой O. Административное расстояние протокола – 110.

Характеристики протокола OSPF. *Быстрая сходимость*. Благодаря отсутствию периодической рассылки обновлений маршрутной информации маршрутизатор, обнаруживший изменения в топологии сети, незамедлительно оповещает об этом все соседние маршрутизаторы.

Групповая рассылка обновлений. В протоколе OSPF рассылка топологической информации о состоянии каналов связи осуществляется по групповому адресу 224.0.0.5 для всех маршрутизаторов OSPF и по адресу 224.0.0.6 для назначенного и резервного назначенного маршрутизатора.

Бесклассовая маршрутизация – протокол разработан как бесклассовый, следовательно, он поддерживает использование VLSM и маршрутизации CIDR.

Безопасность – поддерживает аутентификацию Message Digest 5 (MD5). Если эта функция включена, маршрутизаторы OSPF принимают только зашифрованные сообщения маршрутизации от равноправных узлов с одинаковым предварительно заданным паролём.

Экономия пропускной способности каналов связи Протокол OSPF производит периодическую рассылку информации базы данных топологии сети передачи данных через длительные промежутки времени – 30 минут.

Иерархическое разделение сети передачи данных. Протокол OSPF позволяет произвести иерархическое разделение сети передачи данных на несколько зон (Area) с целью уменьшения нагрузки на маршрутизаторы внутри каждой зоны.

Для распространения обновлений о состоянии каналов передачи данных OSPF маршрутизаторы не используют широковещательные рассылки. Вместо этого они применяют групповую рассылку по зарезервированным для протокола OSPF групповым IP адресам.

Протокол OSPF использует два основных групповых адреса: 224.0.0.5 – для всех маршрутизаторов OSPF и 224.0.0.6 – адрес для назначенного и резервного назначенного маршрутизатора. Маршрутизатор, на котором активизирован протокол OSPF, автоматически становится членом группы многоадресной рассылки с адресом 224.0.0.5 и начинает рассылать и получать групповые сообщения OSPF.

В широковещательных сетях выбирается назначенный маршрутизатор (Designated Router) – DR и резервный назначенный маршрутизатор (Backup Designated Router) – BDR. Оба эти маршрутизатора с момента принятия на себя таких функций становятся членами группы многоадресной рассылки с адресом 224.0.0.6 и начинают принимать групповые сообщения, посылаемые на этот адрес всеми остальными маршрутизаторами OSPF, принадлежащими тому же широковещательному домену.

Рассылка обновлений. Протокол OSPF производит рассылку обновлений о состоянии канала связи сразу после обнаружения изменений в его состоянии. Маршрутизатор отслеживает каждое изменение и рассылает сообщение о состоянии канала (Link State Advertisement) – LSA.

Сообщения LSA рассылаются всем соседним маршрутизаторам, в свою очередь каждый маршрутизатор, получивший LSA, производит обновление своей базы данных топологии сети и производит дальнейшую рассылку LSA всем своим соседям. Такая рассылка называется лавинной, и она информирует все маршрутизаторы о произошедших изменениях в топологии сети, а также о возможной необходимости внесения изменений в таблицу маршрутизации с целью отражения в ней изменений топологии.

Все маршрутизаторы, поддерживающие OSPF, сети и подсети логически объединены в зоны (Area). Сети передачи данных, в которых применяется протокол OSPF, могут составлять одну зону или включать множество зон, организованных по иерархическому признаку. Объединённая сеть передачи данных, использующая протокол OSPF, независимо от того, состоит ли она из одной зоны или включает множество зон, представляет собой один домен маршрутизации, или другими словами одну автономную систему (AS). Такая иерархическая структура позволяет локализовать изменения маршрутов и трафик маршрутных обновлений в пределах каждой зоны, что позволяет уменьшить нагрузку на каналы связи.

В небольших сетях количество каналов связи межу маршрутизаторами не столь велико и расчёт маршрутов для каждой сети получателя не столь сложен. Однако, в больших сетях, где присутствует значительно большее количество каналов связи между маршрутизаторами и число потенциальных маршрутов велико, работа алгоритма SPF требует достаточного большого промежутка времени и значительных вычислительных возможностей маршрутизатора. Протокол OSPF для уменьшения числа расчётов применяют разделение сети передачи данных на зоны. Число маршрутизаторов в каждой зоне, а также число LSA в пределах зоны не велико, следовательно, база данных состояния каналов в пределах зоны значительно меньше. Поэтому расчёт маршрутов становится легче и занимает меньше времени. Различается два основных типа зон.

1. *Транзитная зона.* Главная задача транзитной зоны быстрое и эффективное продвижение пакетов в другие зоны. В транзитной зоне не рекомендуется размещать пользовательские сети, хотя это не запрещено в спецификации. В протоколе OSPF в качестве транзит-

ной зоны применяется зона с номером 0 (Area 0), также именуемая базовой (Backbone Area).

2. *Регулярные зоны***.** В протоколе OSPF зоны, чья основная задача – подключение пользователей, называются регулярными. Регулярные зоны устанавливаются, исходя из функциональных или географических группировок. По умолчанию регулярные зоны не пропускают трафик из других зон. Весь трафик из других зон проходит через транзитную зону.

Применение протокола OSPF вынуждает применять жёсткую двухуровневую иерархию сети передачи данных. Все регулярные зоны должны иметь соединение с базовой зоной.

Протокол OSPF разграничивает функции маршрутизаторов в зависимости от того, какое место в домене маршрутизации они занимают и к какому числу зон принадлежат.

Протокол OSPF использует четыре типа маршрутизаторов:

1) внутренние маршрутизаторы (Internal router);

2) магистральные маршрутизаторы (Backbone router);

3) пограничные маршрутизаторы зоны (Area Border Router – ABR);

4) пограничные маршрутизаторы автономной системы (Autonomous System Boundary Router – ASBR).

Внутренние маршрутизаторы – это маршрутизаторы, все интерфейсы которых находятся в одной зоне протокола OSPF. Магистральные маршрутизаторы – это тип маршрутизаторов, которые находятся в магистральной зоне и имеют, по крайней мере, один интерфейс, подключённый к зоне 0. Пограничные маршрутизаторы – это маршрутизаторы, интерфейсы которых подключены как минимум к двум разным зонам, одна из которых обязательно должна быть магистральной зоной. Маршрутизаторы ASBR располагаются на границе двух и более автономных систем, в которых могут быть запущены различные протоколы маршрутизации.

База данных протокола OSPF. Все маршрутизаторы OSPF создают и поддерживают в своей базе данных две основные таблицы.

1. *Таблица соседства.* Все маршрутизаторы OSPF ведут таблицу соседства, в которой хранится список и вся необходимая информация о соседних OSPF маршрутизаторах.

2. *Таблица топологии.* Каждый маршрутизатор OSPF ведёт таблицу топологии, которая содержит необходимую информацию о состоянии всех сетей, подсетей и маршрутизаторов в пределах зоны OSPF. Если маршрутизатор OSPF имеет подключение к двум и более зонам, то он ведёт отдельную таблицу топологии для каждой из зон OSPF, к которой он подключён.

Таблица соседства. Чтобы начать обмен топологической информацией, маршрутизаторы OSPF, находящиеся в одном и том же сегменте сети в пределах одной зоны OSPF, должны сформировать соседские взаимоотношения. Маршрутизаторы становятся соседями после того, как они обменяются приветственными пакетами. Когда маршрутизатор OSPF находится в процессе инициализации, он должен распознать все соседние OSPF маршрутизаторы и установить с ними соседские взаимоотношения. Этот процесс называется процессом обнаружения соседей. Каждый маршрутизатор в результате обмена приветственными сообщениями создаёт локальную таблицу соседей, в дальнейшем отслеживая всех своих соседей и их состояния.

Вывод команды **show ip ospf neighbor**:

```
    Neighbor ID Pri State Dead Time Address
Interface
    2.2.2.2 0 FULL/ - 00:00:36 172.16.3.2
Serial0/0/0
    3.3.3.3 0 FULL/ - 00:00:39 192.168.10.6
Serial0/0/1
```

Ниже описаны поля, содержащиеся в таблице соседства.

Идентификатор соседа (*Neighbor ID*). Уникальное число, идентифицирующее соседний маршрутизатор.

Приоритет маршрутизатора (*Pri*). Приоритет соседнего маршрутизатора. Это значение используется при выборе маршрутизаторов DR и BDR.

Состояние (*State*). Состояние соседских отношений.

Время до разрыва соседских отношений (*Dead Time*). Временной интервал, по истечении которого будут разорваны соседские отношения, если до его окончания не придет ни одного пакета OSPF от данного соседа.

Адрес соседнего маршрутизатора (Address)**.** Адрес сетевого уровня соседнего маршрутизатора.

Интерфейс (*Interface*). Локальный интерфейс маршрутизатора за которым находится сосед.

Таблица топологии. Все маршрутизаторы OSPF должны создавать и поддерживать в актуальном состоянии таблицу топологии. Эта таблица представляет собой топологическую карту зоны OSPF, в которой находится маршрутизатор. Процесс создания и поддержки в актуальном состоянии таблицы топологии является результатом обмена информацией об элементах топологии. В качестве элементов топологии выступают маршрутизаторы, сети-получатели, суммарные маршруты и другая топологическая информация. Обмен топологической

информацией начинается после завершения установки соседских отношений между смежными OSPF маршрутизаторами. Вывод, приведённый в примере, представляет собой таблицу топологии, созданную в результате обмена топологической информацией по протоколу OSPF. В примере имеются записи о четырёх маршрутизаторах, принадлежащих той же зоне, что и маршрутизатор, рассматриваемый в примере. Также имеются записи о четырёх сетях и о двух суммарных маршрутах в сеть 0.0.0.0.

Вывод команды **show ip ospf database**:

```
    OSPF Router with ID (10.95.56.58) (Process ID 2)
                Router Link States (Area 0)
Link ID         ADV Router  Age  Seq#       Checksum Link
                                                    count

10.95.56.33     10.95.56.33 60   0x8000127F 0x00BE67 2
10.95.56.34     10.95.56.34 1837 0x8000127D 0x00DD37 2
10.95.56.58     10.95.56.58 640  0x80001284 0x00E368 6
10.95.56.59     10.95.56.59 1677 0x8000127C 0x00E956 6
                  Net Link States (Area 0)
Link ID         ADV Router  Age  Seq#       Checksum
10.93.254.2     10.95.56.33 1606 0x80000207 0x0082D9
10.93.255.158 10.95.56.33 1606 0x80000207 0x008E14
10.93.254.2     10.95.56.34 1606 0x80000204 0x0080D9
10.93.255.158 10.95.56.34 1606 0x80000204 0x008014
             Summary Net Link States (Area 0)
Link ID         ADV Router  Age  Seq#       Checksum
0.0.0.0         10.95.56.33 60   0x80001278 0x00E60B
0.0.0.0         10.95.56.34 1837 0x80001278 0x00E010
```

Ниже описаны поля, содержащиеся в таблице топологии.

Идентификатор топологического элемента (*Link ID*) – уникальное число идентифицирующее топологический элемент.

Маршрутизатор (*ADV Router*) – маршрутизатор, объявивший топологический элемент.

Возраст (*Age*) – время существования топологического элемента.

Номер последнего LSA (*Seq#*) – последовательный номер последнего пришедшего LSA о данном топологическом элементе.

Контрольная сумма (*Checksum*) – контрольная сумма последнего LSA.

Число интерфейсов (*Link count*) – количество интерфейсов маршрутизатора, на которых разрешён процесс OSPF.

В протоколе OSPF топология сети описывается, хранится и передаётся в виде сообщений LSA. Содержимое LSA описывает отдельный топологический элемент сети, такой как маршрутизатор, сеть или суммарный маршрут. Как существуют разные типы элементов топологии сети, имеются и разные типы сообщений LSA, каждый из которых соответствует отдельному типу компонентов сети. Маршрутизаторы OSPF создают новую топологическую информацию или производят изменения существующей только после изменений в топологи сети передачи данных.

Когда маршрутизатор объявляет новое сообщение LSA или изменяет существующие, он должен передать его всем своим соседям. По получении нового или обновлённого LSA соседи сначала сохраняют его в своих базах данных, а затем передают его далее своим соседям.

Информация о топологических элементах должна быть синхронизирована между всеми маршрутизаторами, для чего необходимо выполнение следующих условий:

1) достижение надёжной рассылки LSA благодаря применению механизма отправки подтверждений о получении LSA;

2) рассылка LSA производиться последовательно по всем маршрутизаторам входящим в зону или по всему домену маршрутизации, если не применяется разделение на зоны OSPF;

3) сообщения LSA имеют порядковые номера, чтобы каждый маршрутизатор мог сравнить порядковый номер поступившего LSA с уже имеющемся в его базе данных, и при необходимости обновить её.

Благодаря гарантированной рассылке сообщений LSA, каждый маршрутизатор в пределах зоны или домена маршрутизации может гарантировать, что он имеет последнюю и самую точную информацию о топологии сети. Только в данном случае маршрутизатор имеет возможность расчёта достоверных маршрутов до всех сетей получателей.

В протоколах маршрутизации по состоянию канала должно проводиться периодическое обновление записей таблицы топологии для актуализации, имеющейся в ней информации. В протоколе OSPF по умолчанию интервал обновления информации таблицы топологии составляет 30 минут. Интервал рассылки устанавливается не на всю таблицу топологии, а на каждую отдельно взятую запись из таблицы.

По истечении 30 минут маршрутизатор производит рассылку обновлённых LSA сообщений, у которых параметр Seq увеличен на единицу. При получении LSA каждый маршрутизатор OSPF выполняет действия по следующему алгоритму:

1. Если поступившее LSA не присутствует в базе данных состояния каналов или имеет больший порядковый номер:

– маршрутизатор добавляет LSA в свою таблицу топологии;

– посылает подтверждение о получении LSA;

– производит рассылку полученного LSA своим соседям за исключением того, от которого это LSA было получено;

– производит обновление таблицы маршрутизации.

2. Если поступившее LSA присутствует в базе данных состояния каналов и имеет тот же порядковый номер, то поступившее LSA игнорируется.

3. Если поступившее LSA присутствует в базе данных состояния каналов, но имеет меньший порядковый номер, маршрутизатор посылает отправителю последнюю версию данного LSA.

Метрика протокола OSPF. Протокол OSPF для оценки маршрутов в отличие от протокола EIGRP использует не комбинированную метрику, а простую метрику, зависящую от ширины полосы пропускания канала связи.

Метрика протокола OSPF рассчитывается по формуле

$$Metric = \frac{10^8}{bandwitch},$$

где *bandwitch* – ширина полосы пропускания канала связи, выраженная в бит/с.

Для протокола OSPF каналы связи со скоростями выше 100 Мбит/с будут иметь одинаковую метрику равную 1, так как в протоколе OSPF метрика меньше 1 не существует.

Для решения данной проблемы необходимо использовать команду **auto-cost reference-bandwidth** *Mbps*. Обратите внимание, что значение выражено в Мбит/с:

```
Router(config-router)# auto-cost reference-bandwidth 1000
```

Следует отметить, что при необходимости изменения константы для расчёта метрики каналов связи в протоколе OSPF, данные изменения необходимо производить на всех маршрутизаторах, входящих в домен маршрутизации.

Вместо настройки пропускной способности по умолчанию можно вручную настроить на интерфейсе значение стоимости, используя команду конфигурации интерфейса **ip ospf cost** *значение*.

Функционирование протокола. Все служебные пакеты OSPF инкапсулируются непосредственно в протокол IP. Протокол OSPF использует в своей работе пять типов служебных пакетов, описание которых приводится в таблице.

Тип	Назначение пакета
Hello (1)	Пакеты приветствия используются для поиска соседей и дальнейшего поддержания отношения смежности
DBD (2)	Пакет описания базы данных содержит сокращённый список базы данных состояний каналов отправляющего маршрутизатора. Используется принимающими маршрутизаторами для сверки с локальной базой данных о состоянии канала
LSR (3)	Запрос на получение информации о топологическом элементе
LSU (4)	Обновление информации о топологических элементах. Может содержать один или несколько LSA. Пакеты обновления состояния канала поддерживают семь различных типов LSA
LSAck (5)	Подтверждение получения пакетов обновлений

Процесс установки соседских отношений. Соседские отношения между маршрутизаторами устанавливаются в случае, если оба маршрутизатора принадлежат одной и той же зоне. Во время процесса установки соседских отношений маршрутизаторы OSPF последовательно проходят следующие семь состояний:

1) нерабочее (Down);
2) инициализация (Init);
3) двунаправленные отношения (Two-Way);
4) выборы DR и BDR (ExStart);
5) обмен (Exchange);
6) загрузка (Loading);
7) полные соседские отношения (Full).

В зависимости от типа канала связи между маршрутизаторами процесс установки соседских отношений может не содержать некоторые из этапов. Процесс установки соседских отношений можно разбить на две основных части:

1) поиск соседей;
2) обмен топологической информацией.

Пакеты приветствия OSPF передаются на групповой адрес 224.0.0.5 по умолчанию каждые 10 секунд в сетях с множественным доступом и сетях типа «точка-точка» и каждые 30 секунд в не широковещательных сетях множественного доступа, например, Frame Relay.

Интервал простоя является интервалом времени в секундах, в течение которого маршрутизатор ожидает получения пакета приветствия перед тем, как объявить соседнее устройство неработающим. Если интервал простоя истекает до получения маршрутизаторами пакета приветствия, OSPF удаляет это соседнее устройство из своей базы данных состояний каналов. Маршрутизатор выполняет лавинную рассылку базы данных состояний каналов, содержащей данные о неработающем соседнем устройстве, из всех интерфейсов, использующих OSPF. По умолчанию в устройствах Cisco интервал простоя равен четырём интервалам отправки Hello-пакетов: 40 секунд в сетях с множественным доступом и сетях типа «точка-точка» и 120 секунд в нешироковещательных сетях множественного доступа.

Если протокол OSPF активирован, интерфейс, на котором запущен процесс OSPF, переходит из состояния Down в состояние Init. Когда маршрутизатор принимает пакет приветствия, содержащий его идентификатор в списке соседних устройств, он переходит из состояния Init в состояние Two-Way. Действие, выполняемое в состоянии Two-Way, определяется типом взаимодействия между смежными маршрутизаторами. Если два смежных соседних устройства взаимодействуют посредством канала типа «точка-точка», они немедленно переходят из состояния Two-Way в фазу синхронизации базы данных. Если маршрутизаторы взаимодействуют посредством общей сети Ethernet, необходимо выбрать выделенный маршрутизатор (DR) и резервный выделенный маршрутизатор (BDR).

Для выбора DR и BDR маршрутизаторов рассматриваются два параметра:

1) приоритет маршрутизатора;

2) идентификатор маршрутизатора (RouterID).

Сначала маршрутизаторы рассматривают приоритеты маршрутизаторов, а затем идентификатор маршрутизатора. Маршрутизатор с наивысшим приоритетом становится DR маршрутизатором, маршрутизатор со следующим после него значением приоритета становится BDR маршрутизатором. Все остальные устройства получают статус DROther.

Настройка приоритета:

```
Router(config-if)# ip ospf priority number
```

Значение *number* задаёт приоритет маршрутизатора в диапазоне от 0 до 255. По умолчанию приоритет задаётся равным 1.

Если же все маршрутизаторы имеют одинаковые значения приоритета, то DR маршрутизатором становится маршрутизатор с наи-

высшим значением RouterID, а маршрутизатор со следующим по значению RouterID соответственно BDR маршрутизатором.

Маршрутизаторы определяют свой идентификатор на основе одного из трёх критериев в следующем порядке предпочтения.

Идентификатор маршрутизатора настраивается напрямую посредством команды режима глобальной конфигурации OSPF **router-id** *id-value*. Значение *id-value* является любым 32-битным значением, выраженным как IPv4-адрес. Данный метод является рекомендуемым для назначения идентификатора маршрутизатора.

```
Router(config)# router ospf 10
Router(config-router)# router-id 10.10.10.10
```

Если идентификатор маршрутизатора не настроен напрямую, маршрутизатор выбирает самое высокое значение IPv4-адреса любого из настроенных интерфейсов loopback. Это второй способ назначения идентификатора маршрутизатора.

```
Router(config)# interface loopback 0
Router(config-router)# ip address 10.10.10.10
255.255.255.255
```

При отсутствии настроенных интерфейсов loopback маршрутизатор выбирает самое высокое значение активного IPv4-адреса любого из своих физических интерфейсов. Данный метод не рекомендуется использовать, так как в этом случае администратору сложнее различать маршрутизаторы.

Для проверки идентификатора маршрутизатора используется команда **show ip protocols** или **show ip ospf**.

В состоянии ExStart между маршрутизаторами и их смежными маршрутизаторами DR и BDR устанавливаются отношения ведущего и ведомых устройств. Маршрутизатор с более высоким значением идентификатора выступает в роли ведущего устройства в состоянии Exchange.

В состоянии Exchange ведущие и ведомые маршрутизаторы обмениваются одним или несколькими пакетами DBD. Пакет DBD включает информацию о заголовке записи LSA, которая отображается в базе данных состояний каналов маршрутизатора. Записи могут содержать данные о канале или о сети. Каждый заголовок записи LSA содержит данные о типе состояния канала, адресе объявляющего маршрутизатора, стоимости канала и порядковом номере. Маршрутизатор использует порядковый номер для определения актуальности полученных данных о состоянии канала.

Маршрутизатор сравнивает полученные данные с данными, которые содержатся в его собственной базе данных состояний каналов. Если пакет DBD содержит более актуальную запись о состоянии канала, маршрутизатор переходит в состояние Loading. Маршрутизатор отправляет пакет LSR с запросом. Маршрутизатор-сосед отправляет отклик, содержащий полные данные о необходимой сети в пакете LSU. Когда маршрутизатор принимает пакет LSU, он в ответ отправляет пакет LSAck. Маршрутизатор добавляет новые записи о состоянии канала в свою базу данных состояний каналов.

После того как на все пакеты LSR для данного маршрутизатора отправлен отклик, смежные маршрутизаторы считаются синхронизированными и переведёнными в состояние Full.

Пока соседние маршрутизаторы продолжают получать пакеты приветствия, данные о сети, содержащиеся в переданных пакетах LSA, остаются в базе данных топологии. После синхронизации топологических баз данных пакеты обновлений (LSU) отправляются соседним устройствам только в следующих случаях:

1) получение изменений (инкрементные обновления);
2) по истечении 30 минут.

Настройка протокола OSPF в одной зоне. Для запуска протокола OSPF используется команда **router ospf** *process-id*. Параметр *process-id* представляет собой номер локального процесса маршрутизации OSPF запущенного на маршрутизаторе в диапазоне от 1 до 65535. Параметр *process-id* имеет локальное значение и может не совпадать на маршрутизаторах, принадлежащих зоне или домену маршрутизации OSPF. Однако в современных сетях передачи данных на маршрутизаторах может быть запущено несколько процессов маршрутизации OSPF, поэтому хорошим тоном считается использовать один и тот же *process-id* на всех маршрутизаторах домена маршрутизации, на которых запущен один и тот же экземпляр маршрутизатора OSPF.

Команда **network** определяет интерфейсы, участвующие в процессе маршрутизации для области OSPF.

Все интерфейсы на маршрутизаторе, соответствующие сетевому адресу в рамках команды **network**, включены и готовы к отправке и приёму пакетов OSPF. В результате адрес сети интерфейса включается в обновления маршрутизации OSPF.

Базовая команда синтаксиса – **network** *network-address wildcard-mask* **area** *area-id*.

Синтаксис команды **area** *area-id* относится к области OSPF. При настройке OSPF для одной области на всех маршрутизаторах необхо-

димо настроить команду **network** с одинаковым значением *area-id*. Несмотря на то, что можно использовать любой идентификатор области, для OSPF с одной областью рекомендуется использовать идентификатор 0. Такое условное обозначение упрощает включение поддержки OSPF для нескольких областей в случае изменений сети в будущем.

```
    Router(config)# router ospf 10
    Router(config-router)# network 172.16.0.0
0.0.0.255 area 0
    Router(config-router)# network 192.168.0.4
0.0.0.3 area 0
```

Аутентификация. Протокол OSPF обеспечивает аутентификацию соседних маршрутизаторов при передаче обновлений о состоянии каналов передачи данных. Аутентификация маршрутизаторов может осуществляться как при помощи передачи пароля в виде открытого текста, так и при помощи MD5.

Для задания типа аутентификации используется команда **ip ospf authentication**.

Синтаксис команды:

```
    Router(config-if)# ip ospf authentication
[message-digest | null]
```

Параметры	Описание
`message-digest`	Аутентификация с помощью MD5
`null`	Отсутствие аутентификации

При использовании команды **ip ospf authentication** без параметров будет использована аутентификация по паролю.

Для задания текстовой строки, используемой при аутентификации по паролю, используется команда **ip ospf authentication-key**. Синтаксис команды

```
    Router(config-if)# ip ospf authentication-key
password
```

В качестве параметра команды выступает строка, которая будет использоваться как пароль. Необходимо отметить, что для аутентификации будет использоваться только первые восемь символов, а остальные будут отброшены. Пример настройки аутентификации:

```
    Router(config)#interface serial 0
    Router(config-if)#ip address 172.16.1.1
255.255.255.252
    Router(config-if)#ip ospf authentication
    Router(config-if)# ip ospf authentication-key
PassWord
```

Для задания текстовой строки, используемой при аутентификации с помощью MD5, используется команда:

```
    Router(config-if)# ip ospf message-digest-key
key-id encryption-type md5 key
```

Параметры	Описание
key-id	Номер используемого ключа
encryption-type	Тип вводимой строки: 0 – нешифрованная строка; 7 – шифрованная средствами IOS
key	Строка для аутентификации

Пример настройки аутентификации с помощью MD5:

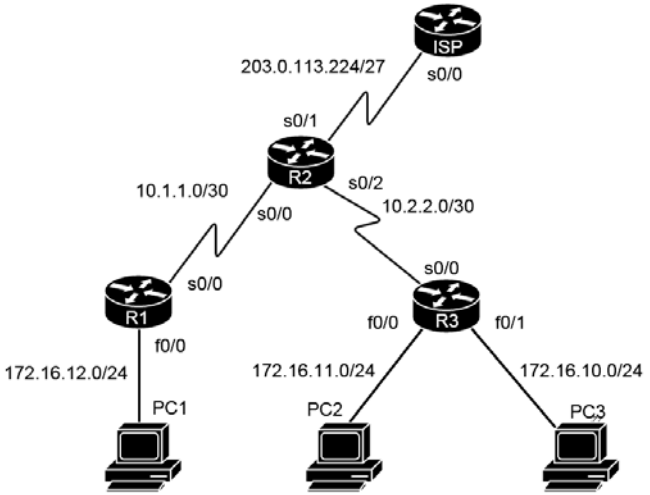
```
    Router(config)#interface serial 0
    Router(config-if)#ip address 172.16.1.1
255.255.255.252
    Router(config-if)# ip ospf authentication
message-digest
    Router(config-if)# ip ospf message-digest-key
1 md5 SeCrEt
```

Контрольные вопросы

1. Принципы настройки и проверки протокола OSPF.
2. Принципы настройки распределения нагрузки в протоколе OSPF.
3. Принципы настройки и проверки аутентификации OSPF.
4. Принципы поиска и устранения ошибок конфигурации протокола OSPF.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить адресацию устройств согласно таблице:

Устройство	Интерфейс	IP адрес	Маска	Шлюз
R1	S0/0	10.1.1.12	255.255.255.252	–
	F0/0	172.16.12.1	255.255.255.0	–
R2	S0/0	10.1.1.	255.255.255.252	–
	Lo0	203.0.113.225	255.255.255.255	–
	S0/2	10.2.2.1	255.255.255.252	–
R3	S0/0	10.2.2.2	255.255.255.252	–
	Fa0/0	172.16.10.1	255.255.255.0	–
	Fa0/1	172.16.11.1	255.255.255.0	–
PC1	NIC	172.16.12.10	255.255.255.0	172.16.12.1
PC2	NIC	172.16.11.10	255.255.255.0	172.16.11.1
PC3	NIC	172.16.10.10	255.255.255.0	172.16.10.1

3. Настроить протокол динамической маршрутизации OSPF:

– объявить все сети;

– проверить доступность всех адресов;

– на маршрутизаторе R2 настроить маршрут по умолчанию и объявить его как маршрут по умолчанию для всех маршрутизаторов в сети.

Практическое занятие 5

ТЕХНОЛОГИЯ VLAN

Цель: изучение особенностей функционирования и настройки технологии VLAN.

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: принципы настройки и проверки VLAN, протокола VTP; принципы поиска и устранения ошибок конфигурации VLAN и протокола VTP;

– *уметь*: настраивать коммутацию с использованием технологий VLAN; маршрутизацию между VLAN; настраивать и устранять неполадки в работе протокола VTP.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [3, с. 54 – 85]

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Virtual Local Area Network, сети VLAN позволяют администратору производить сегментацию по функциям, проектным группам или областям применения вне зависимости от физического расположения пользователя или устройства. Устройства в пределах сети VLAN работают таким образом, будто находятся в собственной независимой сети, даже если делят одну общую инфраструктуру с другими VLAN. Любой порт коммутатора может принадлежать сети VLAN. Одноадресные, широковещательные и многоадресные пакеты пересылаются и рассылаются только к конечным станциям в пределах той сети VLAN,

которая является источником этих пакетов. Каждая сеть VLAN считается отдельной логической сетью, и пакеты, адресованные станциям, не принадлежащим данной сети VLAN, должны пересылаться через устройство, поддерживающее маршрутизацию.

Ниже приведены основные преимуществам использования VLAN.

Безопасность – потоки трафика разных сетей VLAN изолированы друг от друга, благодаря чему снижается вероятность утечки конфиденциальной информации.

Снижение расходов – благодаря экономии на дорогих обновлениях сетевой инфраструктуры и более эффективному использованию имеющейся полосы пропускания и восходящих каналов происходит снижение расходов.

Повышение производительности – разделение однородных сетей 2-го уровня на несколько логических рабочих групп (широковещательных доменов) уменьшает количество лишнего сетевого трафика и повышает производительность.

Уменьшение широковещательных доменов – разделение единой сети на сети VLAN уменьшает количество устройств в каждом широковещательном домене.

Упрощение администрирования – сети VLAN упрощают управление сетью, поскольку пользователи с аналогичными требованиями к сети используют одну и ту же сеть VLAN. При введении в эксплуатацию нового коммутатора на назначенных портах реализуются все правила и процедуры, уже применённые в этой конкретной VLAN.

Каждая VLAN в коммутируемой сети относится к какой-либо IP-сети. Таким образом, в проекте VLAN нужно учитывать реализацию иерархической системы сетевой адресации. Иерархическая адресация подразумевает упорядоченное назначение номеров IP-сети сегментам или сетям VLAN с учётом работы сети в целом.

В современных сетях используется множество различных типов сетей VLAN. Некоторые типы VLAN определяются классами трафика. Другие типы VLAN обусловлены функциями, которые они выполняют.

Наиболее распространённый стандарт работы сетей VLAN – IEEE 802.1Q. Существуют проприетарные протоколы, решающие те же задачи, например, протокол ISL от Cisco.

Типы VLAN. *Data VLAN* (VLAN данных) – сеть VLAN, в которой передаются данные, генерируемые оконечным оборудованием.

Но это не могут быть голосовые данные или управляющий трафик для коммутаторов. Рекомендуется отделять голосовой и управляющий трафик от трафика данных.

Default VLAN (VLAN по умолчанию) – сеть VLAN, в которой находятся все порты коммутатора перед началом конфигурирования. VLAN по умолчанию для коммутаторов Cisco – VLAN 1. По умолчанию управляющий трафик второго уровня, такой как CDP и STP, ассоциируется с VLAN 1.

Native VLAN (родной VLAN) – сеть VLAN, через которую на порту, сконфигурированном как транковый порт и пропускающем тегированный трафик, сможет подключиться рабочая станция и передавать нетегированные фреймы. Это реализовано для того, чтобы к транковому порту можно было подключить оборудование, которое не поддерживает работу с тегами. Рекомендуется использовать native VLAN, отличную от VLAN 1.

Management VLAN (управляющая VLAN) – сеть VLAN, настроенная для доступа к функциям управления коммутатора. Для создания управляющей VLAN на коммутаторе создаётся интерфейс SVI (Switched Virtual Interface), ассоциированный с этой VLAN. Интерфейсу SVI назначаются IP-адрес и маска подсети, благодаря чему коммутатором можно управлять с использованием протоколов HTTP, Telnet, SSH или SNMP. В устаревших версиях Cisco IOS коммутаторы поддерживали только один SVI. В версиях ОС Cisco IOS 15.x и выше для коммутаторов Catalyst возможна настройка более одного активного интерфейса SVI.

Voice VLAN (голосовая VLAN) – сеть VLAN, предназначенная для передачи голоса. Такая сеть VLAN должна обеспечивать:

– гарантированную полосу пропускания для обеспечения качества передачи голоса;

– приоритет голосового трафика по сравнению с другими типами трафика;

– возможность направлять трафик в обход загруженных участков в сети;

– иметь задержку менее 150 миллисекунд в сети.

Тег VLAN. Поле тега сети VLAN состоит из поля типа, поля приоритета, поля идентификатора канонического формата и поля идентификатора VLAN (рис. 11).

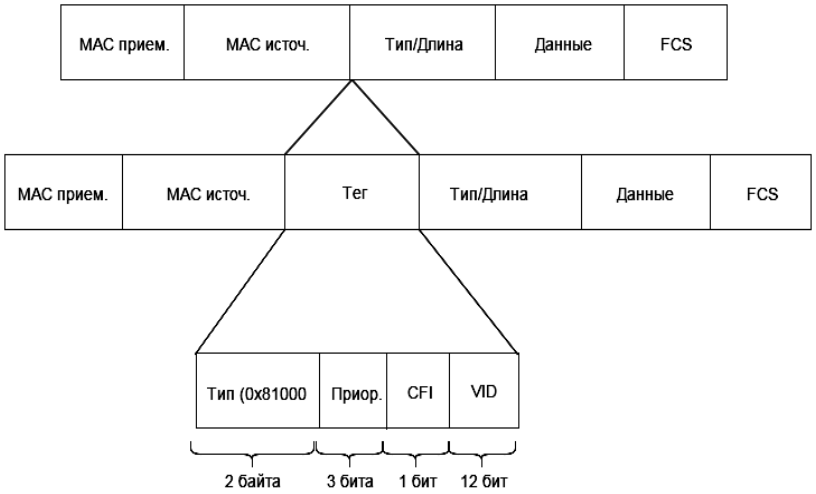


Рис. 11. Структура тега VLAN

Тип – это 2-байтовое значение, которое называется значением идентификатора протокола тегирования (TPID). Значение для Ethernet имеет вид шестнадцатеричного числа 0x8100.

Приоритет пользователя – это 3-битовое значение, которое поддерживает реализацию уровня или сервиса.

Идентификатор канонического формата (*CFI*) – это 1-битовый идентификатор, который обеспечивает передачу кадров Token Ring по каналам Ethernet.

VLAN-идентификатор (VID) – это 12-битный идентификационный номер сети VLAN. Отсюда максимальный номер идентификатора VLAN – 2^12 = 4096.

После того как коммутатор добавит тег в кадр, он пересчитывает значения FCS и добавляет в кадр новое значение FCS.

Различные коммутаторы Cisco поддерживают разное количество сетей VLAN. Количество поддерживаемых сетей VLAN достаточно велико для удовлетворения потребностей большинства организаций. Например, коммутаторы Catalyst 2960 и 3560 способны поддерживать максимально возможное количество сетей VLAN.

Идентификаторы сетей VLAN состоят из двух диапазонов. VLAN стандартного диапазона имеют идентификатор от 1 до 1005, а сети VLAN расширенного диапазона – от 1006 до 4094.

Стандартный диапазон VLAN:

1. Идентификаторы VLAN имеют номера от 1 до 1005.

2. Используются в малых и средних сетях предприятий и организаций.

3. Идентификаторы с 1002 до 1005 зарезервированы для сетей VLAN Token Ring и FDDI.

4. Идентификаторы 1 и идентификаторы с 1002 до 1005 создаются автоматически и не могут быть удалены.

5. Конфигурация VLAN хранится в файле базы данных VLAN под именем vlan.dat. во флеш-памяти коммутатора.

6. Протокол VTP (транковый протокол VLAN), помогающий управлять конфигурациями VLAN, может распознавать и хранить только сети VLAN стандартного диапазона.

Расширенный диапазон:

1. Идентификаторы VLAN имеют номера от 1006 до 4094.

2. Позволяют операторам связи расширять свою инфраструктуру для большого числа клиентов.

3. Конфигурации VLAN не записываются в файл vlan.dat.

4. Поддерживается меньше функций VLAN, чем VLAN из стандартного диапазона.

5. По умолчанию сохраняются в файл текущей конфигурации.

6. Протокол VTP не поддерживает сети VLAN расширенного диапазона.

Создать VLAN можно командой **vlan** *id*, где *id* – номер VLAN. После создания консоль окажется в режиме конфигурирования VLAN. В этом подрежиме можно задать параметры для VLAN. На практике чаще всего при создании VLAN задаётся только идентификатор и имя.

```
Switch(config)#
Switch(config)# vlan 20
Switch(config-vlan)# name Sale
```

Помимо добавления одного VLAN можно создать группу идентификаторов VLAN, разделённых точками, или диапазон идентификаторов VLAN, разделённых дефисами. Например:

```
Switch(config)# vlan 100,102,105-107
```

Удалить VLAN можно командой **no vlan** *id* в режиме глобального конфигурирования:

```
Switch(config)# no vlan 20
```

При настройке сетей VLAN стандартного диапазона сведения о конфигурации хранятся во флеш-памяти коммутатора в файле под именем vlan.dat. Флеш-память является постоянной, поэтому не требует выполнения команды **copy running-config startup-config**. Однако, поскольку во время создания сетей VLAN на коммутаторе Cisco часто необходимо настраивать и другие параметры, рекомендуется сохранять изменения текущей конфигурации в начальную загрузочную конфигурацию.

Команда **show vlan brief** используется для того, чтобы отобразить содержимое файла vlan.dat.

Следующий шаг после создания сети VLAN – назначение портов сетям VLAN. Порт доступа может одновременно принадлежать только одной сети VLAN. Единственным исключением из этого правила является порт, подключённый к IP-телефону. В этом случае с портом связаны две VLAN: одна для голосового трафика и одна для данных.

Каждый порт коммутатора может быть отнесён к конкретному номеру VLAN различными методами.

Статический метод – каждому интерфейсу коммутатора вручную указывается номер VLAN, к которому он принадлежит.

Динамический метод – каждому порту указывается VLAN, к которому он принадлежит, с помощью сервера VMPS (VLAN Membership Policy Server). С помощью VMPS можно назначить VLAN портам коммутатора динамически на основе MAC-адреса устройства, подключённого к порту. Это удобно, когда вы перемещаете устройство с порта одного коммутатора в сети к порту другого коммутатора в сети: коммутатор динамически назначает сеть VLAN для этого устройства на новом порту.

Порт на коммутаторе Cisco может находиться в одном из режимов:

Режим доступа (*Access*) – порт предназначен для подключения оконечного устройства. Порт принадлежит только одной сети VLAN. Входящий трафик от подключённого к порту устройства маркируется (тегируется) заданным на порту идентификатором VLAN.

Режим пересылки (*Trunk*) – порт предназначен для подключения к другому коммутатору или маршрутизатору. Порт передаёт тегированный трафик. Может передавать трафик как одного, так и нескольких VLAN через один физический интерфейс.

На коммутаторах Cisco можно самому задать режим работы порта (access или trunk), либо использовать автоопределение. При автоопределении режима режим работы порта будет согласовываться с соседним

коммутатором, подключённым к этому порту. Согласование режима порта происходит с использованием протокола DTP (Dynamic Trunking Protocol). Протокол DTP – это запатентованный протокол Cisco.

Автоопределение режима порта задаётся командой **switchport mode dynamic** в режиме конфигурации интерфейса.

```
Switch(config-if)# switchport mode dynamic
[auto | desirable]
```

По умолчанию коммутатор периодически отправляет на удалённый порт DTP-кадр, которым извещает, что он работает как транковый порт. Включается командой **switchport mode trunk**.

Режим **dynamic auto**. Коммутатор периодически отправляет на удалённый порт DTP-кадр, которым извещает, что он готов работать как транковый порт, но не требует работать в транковом режиме. Порт включает транковый режим, если удалённый порт уже работает, как транковый или у него настроен режим **dynamic desirable,** порты согласуют работу в транковом режиме. Если удалённый порт настроен тоже как **dynamic auto**, согласование не проводится – порты работают как порты доступа. Если удалённый порт работает в режиме **access**, порт локального коммутатора тоже переводится в режим **access**.

Режим **dynamic desirable**. Коммутатор периодически отправляет на удалённый порт DTP-кадр, которым извещает, что он готов работать как транковый порт и просит работать в транковом режиме. Если удалённый порт работает в режиме **dynamic auto** или **dynamic desirable**, порты переходят в транковый режим. Если удалённый порт не поддерживает согласование, порт локального коммутатора переводится в **access** режим.

Отключить согласование режимов работы можно командой **switchport nonegotiate**.

Согласование режимов работы интерфейсов:

	dynamic auto	**dynamic desirable**	**trunk**	**access**
dynamic auto	access	trunk	trunk	access
dynamic desirable	trunk	trunk	trunk	access
trunk	trunk	trunk	trunk	–
access	access	access	–	access

На практике рекомендуется режимы работы интерфейсов **access** или **trunk** задавать принудительно командами **switchport mode access** или **switchport mode trunk** в режиме конфигурации интерфейса и отключать передачу DTP-кадров командой **switchport nonegotiate**.

Команда для определения порта в качестве порта доступа **switchport mode access**:

```
Switch(config)# interface interface_id
Switch(config-if)# switchport mode access
```

Команда для назначения интерфейсу идентификатора сети VLAN **switchport access vlan** *vlan_id*:

```
Switch(config)# interface interface_id
Switch(config-if)# switchport access vlan
vlan_id
```

Выполнять команду **switchport mode access** необязательно, но настоятельно рекомендуется в целях обеспечения безопасности. С помощью этой команды интерфейс переводится в режим постоянного доступа. Используя команду **interface range**, можно настроить сразу несколько интерфейсов.

Команда **switchport access vlan** *vlan_id* принудительно создаёт VLAN, если таковая ещё не существует на коммутаторе.

Если на порт в режиме работы **trunk** поступает нетегированный кадр, коммутатор автоматически отправляет его в native VLAN. По умолчанию native VLAN является VLAN 1. Но его можно изменить командой **switchport trunk native vlan** *vlan-id*:

```
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan
vlan-id
```

Для проверки номера native VLAN, заданного на интерфейсе, используется команда **show interfaces** *interface-id* **switchpor**t:

```
Switch#sh interfaces fastEthernet 0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
```

```
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation:
dot1q
Administrative private-vlan trunk normal VLANs:
none
Administrative private-vlan trunk private VLANs:
none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Appliance trust: none
```

По умолчанию через интерфейс, находящийся в режиме **trunk**, разрешена передача трафика всех сетей VLAN. Можно ограничить перечень сетей VLAN, трафик которых может передаваться через конкретный порт. Для этого используется команда **switchport trunk allowed vlan** *vlan-id.*

Пример:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk allowed
vlan 1-2,10,15
```

Добавление ещё одной VLAN к списку разрешённых:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk allowed
vlan add 160
```

Удаление сети VLAN из списка разрешённых:

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk allowed
vlan remove 160
```

По завершении настройки сетей VLAN на коммутаторе конфигурацию можно проверить с помощью команд **show vlan id** *vlan-id***, show interfaces, show vlan brief.**

Команда **show vlan id** *vlan-id* отображает подробные сведения об указанной сети VLAN:

```
Switch#show vlan id 10

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
10   VLAN0010                         active

VLAN Type SAID       MTU  Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ---- -------- ----- ------ ------ -------- ---- -------- ------ ------
10   enet 100010  1500 -      -      -        -    -        0      0
```

Используя команду **show vlan brief**, можно отобразить содержимое файла vlan.dat.

```
Switch#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   VLAN0010                         active
20   VLAN0020                         active
30   VLAN0030                         active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

Протокол VTP. VTP (VLAN Trunking Protocol) – проприетарный протокол компании Cisco для распространения базы данных сетей VLAN на несколько коммутаторов. Работает только через транковые порты. Это позволяет уменьшить работу администраторам по настройке базы сетей VLAN при большом количестве коммутаторов. VTP накладывает свои ограничения: протокол VTP версии 1 и 2 поддержи-

вает только стандартный диапазон VLAN (с 1 по 1005), поддержка расширенного диапазона (с 1006 по 4094) возможна только в версии протокола 3. Поддержка протокола VTP 3 версии начинается с Cisco IOS версии 12.2(52)SE и выше.

Перед созданием на коммутаторе VLAN-сети необходимо сначала создать домен управления протокола VTP, в котором можно протестировать созданную VLAN-сетъ. Все коммутаторы в одном и том же домене управления, которые совместно используют информацию о VLAN-сетях. Коммутатор может присутствовать только в одном домене управления протокола VTP. Находящиеся в разных доменах коммутаторы не могут совместно использовать информацию протокола VTP.

В протоколе VTP каждый коммутатор семейства Catalyst передаёт со своих магистральных портов следующую информацию:

– домен управления;

– номер версии конфигурации;

– известные VLAN-сети и их конкретные параметры.

Значение по умолчанию версии конфигурации – 0.

Домен протокола VTP состоит из одного или более соединённых между собой устройств, совместно использующих доменное имя протокола VTP. Коммутатор может принадлежать только одному домену протокола VTP. При передаче сообщений протокола VTP другим коммутаторам сети происходит инкапсуляция этих сообщений во фреймы магистрального протокола, такого как ISL или IEEE 802.1Q.

Протокол VTP может работать в одном из трёх режимов:

1) режим сервера;

2) режим клиента;

3) прозрачный режим.

Режим работы протокола VTP по умолчанию – серверный. Если коммутатор сконфигурирован в режиме сервера, то можно создавать, изменять или удалять VLAN-сети и другие параметры конфигурации для всего VTP-домена. Серверы VTP сохраняют информацию конфигурации VLAN в NVRAM памяти. VTP-серверы рассылают сообщения протокола VTP со всех своих магистральных портов. Они анонсируют конфигурацию своих VLAN-сетей всем коммутаторам своего VTP-домена и согласовывают конфигурацию своих VLAN-сетей с другими коммутаторами на базе анонсов, полученных от них по магистральным каналам.

Коммутатор, который сконфигурирован как клиент протокола VTP, не может создавать сети VLAN, изменять их или удалять. Кроме того, коммутатор-клиент не может сохранять информацию о VLAN-сетях. Этот режим целесообразно использовать для коммутаторов, которые не имеют достаточной памяти для хранения больших таблиц VLAN-сетей, что требуется для серверов VTP. Клиенты VTP обрабатывают изменения в сетях VLAN, как это делают серверы, и рассылают сообщения протокола VTP со всех своих магистральных портов.

Коммутаторы, сконфигурированные в прозрачном режиме, не принимают участия в работе протокола VTP. Коммутатор, работающий в прозрачном режиме, не анонсирует конфигурацию своей базы данных сетей VLAN. Такие коммутаторы рассылают сообщения, анонсируемые другими устройствами протокола VTP (версия 2), но не анализируют содержащуюся в этих сообщениях информацию. В прозрачном режиме коммутатор не изменяет свою базу данных при получении сообщений об изменении топологии и не рассылает сообщений об изменении топологии своей собственной VLAN-сети.

Установка режима работы:

```
Switch(config)# vtp mode
{client | server | transparent}
```

По умолчанию имя VTP домена не указано. Соответственно, рассылка VTP информации не производится. Для назначения имени домена вводится команда

```
Switch(config)# vtp domain name
```

После ввода этой команды коммутатор в режиме работы сервера, на котором производилось назначение имени, осуществляет рассылку. Все коммутаторы, работающие в режиме сервера или клиента и имеющие нулевое имя домена, принимают сообщение и изменяют имя домена. При каждом изменении базы данных сетей VLAN на сервере, номер версии конфигурации увеличивается. Каждое VTP устройство при приёме сообщения с номером версии конфигурации большим, чем у себя, устанавливает больший номер и изменяет текущую конфигурацию.

Если же домен уже существует, то перед добавлением нового коммутатора следует проверить его имя. Если в домене управления введены меры безопасности, то в нём следует задать пароль:

```
Switch(config)# vtp password password
```

При добавлении клиента протокола VTP к существующему VTP-домену обязательно требуется проверить, что его номер версии конфигурации меньше, чем аналогичные номера у других коммутаторов данного VTP-домена. Для этого используется команда **show vtp status**. Если номер версии конфигурации на добавляемом коммутаторе окажется большим, чем номер версии в домене, то будет стерта VLAN-информация сервера VTP и домена VTP.

```
Switch#show vtp status
VTP Version                     : 2
Configuration Revision          : 3
Maximum VLANs supported locally : 255
Number of existing VLANs        : 8
VTP Operating Mode              : Server
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x9A 0xE7 0x75 0xDF 0x7D 0xDE 0x04 0xA3
Configuration last modified by 0.0.0.0 at 3-1-93 00:00:46
Local updater ID is 0.0.0.0 (no valid interface found)
```

VTP Pruning. По умолчанию коммутатор распространяет по сети широковещательные сообщения и неизвестные пакеты. Такое поведение приводит к передаче по сети данных, которые не являются необходимыми. Отсечение протокола VTP (VTP Pruning) повышает эффективность использования полосы пропускания путём сокращения передачи данных, которые не являются необходимыми, таких, как широковещательные данные, данные многоадресной рассылки, неизвестные пакеты и одноадресные пакеты, рассылаемые методом лавинной рассылки. Отсечение протокола VTP увеличивает доступную полосу пропускания путём ограничения передачи данных на магистральные каналы, по которым должны передаваться данные соответствующим сетевым устройствам. По умолчанию отсечения протокола VTP отключено. Если на удалённом коммутаторе некоторой сети VLAN нет доступного устройства, то отсечение протокола VTP предотвращает рассылку этим коммутатором данных.

Отсечение VTP начинает работать через несколько секунд после его включения. По умолчанию отсечение может быть выполнено для VLAN-сетей с номерами 2-1000. В частности, отсечение потоков данных для сети VLAN 1 вообще невозможно. Для того чтобы сделать допустимым отсечение VLAN-сетей на коммутаторе, необходимо выполнить следующую команду:

```
Switch(config)# vtp pruning
```

VLAN-маршрутизация. VLAN выполняет разбиение сети и разделение трафика на втором уровне. Связь между VLAN невозможна без устройства третьего уровня, например маршрутизатора.

Метод «Router-on-a-Stick» – это такой тип конфигурации маршрутизатора, при котором один физический интерфейс маршрутизирует трафик между несколькими VLAN.

Интерфейс маршрутизатора настраивается для работы в качестве транкового канала и подключается к порту коммутатора, который настроен в режиме транка. Маршрутизатор выполняет маршрутизацию между VLAN, принимая на транковом интерфейсе трафик с меткой VLAN, поступающий от смежного коммутатора, и затем с помощью подынтерфейсов маршрутизируя его между VLAN. Затем уже смаршрутизированный трафик посылается с этого же физического интерфейса с меткой VLAN, соответствующей VLAN назначения.

Подынтерфейсы – это программные виртуальные интерфейсы, связанные с одним физическим интерфейсом. Подынтерфейсы настраиваются в программном обеспечении маршрутизатора, и каждому подынтерфейсу назначаются IP-адрес и номер сети VLAN.

Каждый подынтерфейс создаётся с помощью команды режима глобальной конфигурации **interface** *type number.subint*. Сначала указывается физический интерфейс, затем точка и номер подынтерфейса:

```
Router(config)# interface fastethernet 0/0.200
```

Номер подынтерфейса может быть любым, но чаще всего для удобства администрирования он соответствует номеру сети VLAN.

Перед назначением подынтерфейсу IP-адреса подынтерфейс необходимо настроить для работы в конкретной сети VLAN с помощью команды **encapsulation dot1q** *vlan-id:*

```
Router(config)# interface fastethernet 0/0.200
Router(config-if)# encapsulation dot1q 200
```

Подынтерфейс нетегированной сети VLAN, т.е. native VLAN настраивается с помощью команды **encapsulation dot1q** *vlan_id* **native**. Номер сети VLAN, назначенный подынтерфейсу нетегированной сети VLAN должен совпадать с номером нетегированной сети VLAN коммутатора, к которому он подключён.

Пример:

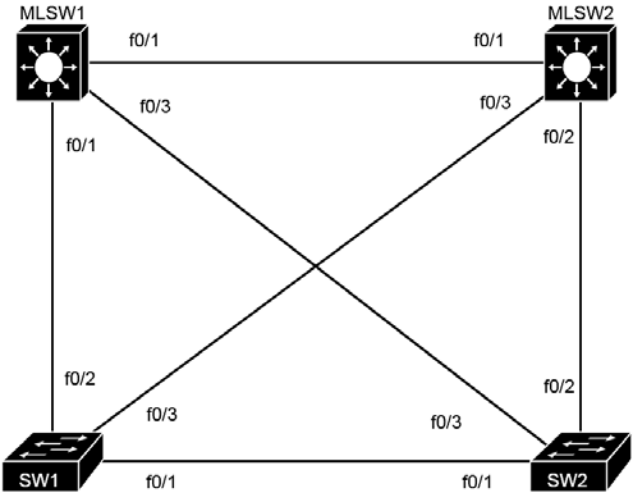
```
Router(config)# interface fastethernet 0/0.100
Router(config-if)# encapsulation dot1q 100
native
```

Контрольные вопросы

1. Принципы настройки и проверки VLAN.
2. Принципы настройки и проверки протокола VTP.
3. Принципы поиска и устранения ошибок конфигурации VLAN и протокола VTP.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить коммутаторы MLSW1, MLSW2, определив интерфейсы FastEthernet 0/1, 0/2 и 0/3 в VLAN 10, 20, 30 соответственно.
3. Настроить коммутаторы SW1, SW2, определив интерфейсы FastEthernet 0/1, 0/2 и 0/3 в VLAN 10, 20, 30 соответственно.

Практическое занятие 6

ПРОТОКОЛЫ СЕМЕЙСТВА STP

Цель: изучение принципов работы протоколов семейства STP.

В результате выполнения практического занятия обучаемые *должны*:

– *знать* принципы функционирования протоколов семейства STP, настройки и проверки протоколов PVST+, PVRST+, поиска и устранения ошибок конфигурации протоколов PVST+, PVRST+;

– *уметь* настраивать и устранять неполадки в работе протоколов семейства STP.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [3, с. 85 –118].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Прокол Spanning Tree Protocol (STP) устраняет проблему петель за счёт управления физическими маршрутами. STP обеспечивает резервирование физических маршрутов и предотвращает нежелательные эффекты, связанные с образованием активных петель в сети. STP – это стандарт комитета IEEE, определённый как 802.1d.

STP работает следующим образом. STP переводит отдельные порты в состояние резервное, в котором они не могут прослушивать, пересылать или выполнять лавинную рассылку кадров. В результате к каждому сегменту ведёт только один постоянно активный маршрут.

Если в подключении к любому из сегментов сети возникает проблема, STP восстанавливает подключение путём автоматической активации неактивного маршрута (если он существует).

В официальном описании работы протокола STP для названия элементов сети используется термин *мост*. Под мостом мы будет по-

нимать любой коммутатор, поддерживающий протокол STP. *Корневой мост* – один из STP-совместимых коммутаторов, с которого начинается построение дерева.

Для создания логической топологии сети с защитой от образования петель STP выполняет следующие действия:

1. *Выбор одного корневого моста*. STP выполняет процесс выбора корневого моста. Только один мост может служить корневым мостом в сети. Все порты корневого моста являются выделенными. Как правило, выделенные порты находятся в режиме пересылки. Порт в режиме пересылки может отправлять и принимать трафик.

2. *Выбор корневого порта на некорневом мосту.* STP выбирает один корневой порт (Root port) на каждом некорневом мосту. Корневой порт представляет маршрут с наименьшей стоимостью от некорневого моста к корневому. Стоимость маршрута протокола Spanning Tree и совокупная стоимость рассчитываются на основе полосы пропускания.

3. *Выбор выделенного* (*назначенного*) *порта для каждого сегмента.* STP выбирает выделенный (назначенный) порт (Designated) в каждом сегменте. Выделенный порт выбирается на мосту, который предоставляет маршрут с наименьшей стоимостью к корневому мосту.

Пример топологии с указанными ролями мостов и портов представлен на рис. 12.

Коммутаторы под управлением алгоритма STP обмениваются сообщениями конфигурации с другими коммутаторами через регулярные интервалы (по умолчанию через каждые две секунды).

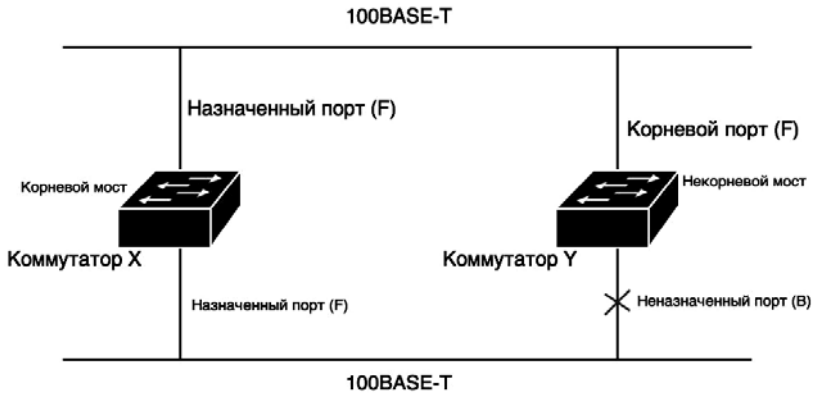


Рис. 12. Роли мостов и портов в протоколе STP

Коммутаторы обмениваются этими сообщениями с помощью многоадресных кадров, отправляемых на мультикастовый Ethernet-адрес 01-80-c2-00-00-00. Такие сообщения называются *блоками данных протокола* (*BPDU*). Один из информационных элементов, входящих в BPDU, – *идентификатор моста* (*BID*).

Формат BPDU-кадра:

Идентификатор протокола (2 байта)	Версия (1 байт)	Тип сообщения (1 байт)	Флаги (1 байт)	Идентификатор корневого моста (8 байт)	Стоимость корневого пути (4 байта)
Идентификатор моста (8 байт)	Идентификатор порта (2 байта)	Возраст сообщения (2 байта)	Максимальный возраст (2 байта)	Время приветствия (2 байта)	Задержки при пересылке (2 байта)

STP для каждого коммутатора назначает уникальный BID. Как правило, BID состоит из значения приоритета (2 байта) и MAC-адреса моста (6 байт). В соответствии со стандартом IEEE 802.1d приоритет по умолчанию – 32 768 (1000 0000 0000 0000 в двоичном исчислении или 0x8000 в шестнадцатеричном). Это среднее значение. Корневой мост – это мост с наименьшим BID.

В STP определено пять режимов работы порта:

1) режим *блокировки* (blocking);
2) режим *прослушивания* (listening);
3) режим *обучения* (learning);
4) режим *пересылки* (forwarding);
5) *отключен* (disabled).

Когда протокол STP активирован, во время загрузки все мосты в сети проходят через блокирующий режим, а затем переходные режимы прослушивания и обучения. Если сеть настроена верно, все порты стабилизируются в режиме пересылки или блокирующем режиме. Порты в режиме пересылки обеспечивают маршрут с наименьшей стоимостью к корневому мосту. При изменении топологии порт временно переходит в режимы прослушивания и обучения.

Все порты моста начинают работу в блокирующем режиме, в котором они ожидают получения блока BPDU. При первой загрузке мост функционирует как корневой и переходит в режим прослушивания. Отсутствие блока BPDU в течение определённого периода времени называется максимальным возрастом (max_age), значение по умолчанию – 20 секунд. Если порт в блокирующем режиме не получает нового блока BPDU в течение периода max_age, мост переходит из блокирующего режима в режим прослушивания. Когда порт находится

в переходном режиме прослушивания, он может отправлять и принимать блоки BPDU для определения активной топологии.

В этот момент через мост не проходят пользовательские данные. В режиме прослушивания мост выполняет три действия:

1) выбор корневого моста;

2) выбор корневых портов на некорневых мостах;

3) выбор выделенных портов для всех сегментов.

Время, которое уходит на переход порта из режима прослушивания в режим обучения и из режима обучения в режим пересылки называется задержкой пересылки. Значение задержки по умолчанию – 15 секунд.

Режим обучения уменьшает объем рассылки, необходимой при запуске пересылки. Если порт остаётся выделенным или корневым по окончании режима обучения, он переходит в режим пересылки. В режиме пересылки порт может отправлять и передавать пользовательские данные. Порты, которые не являются выделенными или корневыми возвращаются в блокирующий режим.

Как правило, порт переходит из блокирующего режима в режим пересылки за 30 – 50 секунд. Вы можете настроить таймеры протокола STP, чтобы изменить эти периоды времени, однако эти таймеры рассчитаны для работы со значениями по умолчанию. Значение по умолчанию дают сети достаточно времени для сбора всех необходимых данных.

Для портов коммутаторов, подключённых только к конечным пользовательским станциям (а не к другому мосту), необходимо включить функцию коммутаторов Cisco Catalyst под названием *PortFast*. Порт коммутатора, на котором включена функция PortFast, автоматически переходит с блокирующего режима в режим пересылки при первоначальном старте. Это приемлемо, так как порт, к которому не подключены другие коммутаторы, не может создавать петли.

Если интерфейс, настроенный с функцией PortFast, получает блок BPDU, протокол STP может перевести порт в блокирующее состояние с помощью функции под названием *BPDU guard.*

Ещё одна функция *BPDU Filtering* – после включения функции, порт не принимает и не отправляет BPDU. Может быть включена глобально на коммутаторе или на интерфейсе.

Настройка функций PortFast на интерфейсе:

```
Switch(config-if)# spanning-tree portfast
```

Включение функции PortFast на всех нетранковых интерфейсах:

```
Switch(config)# spanning-tree portfast default
```

Отключение функции Port Fast на интерфейсе:

```
Switch(config-if)# spanning-tree portfast disable
```

Просмотр информации о статусе функции Port Fast на интерфейсе:

```
Switch# show spanning-tree interface fa 0/1
portfast
VLAN0001 enabled
```

Просмотр информации о настройках STP на интерфейсе:

```
Switch# show spanning-tree interface fa 0/1 detail
Port 1 (FastEthernet0/1) of VLAN0001 is designated
forwarding
  Port path cost 19, Port priority 128, Port Iden-
tifier 128.1.
  Designated root has priority 32769, address
000a.b8ab.eb80
  Designated bridge has priority 32769, address
0012.0111.e580
  Designated port id is 128.1, designated path cost
19
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast mode
  Link type is point-to-point by default
  BPDU: sent 75684, received 0
```

Включение функции BPDU Guard глобально на коммутаторе, на портах с включённой функцией Port Fast:

```
Switch(config)# spanning-tree portfast bpduguard
default
```

Настройка функции BPDU Guard на интерфейсе:

```
Switch(config-if)# spanning-tree bpduguard enable
```

Включение BPDU Filtering глобально на коммутаторе, на портах с включённой функцией Port Fast:

```
Switch(config)# spanning-tree portfast bpdufilter
default
```

Настройка BPDU Filtering на интерфейсе:

```
Switch(config-if)# spanning-tree bpdufilter enable
```

Одна из проблем с STP в том, что само оборудование, которое его использует, может быть причиной сбоя и создания петли. Для предотвращения подобных сбоев была создана функция Loop Guard. Loop Guard обеспечивает дополнительную защиту на втором уровне от возникновения петель. STP- петля возникает, когда блокированный порт в избыточной топологии ошибочно переводится в состояние forwarding. Такая ситуация может возникнуть, например, когда блокированный порт перестаёт получать BPDU. Работа протокола STP основана на постоянном присутствии BPDU-пакетов в сети – назначенный порт постоянно должен передавать BPDU пакеты, а non-designated должен их получать. Как только на порт перестают поступать BPDU, STP понимает это как изменение топологии, т.е. исчезновение петли и переводит порт в состояние forwarding. В случае использования Loop Guard порт после прекращения получения пакетов BPDU переводится в состояние loop-inconsistent и остаётся по-прежнему блокированным. Как только на порт снова начинают поступать BPDU, порт переводится в состояние согласно содержанию пакетов BPDU.

Loop guard должен быть включён на non-designated портах (более точно root и alternate портах).

Включение функции Loop guard глобально:

```
Switch(config)# spanning-tree loopguard default
```

Настройка Loop guard на интерфейсе:

```
Router(config-if)# spanning-tree guard loop
```

Стоимость маршрута протокола STP – это совокупная стоимость маршрута, которая определяется полосой пропускания всех каналов, участвующих в этом маршруте. В новой версии используется нелинейная шкала, позволяющая учитывать высокопроизводительные интерфейсы.

Скорость	Стоимость по актуальной спецификации	Стоимость по старой спецификации
10 Гбит/с	2	1
1 Гбит/с	4	1
100 Мбит/с	19	10
10 Мбит/с	100	100

Протокол Per VLAN Spanning Tree+ (PVST+). Стандарт 802.1в определяет протокол Common Spanning Tree (CST), который разрешает только один экземпляр протокола STP на коммутируемую сеть, независимо от количества VLAN. В сети под управлением CST справедливы следующие утверждения:

1) выравнивание нагрузки невозможно, один восходящий канал должен блокировать все VLAN;

2) потребление ресурсов ЦП невелико, рассчитывается только один экземпляр протокола STP.

Стандарт PVST+ определяет протокол STP, который поддерживает работу нескольких экземпляров протокола в сети, по одному на каждую VLAN. В сети с несколькими экземплярами протокола STP справедливы следующие утверждения:

1) можно обеспечить оптимальное выравнивание нагрузки;

2) поддержка одного экземпляра протокола STP на каждую VLAN может привести к значительному потреблению ресурсов ЦП для всех коммутаторов в сети (в дополнение к потреблению полосы пропускания в связи с тем, что каждый экземпляр отправляет свои блоки BPDU).

Рисунок 13 иллюстрирует топологию с настроенным механизмом распределения нагрузки.

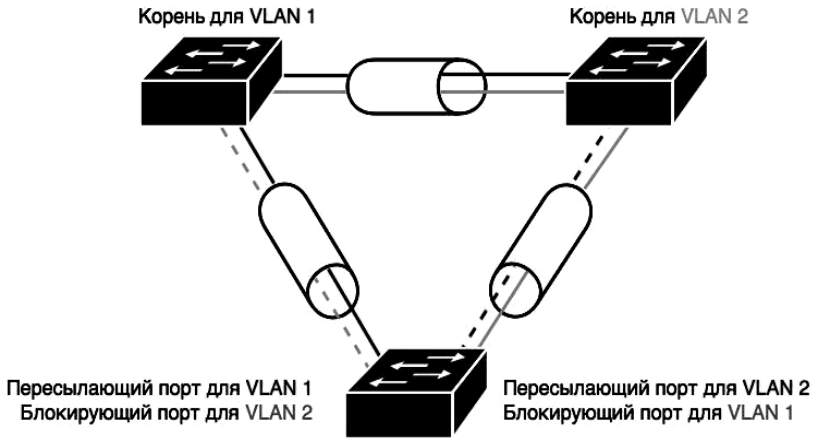


Рис. 13. Распределение нагрузки в протоколе PVST+

В среде Cisco PVST+ можно настроить параметры протокола Spanning Tree так, чтобы по половине сетей VLAN выполняли пересылку на своем транковом канале. Для этого необходимо настроить один из коммутаторов так, чтобы он был выбран корневым в половине VLAN. Другой коммутатор должен быть выбран в качестве корневого коммутатора для другой половины VLAN. Разные корневые коммутаторы STP для разных VLAN позволяют добиться более эффективного резервирования.

Для функционирования протокола каждый коммутатор должен иметь уникальное значение BID. Поскольку протокол PVST+ требует отдельного экземпляра протокола Spanning Tree для каждой VLAN, поле BID должно включать данные об идентификаторе VLAN (VID). Для этого часть поля приоритета используется для передачи расширенного идентификатора системы, содержащего VID. Структура расширенного идентификатора BID представлена на рис. 14. Для поддержки расширенного идентификатора системы оригинальное 16-битное поле приоритета моста 802.1d разделяется на два. Изменённый BID состоит из следующих компонентов:

Приоритет моста – 4-битное поле все ещё используется для данных о приоритете моста. Из-за ограниченного числа битов приоритет передаётся дискретными значениями с шагом 4096, а не с шагом 1, который использовался бы в полном 16-битном поле. Приоритет по умолчанию, согласно стандарту IEEE 802.1D – 32 768 (среднее значение).

Расширенный идентификатор системы – 12-битное поле, используемое для передачи VID для PVST+ (в данном случае).

MAC-адрес – 6-байтное поле с MAC-адресом конкретного коммутатора.

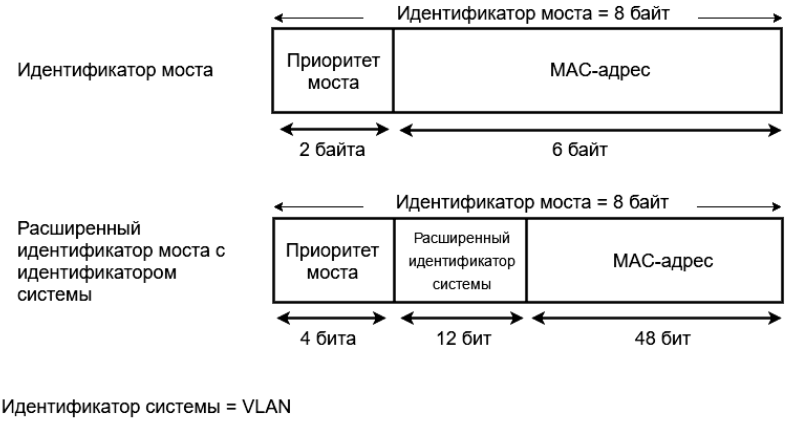


Рис. 14. Структура идентификатора моста

Если приоритет не задан, коммутаторы будут использовать одинаковое значение приоритета по умолчанию и выбор корневого моста для каждой VLAN будет выполняться на основе MAC-адреса. Этот метод выбора корневого моста является случайным, поэтому рекомендуется назначить более низкий приоритет коммутатору, который должен служить корневым мостом.

RSTP. Протокол *Rapid Spanning Tree Protocol* (*RSTP*), определённый стандартом IEEE 802.1w, заменяет протокол STP, определённый в стандарте 802.1d, но остаётся совместимым с STP. RSTP значительно уменьшает время повторного сведения активной топологии при изменении физической топологии или параметров её конфигурации. RSTP определяет дополнительные роли портов (альтернативный и резервный) и использует следующие режимы портов: режим отбрасывания, режим обучения и режим пересылки.

RSTP выбирает один коммутатор корневым мостом активной топологии протокола Spanning Tree и назначает роли отдельным портам коммутатора, в зависимости от того, входят ли они в активную топологию.

RSTP обеспечивает быстрое восстановление подключения после отказа коммутатора, порта коммутатора или сегмента сети. Новые корневой порт и выделенный порт на другой стороне моста переходят в режим пересылки в результате развёрнутого процесса согласования подключения. RSTP разрешает настройку портов коммутатора, что позволяет переводить порты в режим пересылки сразу после повторной инициализации коммутатора.

RSTP определяет следующие роли портов (рис. 15).

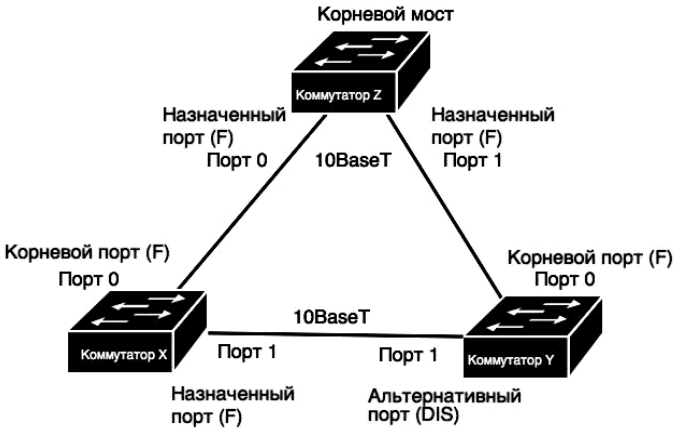


Рис. 15. Роли мостов и портов в протоколе RSTP

Корневой (*Root*) – порт в режиме пересылки, выбранный для топологии Spanning Tree.

Выделенный (*Designated*) – порт в режиме пересылки, выбирается для каждого сегмента коммутируемой сети.

Альтернативный (*Alternate*) – альтернативный маршрут к корневому мосту, отличный от маршрута корневого порта.

Резервный (*Backup*) – маршрут, который предлагает резервное (но менее предпочтительное) подключение к сегменту, к которому уже подключён другой порт коммутатора. Резервные порты могут существовать только если два порта соединены петлёй с помощью канала типа «точка-точка» или если мост имеет два или более подключений к общему сегменту сети.

Роли «Корневой» и «Выделенный» включают порт в активную топологию. Роли «Альтернативный» и «Резервный» исключают порт из активной топологии.

Режим порта контролирует процессы пересылки и обучения и предоставляет значения параметров для отбрасывания, обучения и пересылки. В таблице ниже приводится сравнение режимов портов для STP и RSTP:

Режим порта STP	Режим порта RSTP	Порт входит в активную топологию
Блокирующий	Отбрасывания	Нет
Прослушивания	Отбрасывания	Нет
Обучения	Обучения	Да
Пересылки	Пересылки	Да

В стабильной топологии RSTP обеспечивает переход всех корневых и выделенных портов в режим пересылки, в то время как альтернативные и резервные порты находятся в режиме отбрасывания.

В протоколе STP для того, чтобы убедиться, что порт может участвовать в передаче данных, требовались таймеры, т.е. мост пассивно ждал в течение означенного времени, слушая BPDU. Ключевой особенностью RSTP стало введение концепции типов портов, основанных на режиме работы соединения – полный дуплекс или полудуплекс (типы портов *p2p* или *shared*, соответственно), а также понятия пограничный порт (тип *edge p2p*) для конечных устройств.

Пограничные порты назначаются командой spanning-tree portfast: при включении провода порт сразу переходит к forwarding-состоянию.

Shared-порты работают по старой схеме с прохождением через состояния BLK – LIS – LRN – FWD. А вот на p2p-портах RSTP использует процесс предложения и соглашения (*proposal and agreement*). Вместо ожидания входящих BPDU мост пытается связаться с удалённым коммутатором на том конце провода с помощью специальных proposal BPDU, в которых есть информация о стоимости маршрута к корневому мосту. Удалённый мост сравнивает полученную информацию с текущей и принимает решение, о чём извещает первый мост посредством agreement BPDU. Так как весь этот процесс не привязан к таймерам, происходит он очень быстро.

Изменить тип соединения порта можно следующей командой:

```
Switch(config-if)# spanning-tree link-type
{point-to-point | shared}
```

Протокол PVRST+. Стандарт RSTP (802.1w) использует общий протокол Spanning Tree, что подразумевает использование одного экземпляра протокола Spanning Tree на сеть, независимо от количества VLAN. Стандарт Per VLAN Rapid Spanning Tree Plus (PVRST+) определяет протокол Spanning Tree, который позволяет использовать отдельный экземпляр RSTP для каждой VLAN.

Протокол MSTP. Протокол Multiple Spanning Tree Protocol (MSTP) изначально был определён в стандарте IEEE 802.1s, а затем добавлен в стандарт IEEE 802.1q 2003. Он представляет собой протокол Spanning Tree с несколькими экземплярами Spanning Tree на сеть. Но в отличие от протокола PVRST+, который подразумевает использование одного RSTP на VLAN, MSTP уменьшает нагрузку на коммутаторы благодаря использованию одного экземпляра протокола Spanning Tree для нескольких VLAN.

Коммутаторы Cisco поддерживают три типа протокола Spanning Tree:

PVST+;

PVRST+;

MSTP.

Инструкции по настройке PVRST+.

1. Включите PVRST+.
2. Выделите и настройте коммутатор в качестве корневого моста.
3. Выделите и настройте коммутатор в качестве вспомогательного корневого моста.
4. Проверьте конфигурацию.

Включение протокола PVRST+:

```
Switch(config)# spanning-tree mode rapid-pvst
```

Проверка конфигурации:

```
Switch# show spanning-tree vlan number
```

Проверка конфигурации STP:

```
Switch# show spanning-tree vlan number
```

Пример вывода команды:

```
Switch#show spanning-tree vlan 100

VLAN0100
  Spanning tree enabled protocol ieee
  Root ID Priority 32868
  Address aabb.cc00.0100
  Cost 100
  Port 1 (Ethernet0/0)
  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32868 (priority 32768 sys-id-ext 100)
  Address aabb.cc00.0300
  Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
  Aging Time 300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------
Et0/0               Root FWD 100       128.1    Shr
Et0/1               Altn BLK 100       128.2    Shr
```

VLAN0100 – номер VLAN, в котором работает процесс STP. Тип протокола – ieee. В случае использования PVRST+ будет указан протокол rstp, для MSTP – mstp. Текущий мост не является корневым. Приоритет 32868 складывается из стандартного приоритета 32868 и номера VLAN 100. MAC-адрес коммутатора aabb.cc00.0100. Стоимость пути до корневого моста – 100. В нижней части расположена сводная таблица состояния портов, которая состоит из следующих колонок (слева направо):

1) порт;
2) его роль (Root – корневой порт, Desg – выделенный порт, Altn – альтернативный, Back – резервный);
3) его статус (FWD – пересылка, BLK – блокирующий, LIS – прослушивание, LRN – обучение);
4) стоимость маршрута до корневого моста;
5) приоритет порта в формате «приоритет порта.номер порта»;
6) тип соединения.

Возможны случаи, когда стоимость пути по двум и более портам коммутатора будет одинакова, тогда выбор корневого порта будет происходить на основании приоритета и порядкового номера порта, например, из портов fa0/1, fa0/2, fa0/3 корневым станет порт с наименьшим номером.

Приоритет порта может иметь значения от 0 до 192. По умолчанию – 128. Изменить стандартное значение приоритета порта можно командой

```
Switch(config-if)#spanning-tree port-priority number
```

Включение режима отладки протокола PVRST+:

```
Switch# debug spanning-tree pvst+
```

Если все коммутаторы в сети настроены с параметрами протокола Spanning Tree по умолчанию, коммутатор с наименьшим MAC-адресом становится корневым мостом. Однако корневой мост по умолчанию может не быть идеальным корневым мостом в соответствие с моделью трафика, количеством интерфейсов в режиме пересылки или типами каналов.

Перед настройкой STP выберите коммутатор, который будет корневым мостом протокола Spanning Tree. Этот коммутатор не обязательно должен являться самым производительным коммутатором, но он должен быть самым центральным коммутатором в сети. Все потоки данных в сети будут проходить через этот коммутатор.

Повышая приоритет (уменьшая численное значение) предпочтительного коммутатора, чтобы сделать его корневым мостом, вы заставляете протокол Spanning Tree выполнить повторный расчёт в соответствии с новой топологией, в которой предпочтительный коммутатор является корневым мостом.

Эта команда делает коммутатор корневым мостом для VLAN 1:

```
Switch(config)# spanning-tree vlan 1 root primary
```

При вводе этой команды коммутатор проверяет приоритет корневого моста для указанной VLAN. Из-за расширенного идентификатора системы на коммутаторе задаётся значение приоритета 24 576 для указанной VLAN, если это значение установит коммутатор корневым мостом текущей VLAN. Если в указанной VLAN есть другой коммутатор с приоритетом ниже 24 576, коммутатор, на котором вы настраиваете команду root primary, устанавливает свой приоритет для указанной VLAN на 4096 ниже самого низкого значения приоритета.

Вспомогательный корневой мост – это коммутатор, который становится корневым мостом VLAN при отказе основного корневого моста. Эта команда настраивает коммутатор в качестве вспомогательного корневого моста для VLAN 2:

```
Switch(config)# spanning-tree vlan 2 root secondary
```

При вводе этой команды приоритет коммутатора меняется со значения по умолчанию 32 768 на 28 672. Если другие мосты VLAN сохранят приоритет STP по умолчанию, этот коммутатор становится корневым мостом при отказе основного корневого моста. Эту команду можно выполнить на нескольких коммутаторах, чтобы настроить несколько резервных корневых мостов.

Изменение приоритета моста для конкретного номера VLAN:

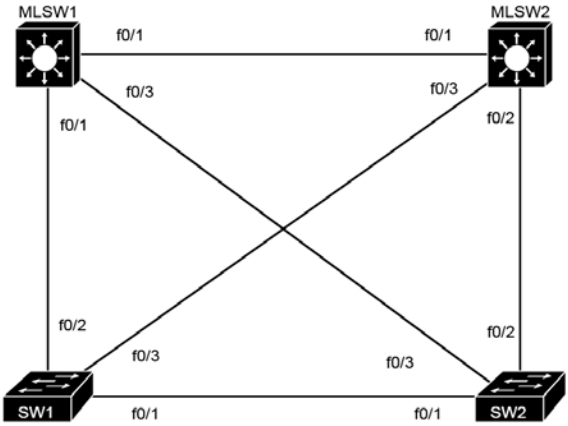
```
Switch(config)# spanning-tree vlan vlan-id priority number
```

Контрольные вопросы

1. Принципы функционирования протоколов семейства STP.
2. Принципы настройки и проверки протоколов PVST+, PVRST+.
3. Принципы поиска и устранения ошибок конфигурации протоколов PVST+, PVRST+.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить протокол VTP:
– VTP-домен – cisco;
– MLSW1, MLSW2 – VTP-серверы;
– SW1, SW2 – VTP-клиенты;
– создать VLAN 10, 20, 30, 40.
3. Настроить протокол STP:
– режим PVRST+;
– корень для VLAN 10, 20 – MLSW1;
– корень для VLAN 30, 40 – MLSW2.

Практическое занятие 7

СПИСКИ КОНТРОЛЯ ДОСТУПА

Цель: изучение принципов использования списков контроля доступа для обеспечения безопасности сети.

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: принципы настройки и проверки стандартных, расширенных, нумерованных и именованных списков контроля доступа; принципы поиска и устранения ошибок конфигурации списков контроля доступа.

– *уметь* настраивать и проверять списки контроля доступа.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.
2. Основная часть – устный или письменный опрос, решение предложенных задач.
3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 638 – 698], [10].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Одним из наиболее распространённых способов фильтрации трафика является использование *списков контроля доступа* (*ACL-списков*). ACL-списки можно использовать для управления входящим и существующим трафиком в сети и его фильтрации.

В списках контроля доступа содержится одна или более инструкций. Каждая инструкция даёт либо разрешение, либо запрещает трафик на основе указанных параметров. Трафик сравнивается с каждой инструкцией в ACL-списке по порядку, пока не будет найдено совпадение или не закончится список инструкций.

Последняя инструкция в ACL-списке всегда содержит *неявный запрет трафика*. Эта инструкция автоматически вставляется в конец каждого ACL-списка, хотя и не присутствует в нем физически. Неяв-

ный запрет блокирует весь трафик. Эта возможность позволяет предотвратить случайное попадание нежелательного трафика.

Стандартным спискам контроля доступа для IPv4 назначаются номера из диапазонов *1 – 99* и *1300 – 1999*. Они фильтруют пакеты в зависимости от *адреса источника и маски* и запрещают или разрешают все пакеты стека протоколов TCP/IP. Стандартные списки контроля доступа могут не обеспечивать нужного уровня контроля над фильтрацией. Администратору могут потребоваться более точные методы фильтрации трафика.

Для настройки стандартного списка доступа для IPv4 на маршрутизаторе Cisco необходимо создать стандартный список для IPv4 и активировать его на интерфейсе. Для создания записи в стандартном списке фильтров трафика для IPv4 используется команда **access-list**:

```
Router(config)# access-list access-list-number
{permit | deny | remark} source [source-wildcard]
[log]
```

Параметры команды access-list	Описание
access-list-number	Назначает списку номер из диапазона 1 – 99 1300 – 1999
permit | deny	Разрешает или блокирует указанный адрес
source	Идентифицирует IP-адреса источника: **host** – единственный хост-источник **any** – любой хост-источник
source-wildcard	Шаблонная маска источника
log	Отправляет сообщение журнала на консоль

Команда **ip access-group** привязывает существующий список контроля доступа к интерфейсу. Допускается использование **только одного** списка контроля доступа на протокол, направление и интерфейс:

```
Router(config-if)# ip access-group access-list-number
{in | out}
```

Аргумент **in** означает входящее направление, **out** – исходящее направление. При выборе направления, для которого будет использоваться ACL-список, необходимо представить поток трафика с точки зрения маршрутизатора.

Входящий трафик – это трафик, поступающий в интерфейс маршрутизатора извне. Маршрутизатор сравнивает входящий пакет с ACL-списком перед поиском сети назначения в таблице маршрутизации. Пакеты, отбрасываемые в этой точке, позволяют исключить излишние операции поиска маршрутизатора. Это делает входящий список контроля доступа более эффективным для маршрутизатора, чем исходящий список контроля доступа.

Исходящий трафик проходит через маршрутизатор по интерфейсу. Для исходящего пакета маршрутизатор уже осуществил поиск по таблице маршрутизации и переключил пакет на правильный интерфейс. Пакет сравнивается с ACL-списком непосредственно перед выходом из маршрутизатора.

Чтобы удалить список ACL для протокола IP с интерфейса сначала необходимо ввести команду **no ip access-group** *access-list-number* на этом интерфейсе, а затем глобальную команду **no access-list** *access-list-number*, чтобы полностью удалить список доступа.

Для контроля входящего и исходящего трафика маршрутизатора необходимо защитить виртуальные порты маршрутизатора (**VTY**). Ограничение доступа к VTY повышает безопасность сети и подразумевает определение адресов, для которых разрешён доступ по протоколу Telnet или SSH.

```
Router(config-line)# access-class access-list-number
{in | out}
```

Пример стандартного списка доступа:

```
Router(config)# access-list 1 permit 172.16.0.0
0.0.255.255
(неявное утверждение deny all – не отображается
в списке)
(access-list 1 deny 0.0.0.0 255.255.255.255)
Router(config)# interface fastethernet 0/0
Router(config-if)# ip access-group 1 out
```

Параметры команды access-list	Описание
1	Номер списка ACL, который указывает, что список является стандартным
permit	Разрешает пересылку трафика, соответствующего указанным параметрам
172.16.0.0	IP-адрес, который используется вместе с шаблонной маской для идентификации сети-источника
0.0.255.255	Шаблонная маска, нули соответствуют позициям, которые должны совпадать, единицы – игнорируемые позиции
ip access-group 1 out	Привязывает список контроля доступа к интерфейсу в качестве исходящего фильтра

Этот список контроля доступа разрешает пересылку от интерфейса FastEhternet 0/0 только трафика сети 172.16.0.0. Трафик из всех остальных сетей блокируется.

```
Router(config)# access-list 1 deny 172.16.4.13
0.0.0.0
Router(config)# access-list 1 permit 0.0.0.0
255.255.255.255
(неявное утверждение deny all)
(access-list 1 deny 0.0.0.0 255.255.255.255)
Router(config)# interface fastethernet 0/0
Router(config-if)# ip access-group 1 out
```

Этот список контроля доступа блокирует трафик с определённого адреса (172.16.4.13) и разрешает пересылку всего остального трафика от интерфейса FastEthernet 0/0. Комбинация IP-адреса и шаблонной маски 0.0.0.0 можно записать с помощью ключевого слова host с последующим указанием адреса. Комбинация IP-адреса и шаблонной маски 0.0.0.0 255.255.255.255 разрешает трафик из любого источника. Эту комбинацию также можно записать с помощью ключевого слова any:

```
Router(config)# access-list 1 deny host 172.16.4.13
Router(config)# access-list 1 permit any
```

Для более точного контроля над фильтрацией трафика используются расширенные списки доступа IPv4 с номерами диапазонов *100 – 199* и *2 000 – 2 699*, которые проверяют IPv4 адреса источника и назначения. Кроме того, в конце утверждения расширенного списка контроля доступа можно указать протокол и приложение TCP или UDP (необязательно) для более точной фильтрации. Чтобы задать приложение, необходимо указать номер порта или имя хорошо известного приложения.

Чтобы настроить расширенный нумерованный список контроля доступа для IPv4 на маршрутизаторе Cisco, создайте этот список и активируйте его на интерфейсе. Используйте команду **access-list**, чтобы создать запись с утверждениями сложного фильтра:

```
Router(config)# access-list access-list-number
{permit | deny} protocol source [source-wildcard]
destination [destination-wildcard]
```

Расширенный ACL при указании протоколов TCP или UDP, позволяет указывать и порты отправителя и/или получателя:

```
Router(config)#access-list access-list-number
{permit | deny | remark} protocol source [source-
wildcard] [operator port] destination [destination-
wildcard] [operator port] [established] [log]
```

Параметры команды access-list	Описание
access-list-number	Назначает списку номер из диапазона 100 – 199 и 2000 – 2699
permit | deny	Разрешает или блокирует трафик
protocol	Указание протокола. Возможные значения: AHP, EIGRP, ESP, GRE, ICMP, IP, OSPF, TCP, UDP
source	Идентифицирует IP-адреса источника: host – единственный хост any – любой хост
source-wildcard	Шаблонная маска источника
destination	Идентифицирует IP-адреса назначения: host – единственный хост any – любой хост
destination-wildcard	Шаблонная маска получателя

Продолжение табл.

Параметры команды access-list	Описание
operator	Вид оператора: eq – только определённый номер порта; gt – только пакеты с большим номером порта; lt – только пакеты с меньшим номером порта; neq – только пакеты с не равным номером; range – диапазон портов
port	Число в диапазоне 0 – 65 535. В качестве альтернативы номеру порта можно использовать широко известные имена приложений, такие как: FTP, POP3, SMTP, Telnet, WWW
established	Только для входящего трафика TCP. Разрешает трафик TCP, если пакет сгенерирован в ответ на сеанс, созданный во внутренней сети. Для этого типа трафика устанавливаются биты подтверждения (ACK)
log	Отправляет сообщение журнала на консоль

Команда **ip access-group** привязывает существующий список контроля доступа к интерфейсу. Допускается использование только одного списка контроля доступа на протокол, направление и интерфейс:

```
Router(config)# access-list 101 deny tcp 172.16.4.0
0.0.0.255 172.16.3.0 0.0.0.255 eq 21
Router(config)# access-list 101 deny tcp 172.16.4.0
0.0.0.255 172.16.3.0 0.0.0.255 eq 20
Router(config)# access-list 101 permit ip any any
(неявное утверждение deny all)
(access-list 101 deny ip 0.0.0.0 255.255.255.255
0.0.0.0 255.255.255.255)
Router(config)# interface fastethernet 0/0
Router(config-if)# ip access-group 101 out
```

Параметры команды access-list	Описание
`101`	Номер списка контроля доступа, указывает на расширенный список
`deny`	Запрещает пересылку трафика, соответствующего заданным параметрам
`tcp`	Протокол TCP
`172.16.4.0 0.0.0.255`	IP-адрес источника и маска, первые три октета должны совпадать, последний – нет
`172.16.3.0 0.0.0.255`	IP-адрес назначения и маска, первые три октета должны совпадать, последний – нет
`eq 21`	Порт назначения, в данном примере широко известный порт управления FTP
`eq 20`	Порт назначения, в данном примере широко известный порт передачи данных FTP

Запрещающие инструкции ACL в примере запрещают FTP-трафик из подсети 172.16.4.0/24 в подсеть 172.16.3.0/24.

Именованные списки контроля доступа позволяют идентифицировать стандартные и расширенные списки контроля доступа для протокола IP с помощью цифро-буквенной строки (имени) вместо номера. Именованные списки контроля доступа поддерживают удаление отдельных записей. Версия Cisco IOS 12.3 и выше позволяет использовать номера последовательности для вставки инструкций в любое место именованного списка контроля доступа. Версии, предшествующие Cisco IOS 12.3, позволяют добавлять инструкции только в конец именованного списка контроля доступа. Поскольку именованные списки позволяют удалять отдельные записи, их можно изменять без необходимости в удалении и повторном создании всего списка. Используйте именованные списки контроля доступа для протокола IP, если необходима интуитивно понятная идентификация.

Процедура создания стандартных именованных списков доступа протокола IP:

```
    Router(config)# ip access-list standard
access-list-name
    Router(config-std-nacl)# [sequence-number]
{deny | permit} source [source-wildcard]
```

Для создания стандартного именованного списка доступа протокола IP выполните действия, описанные в таблице. Первое действие следует выполнить в режиме глобальной конфигурации.

№	Действие	Примечания
1	`ip access-list standard` ***`access-list-name`***	Задает стандартный список контроля доступа и присваивает ему имя

2	`[sequence-number] deny source [source-wildcard]`
	
	`[sequence-number] permit source [source-wildcard]`
		В режиме конфигурации списка контроля доступа укажите одно или несколько разрешающих или запрещающих условий.
		Они определяют, будет пакет пропущен или отброшен

Пример стандартного именованного списка контроля доступа для IPv4:

```
    Router(config)#ip access-list standard trouble-
maker
    Router(config-std-nacl)#deny host 172.16.4.13
    Router(config-std-nacl)#permit 172.16.4.0
0.0.0.255
    Router(config-std-nacl)#interface f0/0
    Router(config-if)#ip access-group troublemaker
out
```

Параметры команды access-list	Описание
`standard`	Определяет именованный список контроля доступа как стандартный
`troublemaker`	Имя списка контроля доступа
`deny`	Запрещает пересылку трафика, соответствующего заданным параметрам
`host 172.16.4.13`	IP-адрес источника, ключевое слов «host» соответствует шаблонной маске 0.0.0.0
`permit`	Разрешает пересылку трафика, соответствующего указанным параметрам
`172.16.4.0 0.0.0.255`	IP-адрес источника и маска, первые три октета должны совпадать, последний – нет

Для создания расширенного именованного списка доступа протокола IP выполните действия, описанные в таблице. Первое действие следует выполнить в режиме глобальной конфигурации.

№	Действие	Примечания
1	**ip access-list extended** *access-list-name*	Задаёт расширенный список контроля доступа и присваивает ему имя
2	Введите один из следующих параметров: [*sequence-number*] {**deny** | **permit**} *protocol source* [*source-wildcard*] *destination* [*destination-wildcard*] [**precedence** *precedence*] [**tos** *tos*] [*sequence-number*] {**deny** | **permit**} *protocol* **any any** [*sequence-number*] {**deny** | **permit**} *protocol* **host** *source* **host** *destination*	В режиме конфигурации списка контроля доступа задайте условия разрешения или запрета: host – единственный хост; any – любой хост

Пример расширенного именованного списка контроля доступа:

```
    Router(config)#ip access-list extended badgroup
    Router(config-ext-nacl)#deny tcp 172.16.4.0
0.0.0.255 any eq 23
    Router(config-ext-nacl)#permit ip any any
    Router(config-ext-nacl)#interface f0/0
    Router(config-if)#ip access-group badgroup out
```

Параметры команды access-list	Параметры команды access-list Описание
`extended`	Определяет именованный список контроля доступа как расширенный
`badgroup`	Имя списка контроля доступа
`deny`	Запрещает пересылку трафика, соответствующего заданным параметрам
`tcp`	Протокол TCP
`172.16.4.0 0.0.0.255`	IP-адрес источника и маска, первые три октета должны совпадать, последний – нет
`any`	Любой IP-адрес назначения
`eq 23 or eq telnet`	Порт назначения или имя приложения. В этом примере указывается широко известный порт для Telnet (23)
`permit`	Разрешает пересылку трафика, соответствующего указанным параметрам
`ip`	Протокол сетевого уровня
`any`	Ключевое слово, соответствующее трафику из любого источника к любому месту назначения

Комментариями или примечаниями называются инструкции списков контроля доступа, которые не обрабатываются. Это простые описательные утверждения, которые помогают лучше понять именованные или нумерованные списки контроля доступа, а также устранять неполадки в них.

Длина примечания ограничена 100 символами. Примечание можно добавлять до или после разрешающей или запрещающей инструкции. Однако при добавлении примечаний следует использовать согласованный подход, чтобы пользователь всегда мог понять, к какой запрещающей или разрешающей инструкции относится примечание. Размещение одной части примечаний до инструкций, а другой – после инструкций может привести к путанице.

Для добавления комментария в именованный список контроля доступа по протоколу IP используется команда remark. Чтобы доба-

вить комментарий в нумерованный список, используйте команду **access list** *access-list-number* **remark** *remark.*

Пример добавления комментария в нумерованный список контроля доступа:

```
    Router(config)# access list 101 remark Permit-
ting_John to Telnet to Server
    Router(config)# access list 101 permit tcp host
10.1.1.2 host 172.16.1.1 eq telnet
```

Пример добавления комментария в именованный список контроля доступа:

```
    Router(config)# ip access list standard PREVEN-
TION remark Do not allow Jones subnet through
    Router(config-std-nacl)# deny 171.69.0.0
0.0.255.255
```

Завершив настройку списка контроля доступа, воспользуйтесь командами show, чтобы проверить конфигурацию. Команда **show access-lists** отображает содержимое всех списков контроля доступа. Добавив имя или номер списка контроля доступа в качестве параметра этой команды, можно вывести определённый список. Чтобы вывести только содержимое списков доступа по протоколу IP, используйте команду **show ip access-list**.

Редактирование списков контроля доступа. Из стандартного или расширенного ACL-списка нельзя удалить одну строку. Вместо этого ACL-список удаляется полностью и его необходимо заменить.

В текущих версиях IOS для редактирования нумерованных и именованных ACL-списков используется команда **ip access-list**. ACL-список выводится со строками с нумерацией 10, 20, 30 и так далее. Для просмотра номеров строк используется команда **show access-lists**.

Чтобы изменить существующую строку, выполните следующие действия:

– удалите строку при помощи команды no с номером строки;

– повторно добавьте эту же строку с её номером.

Используйте команду **show access-lists** для отображения переупорядоченных и перенумерованных строк:

```
Router(config)#ip access-list standard LIST1
Router(config-std-nacl)#permit 10.1.1.0 0.0.0.15
Router(config-std-nacl)#permit 192.168.0.0
0.0.31.255
Router(config-std-nacl)#end
```

```
Router#show ip access-lists
Standard IP access list LIST1
    10 permit 10.1.1.0 0.0.0.15
    20 permit 192.168.0.0 0.0.31.255

Router(config)#ip access-list standard LIST1
Router(config-std-nacl)#25 deny 10.2.2.0 0.0.0.255
Router(config-std-nacl)#no 20
Router(config-std-nacl)#20 permit 192.168.0.0
0.0.0.255

Router#show ip access-lists
Standard IP access list LIST1
    10 permit 10.1.1.0 0.0.0.15
    20 permit 192.168.0.0 0.0.0.255
    25 deny 10.2.2.0 0.0.0.255
```

Произвести перенумерацию строк в списке доступа можно с помощью команды

```
Router(config)# ip access-list resequence
access-list-name starting-sequence-number increment
```

При использовании нумерованных ACL-списков инструкции, вводимые после создания ACL-списка, добавляются в конец. Такой порядок может не дать ожидаемых результатов. Чтобы решить эту проблему, удалите исходный ACL-список и создайте его заново.

Часто рекомендуют создавать ACL-списки в текстовом редакторе. Это позволит легко изменять и вставлять ACL-список в конфигурацию маршрутизатора. Однако следует помнить, что при копировании и вставке ACL-списка важно сначала удалить текущий применённый ACL-список, в противном случае все инструкции будут добавлены в конец.

Правила размещения списков контроля доступа. Стандартные ACL-списки легко создавать и внедрять. Однако фильтрация по стандартным ACL-спискам возможна только на основе исходящего адреса и применяется ко всему трафику без учёта его типа или назначения. При маршрутизации в несколько сетей слишком близкое размещение стандартного ACL-списка к источнику может непреднамеренно блокировать допустимый трафик. Следовательно, важно размещать *стандартные ACL-списки как можно ближе к узлу назначения*.

В случае более сложных требований к фильтрации следует использовать расширенный ACL-список. Расширенные ACL-списки дают больший контроль, чем стандартные. Они допускают фильтрацию по исходным и конечным адресам. Эти списки также обеспечи-

вают фильтрацию по протоколу сетевого уровня, протоколу транспортного уровня и номерам портов, если это необходимо. Такая более точная фильтрация позволяет администратору сети создавать ACL-списки, отвечающие определённым потребностям плана по обеспечению безопасности.

Размещайте *расширенный ACL-список ближе к адресу источника*. Благодаря анализу по исходному и конечному адресу, ACL-список позволяет блокировать пакеты, направляемые в определённую конечную сеть прежде, чем они покинут исходный маршрутизатор. Пакеты фильтруются прежде, чем они пересекут границы сети, что помогает поддерживать пропускную способность.

Дополнительные списки контроля доступа. Динамические списки контроля доступа зависят от возможностей подключения по Telnet, аутентификации (локальной или удалённой) и расширенных списков контроля доступа. Настройка начинается с внедрения расширенного списка контроля доступа для блокировки трафика, проходящего через маршрутизатор.

Пользователи, которые пытаются передать данные через маршрутизатор, блокируются расширенным списком контроля доступа, пока они не подключатся к маршрутизатору по Telnet и не аутентифицируются. Затем Telnet-подключение сбрасывается и к существующему расширенному списку контроля доступа добавляется новый динамический список с одной записью. Он пропускает трафик в течение определённого периода времени, который можно задать в качестве периода бездействия или как абсолютное значение времени ожидания.

Следующая конфигурация создаёт имя пользователя и пароль для аутентификации. Время бездействия установлено на 10 минут:

```
    Router(config)# username test password 0 test
    Router(config)# username test autocommand
access-enable host timeout 10
```

Следующая конфигурация позволяет пользователям открыть сеанс Telnet с маршрутизатором для аутентификации и блокирует весь остальной трафик:

```
    Router(config)# access-list 101 permit tcp any
host 10.1.1.1 eq telnet
    Router(config)# interface Ethernet0/0
    Router(config-if)# ip address 10.1.1.1
255.255.255.0
    Router(config-if)# ip access-group 101 in
```

Следующая конфигурация создаёт динамический список контроля доступа, который будет автоматически применяться к существую-

щему списку access-list 101. Задано абсолютное время ожидания 15 минут:

```
Router(config)# access-list 101 dynamic test-
list timeout 15 permit ip 10.1.1.0 0.0.0.255
172.16.1.0 0.0.0.255
```

Следующая конфигурация включает аутентификацию пользователей, пытающихся открыть сеанс Telnet с маршрутизатором:

```
Router(config)# line vty 0 4
Router(config-line)# login local
```

После создания этой конфигурации, если пользователь сети 10.1.1.0/24 успешно создаёт сеанс Telnet с интерфейсом 10.1.1.1, применяется динамический список доступа. Затем сеанс сбрасывается и пользователь получает доступ к сети 172.16.1.0/24.

Рефлексивные списки контроля доступа обеспечивают фильтрацию IP-пакетов в соответствии с данными сеанса верхнего уровня. Они используются для разрешения исходящего трафика и ограничения входящего трафика в зависимости от сеансов, созданных из внутренней сети маршрутизатора. Рефлексивные списки контроля доступа создают только временные записи. Эти записи автоматически генерируются при запуске нового сеанса IP, например исходящим пакетом. Записи автоматически удаляются в конце сеанса. Рефлексивные списки контроля доступа не применяются напрямую к интерфейсу, но вносятся в расширенный именованный список контроля доступа, который активируется на интерфейсе.

Рефлексивные списки контроля доступа предлагают более «истинную» форму фильтрации сеансов, чем расширенный список контроля доступа с параметром established. Рефлексивные списки контроля доступа гораздо сложнее обмануть, так как перед принятием пакета необходимо обеспечить соответствия большему числу критериев. Проверяются не только биты подтверждения (ACK) и сброса (RST), но и адреса источника и назначения, а также номера портов.

Пример рефлексивного списка контроля доступа разрешает входящий и исходящий трафик ICMP и пропускает только трафик TCP, отправленный изнутри. Весь остальной трафик отклоняется.

Следующая конфигурация заставляет маршрутизатор отслеживать трафик, инициированный изнутри:

```
Router(config)# ip access-list extended out-
boundfilters
Router(config-ext-nacl)# permit icmp 10.1.1.0
0.0.0.255 172.16.1.0 0.0.0.255
```

```
    Router(config-ext-nacl)# permit tcp 10.1.1.0
0.0.0.255 172.16.1.0 0.0.0.255 reflect tcptraffic
```

Следующая конфигурация создаёт политику, которая требует, чтобы маршрутизатор проверял весь входящий трафик, чтобы определить, был ли он инициирован изнутри и привязывает рефлексивную часть списка контроля доступа outboundfilters (которая называется tcptraffic) к списку контроля доступа inboundfilters:

```
    Router(config)# ip access-list extended in-
boundfilters
    Router(config-ext-nacl)# permit icmp 172.16.1.0
0.0.0.255 10.1.1.0 0.0.0.255 evaluate tcptraffic
```

Следующая конфигурация применяет входящий и исходящий список контроля доступа к интерфейсу:

```
    Router(config)# interface FastEthernet0/1
    Router(config-if)# ip address 172.16.1.2
255.255.255.0
    Router(config-if)# ip access-group
inboundfilters in
    Router(config-if)# ip access-group
outboundfilters out
```

Рефлексивные списки контроля доступа могут быть заданы только как расширенные именованные списки контроля доступа протокола IP. Их нельзя задать как нумерованные или стандартные списки доступа протокола IP или как списки доступа другого протокола. Рефлексивные списки контроля доступа можно использовать с другими стандартными и статическими расширенными списками контроля доступа.

Временные списки контроля доступа аналогичны расширенным спискам, но они поддерживают контроль доступа в зависимости от времени. Чтобы внедрить временные списки контроля доступа, необходимо задать временной диапазон для каждого дня и недели. Временной диапазон идентифицируется именем, на которое ссылается утверждение. Поэтому временные ограничения применяются к самому утверждению.

В примере запуск сеансов Telnet разрешён из внутренней сети во внешние сети по понедельникам, средам и пятницам.

Конфигурация ниже определяет временной диапазон списка контроля доступа и назначает ему имя:

```
    Router(config)# time-range EVERYOTHERDAY
    Router(config-time-range)# periodic Monday
Wednesday Friday 8:00 to 17:00
```

Следующая конфигурация применяет временной диапазон к списку контроля доступа:

```
Router(config)# access-list 101 permit tcp
10.1.1.0 0.0.0.255 172.16.1.0 0.0.0.255 eq telnet
time-range EVERYOTHERDAY
```

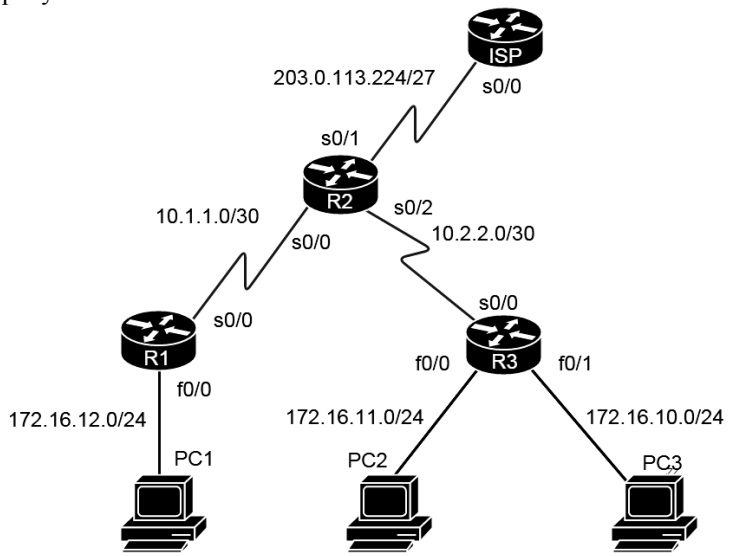
Временной диапазон зависит от системных часов маршрутизатора. Можно использовать часы маршрутизатора, однако эта функция лучше всего работает при настроенной синхронизации NTP.

Контрольные вопросы

1. Принципы настройки и проверки стандартных нумерованных списков контроля доступа.
2. Принципы настройки и проверки расширенных нумерованных списков контроля доступа.
3. Принципы настройки и проверки расширенных и стандартных именованных списков контроля доступа.
4. Принципы поиска и устранения ошибок конфигурации списков контроля доступа.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить адресацию устройств согласно таблице:

Устройство	Интерфейс	IP адрес	Маска	Шлюз
R1	S0/0	10.1.1.12	255.255.255.252	–
	F0/0	172.16.12.1	255.255.255.0	–
R2	S0/0	10.1.1.	255.255.255.252	–
	Lo0	203.0.113.225	255.255.255.255	–
	S0/2	10.2.2.1	255.255.255.252	–
R3	S0/0	10.2.2.2	255.255.255.252	–
	Fa0/0	172.16.10.1	255.255.255.0	–
	Fa0/1	172.16.11.1	255.255.255.0	–
PC1	NIC	172.16.12.10	255.255.255.0	172.16.12.1
PC2	NIC	172.16.11.10	255.255.255.0	172.16.11.1
PC3	NIC	172.16.10.10	255.255.255.0	172.16.10.1

3. Настроить списки контроля доступа:

– на маршрутизаторе R1 запретить входящий трафик из сети 172.16.11.0/24;

– на маршрутизаторе R3 запретить трафик из сети 172.16.10.0/24 к хосту 203.0.113.225;

– запретить доступ на маршрутизатор R2 по протоколу telnet из сети 172.16.12.0/24.

Практическое занятие 8

ТРАНСЛЯЦИЯ СЕТЕВЫХ АДРЕСОВ

Цель: изучение технологий трансляции сетевых адресов.

В результате выполнения практического занятия обучаемые *должны*:

– *знать*: принципы настройки и проверки статического, динамического, перегруженного преобразования NAT; принципы поиска и устранения ошибок конфигурации преобразования NAT.

– *уметь* настраивать и проверять технологию преобразования сетевых адресов NAT.

Практическое занятие включает три этапа:

1. Предварительная подготовка – проработка теоретического материала студентами самостоятельно.

2. Основная часть – устный или письменный опрос, решение предложенных задач.

3. Оформление отчёта и защита полученных результатов.

Отчёт должен быть представлен в печатном виде и содержать:

– краткие ответы на поставленные вопросы;

– решение предложенных задач;

– выводы по каждой задаче и отчёту в целом.

Литература: [2, с. 700 – 732].

1. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Трансляция NAT, определённая в RFC 3022, позволяет узлу, который не имеет действительного, зарегистрированного глобально уникального IP-адреса, осуществлять связь с другими узлами через Интернет. Эти узлы могут использовать частные адреса или адреса, назначенные другим организациям. В любом из этих случаев трансляция NAT позволяет продолжать использование этих адресов, не готовых для Интернета, и осуществлять связь с узлами в Интернете.

Эта цель достигается трансляцией NAT путём использования действительных зарегистрированных IP-адресов для представления данного частного адреса всем остальным узлам Интернета. Функция NAT заменяет частные IP-адреса открытыми зарегистрированными IP-адресами в каждом пакете протокола IP.

Обратите внимание на то, что, выполняя трансляцию NAT, маршрутизатор изменяет IP-адрес отправителя в тот момент, когда пакет покидает организацию. Маршрутизатор, выполняющий NAT, также изменяет адрес получателя каждого пакета, который возвращается назад в частную. Программное обеспечение Cisco IOS поддерживает несколько разновидностей трансляции NAT.

В терминологии NAT под «внутренней сетью» подразумевается набор преобразуемых сетей. Термин «внешняя сеть» относится ко всем остальным адресам. Как правило, подразумеваются действующие адреса, расположенные в Интернете.

Список терминов NAT, используемых компанией Cisco, приводится ниже.

Внутренний локальный адрес (*Inside local*). При проектировании NAT термин «внутренний» относится к адресу, используемому для узла на предприятии. Внутренним локальным называется действующий IP-адрес, назначенный узлу в частной сети предприятия. Более наглядным термином мог бы быть «внутренний частный».

Внутренний глобальный адрес (*Inside global*). При типичном проектировании NAT термин «внутренний» относится к адресу, используемому для узла на предприятии. Трансляция NAT использует внутренний глобальный адрес для представления внутреннего узла, когда пакет пересылается через внешнюю сеть, обычно через сеть Интернет. Маршрутизатор NAT изменяет IP-адрес отправителя в пакете, посылаемом внутренним узлом, с внутреннего локального адреса на внутренний глобальный адрес, в то время, когда пакет пересылается из внутренней сети во внешнюю. Более наглядным мог бы быть термин «внутренний открытый (общедоступный)», поскольку при использовании на предприятии адресов RFC 1918 внутренний глобальный представляет внутренний узел с открытым IP-адресом, который может быть использован для маршрутизации в открытой сети Интернет

Внешний глобальный адрес (*Outside global*). При типичном проектировании NAT термин «внешний» относится к адресу, используемому для узла вне предприятия, иными словами – в Интернете. Внешний глобальный адрес представляет собой реальный IP-адрес, назначенный узлу, который находится в сети, обычно – в сети Интернет. Более содержательным (точным) термином мог бы быть «внешний открытый», поскольку внешний глобальный адрес представляет внешний узел открытым IP-адресом, который может использоваться для маршрутизации в открытой сети Интернет.

Внешний локальный адрес (*Outside local*). NAT может транслировать внешние IP-адреса, т.е. IP-адреса, представляющие узел вне сети предприятия, хотя эта опция не очень популярна. Когда маршрутизатор NAT пересылает пакет из внутренней сети во внешнюю, используя NAT для изменения внешнего адреса, IP-адрес, представляющий внешний узел в качестве IP-адреса получателя в заголовке пакета, называется внешним локальным IP-адресом.

Преобразование NAT может работать в следующих режимах:

Статическое преобразование NAT. В этом случае один внутренний адрес преобразуется в один внешний. При этом все запросы, приходящие на внешний адрес, будут транслироваться на внутренний.

Динамическое преобразование NAT. Ситуация похожа на статический NAT – один приватный адрес транслируется на один внешний, – но теперь внешний не чётко зафиксирован, а будет выбираться динамически из заданного диапазона.

Перегрузка NAT (*overloading*). Несколько приватных адресов преобразуются в один внешний, задействуя возможности транспортного уровня – номера портов.

Статическое преобразование. На рисунке 16 представлена топология с реализованным в ней статическим NAT-преобразованием.

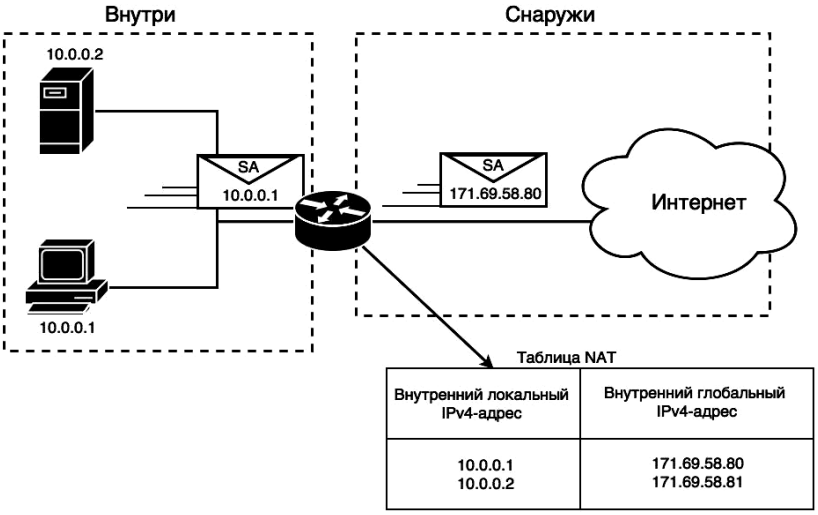


Рис. 16. Технология трансляции адресов

В примере маршрутизатор изменяет адрес отправителя 10.1.1.1 на адрес 171.69.58.80. При использовании статической трансляции NAT маршрутизатор осуществляет взаимно однозначное преобразование между частным адресом и зарегистрированным адресом, от имени которого он выступает. На маршрутизаторе NAT сконфигурировано статическое преобразование частного адреса 10.1.1.1 в открытый зарегистрированный адрес 171.69.58.80.

Поддержка двух IP-узлов в частной сети требует второго взаимно однозначного преобразования с использованием второго IP-адреса в диапазоне открытых адресов. Например, для поддержки адреса 10.1.1.2 маршрутизатор преобразует адрес 10.1.1.2 в адрес 171.69.58.81. Поскольку предприятие имеет одну зарегистрированную сеть класса С, используя трансляцию NAT, оно может поддерживать до 254 частных IP-адресов (при этом зарезервированы два обычных адреса – номер сети и её широковещательный адрес).

В терминологии Cisco сеть предприятия, которая использует частные адреса и, следовательно, требует использования NAT, является «внутренней» частью сети. Интернет-интерфейс трансляции NAT является «внешней» частью сети. Узел, которому необходима трансляция NAT (в данном примере 10.1.1.1), имеет 1P-адрес. который он использует внутри сети, и ему требуется IP-адрес, который будет представлять его вне этой сети. Поскольку узлу фактически нужны два разных адреса для его представления, требуются два термина. В документации Cisco частные IP-адреса, используемые во внутренней сети, называются внутренними локальными адресами, а адреса, используемые для представления узла в Интернете, – внутренними глобальными адресами.

В большинстве типичных конфигураций NAT изменяется только IP-адрес внутренних узлов. Однако внешний IP-адрес узла также может быть изменён с помощью NAT. Когда это происходит, термины «внешний локальный» и «внешний глобальный» означают IP-адреса, используемые для представления этого узла во внутренней сети и во внешней сети соответственно.

Настройка статического преобразования между внутренним локальным адресом и внутренним глобальным адресом:

```
Router(config)# ip nat inside source static
inside-local inside-global
```

Отмечаем интерфейс, как подключённый к внутренней сети:

```
Router(config-if)# ip nat inside
```

Отмечаем интерфейс, как подключённый к внешней сети:

```
Router(config-if)# ip nat outside
```

Просмотр активных процессов преобразования:

```
Router# show ip nat translations
```

Просмотр статистики преобразования:

```
Router# show ip nat statistics
```

Статические соответствия создаются с помощью команды **ip nat inside source static.** Ключевое слово inside означает, что NAT транслирует адреса для узлов, находящихся во внутренней части сети. Ключевое слово source означает, что NAT транслирует IP-адреса отправителя в пакетах, поступающих на ее внутренние интерфейсы. Ключевое слово static означает, что эти параметры определяют статическую запись, которая никогда не должна удаляться из таблицы NAT в связи с истечением времени тайм аута.

После создания записей статической трансляции NAT маршрутизатору нужно знать, какие интерфейсы являются внутренними (inside), а какие – внешними (outside). Подкоманды интерфейса **ip nat inside** и **ip nat outside** идентифицируют соответствующим образом каждый интерфейс.

Команда **show ip nat translations** выводит две записи статической NAT, созданные в конфигурации. Команда show ip nat statistics выводит статистическую информацию, такую как количество активных в данный момент записей в таблице трансляции. Эта статистика также включает в себя количество повторных попаданий (hits), которое увеличивается на единицу с каждым пакетом, для которого NAT должна транслировать адреса.

Пример настройки:

```
    Router(config)# interface Ethernet0/0
    Router(config-if)# ip address 10.1.1.3
255.255.255.0
    Router(config-if)# ip nat inside

    Router(config)# interlace Serial0/0
    Router(config-if)# ip address 200.1.1.251
255.255.255.0
    Router(config-if)# ip nat outside
```

```
    Router(config)# ip nat inside source static
10.1.1.2 200.1.1.2
    Router(config)# ip nat inside source static
10.1.1.1 200.1.1.1
```

Динамическое преобразование. В сравнении со статической динамическая трансляция NAT имеет как сходства, так и отличия. Как и в случае использования статического преобразования, маршрутизатор NAT осуществляет взаимно однозначное преобразование между внутренним локальным и внутренним глобальным адресами и изменяет IP-адреса источника в пакетах, когда они входят во внутреннюю сеть и выходят из неё. Однако преобразование внутренних локальных адресов во внутренние глобальные адреса происходит динамически.

Динамическая трансляция NAT создаёт пул возможных внутренних глобальных адресов и определяет критерий соответствия для определения того, какие внутренние глобальные IP-адреса должны транслироваться с помощью NAT.

Трансляция NAT может быть сконфигурирована с большим количеством IP-адресов в списке внутренних локальных адресов, чем в пуле внутренних глобальных адресов. Маршрутизатор выделяет адреса из пула до тех пор, пока все они не будут выделены. Если поступает новый пакет от ещё одного внутреннего узла и ему требуется запись NAT, а все находящиеся в пуле IP-адреса уже используются, то маршрутизатор просто отбрасывает данный пакет. По существу, размер внутреннего глобального пула адресов должен соответствовать максимальному количеству конкурирующих узлов, которым требуется одновременный доступ к сети Интернет (кроме случая использования трансляции PAT, который описан в следующем разделе).

Конфигурирование динамической NAT в определённой степени отличается от статической NAT, однако имеются и общие черты. Динамический NAT по-прежнему требует идентификации каждого интерфейса как внутреннего или внешнего, и, конечно, уже не нужно задавать статическое преобразование. Динамическая трансляция NAT использует списки управления доступом для указания внутренних локальных (частных) IP-адресов, адреса которых должны транслироваться, и определяет пул зарегистрированных открытых IP-адресов, которые будут выделяться. Эти конкретные действия приведены ниже.

Сначала необходимо задать пул глобальных адресов, которые буду выделяться при необходимости:

```
    Router(config)# ip nat pool name first-address
last-address {netmask subnet-mask | prefix-length
prefix-length}
```

Затем необходимо задать стандартный список контроля доступа по протоколу IP, который разрешает преобразуемые внутренние локальные адреса:

```
Router(config)# access-list access-list-number
permit source [source-wildcard]
```

Задание динамического преобразования источника с использованием списка контроля доступа, заданного во время предыдущего действия:

```
Router(config)# ip nat inside source list
access-list-number pool name
```

Просмотр активных процессов преобразования:

```
Router# show ip nat translations
```

Конфигурирование динамической трансляции NAT включает в себя создание пула внутренних глобальных адресов, а также списка доступа протокола IP для определения внутренних локальных адресов, в отношении которых выполняется адресация NAT. В команде **ip nat pool** указываются первый и последний номера в диапазоне внутренних глобальных адресов. Обязательный параметр **netmask** выполняет нечто вроде проверки действительности диапазона адресов. Если диапазон адресов с учётом используемого параметра **netmask** не окажется в той же самой полсети, то операционная система IOS отвергнет команду **ip nat pool**.

Как и для статической трансляции, динамическая NAT использует команду отправителя **ip nat inside**. Однако в отличие от статической NAT динамическая версия NAT этой команды ссылается на имя пула трансляции NAT, который предполагается использовать для внутренних глобальных адресов – в данном случае этим именем является fred. Она также ссылается на IP-список доступа ACL, который задаёт логику соответствия для внутренних локальных IP-адресов. Команда **ip nat inside source list** 1 pool fred устанавливает соответствие между узлами. отвечающими условиям списка ACL 1, и пулом с именем fred, который был создан командой ip nat pool fred.

Команда **show ip nat statistics** выводит информацию для поиска ошибок конфигурации в двух различных счётчиках, называемых счётчиками пропусков (misses). При первом появлении этого счётчика в нем отображается, сколько раз появился новый пакет, требующий записи NAT, но не получивший её. В этот момент реагирует динамическая трансляция NAT и создаёт необходимую запись. Второй счёт-

чик пропусков (misses) в конце вывода по команде отображает количество пропусков (misses) в пуле. Этот счётчик увеличивает своё значение на единицу, когда динамическая NAT пытается выделить новую запись в таблице NAT и не находит доступного адреса, поэтому пакет не может быть транслирован, что, вероятно, приводит к тому, что конечный пользователь не может получить доступ к приложению.

Пример:

```
    Router(config)# interface Ethernet0/0
    Router(config-if)# ip address 10.1.1.3
255.255.255.0
    Router(config-if)# ip nat inside

    Router(config)# interface Serial0/0
    Router(config-if)# ip address 200.1.1.251
255.255.255.0
    Router(config-if)# ip nat outside

    Router(config)# ip nat pool fred 200.1.1.1
200.1.1.2 netmask 255.255.255.252
    Router(config)# ip nat inside source list 1
pool fred

    Router(config)# access-list 1 permit 10.1.1.2
    Router(config)# access-list l permit 10.1.1.1
```

Отметим, что запись в таблице NAT удаляется (times out) после истечения определённого периода отсутствия активности. Однако можно принудительно удалить запись из таблицы с помощью команды **clear ip nat translation *.**

С помощью команды **debug ip nat** маршрутизатор отправляет сообщение каждый раз, когда адрес пакета транслируется для NAT.

Перегрузка внутреннего глобального адреса. Перезагрузка NAT позволяет ей поддерживать много внутренних локальных адресов при наличии лишь одного или нескольких внутренних глобальных IP-адресов. По существу, транслируя частный IP-адрес и номер порта в один внутренний глобальный адрес, но с уникальным номером порта, трансляция NAT может поддерживать большое количество (более 65 тысяч) частных узлов всего лишь с одним открытым глобальным адресом.

В операционной системе IOS существует две конфигурации PAT. Если PAT использует пул внутренних глобальных адресов, то конфигурации выглядит так же, как и у динамической NAT, за исключением того, что в конце глобальной команды **ip nat inside source list** добавля-

ется ключевое слово **overload**. Если трансляции PAT необходим только один внутренний глобальный IP-адрес, то она может использовать один из своих IP-адресов интерфейсов. Поскольку NAT может поддерживать до 65 тысяч конкурирующих потоков данных, один открытый IP-адрес может удовлетворить потребности NAT целой организации.

Чтобы сконфигурировать трансляцию NAT с перезагрузкой адресов и использованием пула, в описанную выше процедуру конфигурирования нужно добавить только указанный ниже пункт.

Выполните те же действия по конфигурированию динамической трансляции NAT, как было описано в предыдущих разделах, но включите в конце глобальной команды **nat inside source list** ключевое слово **overload**.

Чтобы использовать IP-адрес интерфейса в качестве единственного внутреннего глобального IP-адреса в трансляции NAT с перезагрузкой, следует выполнить описанные ниже действия:

```
Router(config)# ip nat inside source list
access-list-number interface type number overload
```

Пример:

```
Router(config)# interface Ethernet0/0
Router(config-if)# ip address 10.1.1.3
255.255.255.0
Router(config-if)# ip nat inside

Router(config)# interface serial0/0
Router(config-if)# ip address 200.1.1.249
255.255.255.252
Router(config-if)# ip nat outside

Router(config)# ip nat inside source list 1
interface Serial0/0 overload

Router(config)# access-list l permit 10.1.1.2
Router(config)# access list 1 permit 10.1.1.1
```

Трансляция перекрывающихся адресов. Первые три опции трансляции NAT, рассмотренные выше, являются наиболее вероятным решением для большинства сетей. Однако существует ещё одна разновидность трансляции NAT, которая позволяет выполнить трансляцию IP-адресов как отправителя, так и *получателя. Эта опция особенно полезна*, когда две объединённые сети используют перекрывающиеся диапазоны IP-адресов, например когда одна организация вместо част-

ной адресации использует сетевой номер, зарегистрированный другой компанией. Если одна компания некорректно использует сетевой номер, который правильно зарегистрирован другой компанией, и обе они подсоединены к Интернету, то трансляция NAT может быть использована для того, чтобы позволить обеим компаниям связываться как с другими узлами в Интернете, так и друг с другом. Для этого в данном случае NAT транслирует и адрес отправителя, и адрес получателя.

Пример конфигурации:

```
    Router(config)# ip nat pool net-208
171.69.233.208 171.69.233.223 prefix-length 28
    Router(config)# ip nat pool net-10 10.0.1.0
10.0.1.255 prefix-length 24
    Router(config)# ip nat inside source list 1
pool net-208
    Router(config)# ip nat outside source list 1
pool net-10
    !
    Router(config)# interface serial 0/0
    Router(config-if)# ip address 171.69.232.192
255.255.255.240
    Router(config-if)# ip nat outside
    !
    Router(config)# interface ethernet0/0
    Router(config-if)# ip address 192.168.1.94
255.255.255.0
    Router(config-if)# ip nat inside

    Router(config)# access-list 1 permit
192.168.1.0 0.0.0.255
```

В приведённом выше примере адреса в локальной сети используются кем-то ещё в качестве легальных адресов Интернет. Во внешней сети необходимо производить дополнительную трансляцию. Пул net-10 является пулом внешних локальных адресов IP. Выражение **ip nat outside source list** *1* **pool** *net-10* транслирует адреса узлов из внешней перекрывающейся сети в адреса данного пула.

Перенаправление портов. Когда мы только начали говорить про NAT, трансляция у нас была один-в-один и все запросы, приходящие извне автоматически перенаправлялись на внутренний хост. Таким образом можно было бы выставить сервер наружу в Интернет. Но если у вас нет такой возможности – вы ограничены в белых адресах, или не хотите выставлять всем пучком портов его наружу, что делать?

Вы можете указать, что все запросы, приходящие на конкретный белый адрес и конкретный порт маршрутизатора, должны быть перенаправлены на нужный порт нужного внутреннего адреса:

```
Router(config)# ip nat inside source static
{tcp | udp} inside-local port inside-global port
```

Пример:

```
Router(config)# ip nat inside source static tcp
172.16.0.2 80 198.51.100.2 80 extendable
```

Применение данной команды означает, что TCP-запрос, пришедший из интернета на адрес 198.51.100.2 по порту 80, будет перенаправлен на внутренний адрес 172.16.0.2 на тот же 80-й порт. Разумеется, вы можете пробрасывать и UDP и делать перенаправление с одного порта на другой.

Распределения нагрузки TCP. Другая сфера применения NAT не относится к использованию адресов Интернет. Допустим, что организация имеет множество узлов, которые должны подключаться к одному узлу, характеризующемуся высокой степенью загрузки запросами пользователей. Используя NAT можно организовать виртуальный узел во внутренней сети, который будет координировать разделение нагрузки между реальными узлами сети. Адреса узлов назначения, совпадающие с условиями списка доступа, заменяются на адреса из постоянно перебираемого пула. Выбор адреса осуществляется по механизму round-robin, причём такой выбор производится только при установлении нового соединения из внешней сети во внутреннюю. Трафик «не-TCP» передаётся без изменений.

В приведённом ниже примере основной целью определения виртуального адреса 192.168.15.1 является то, что все соединения распределяются между несколькими реальными узлами. Пул адресов real-hosts определяет эти узлы. Список доступа определяет виртуальный адрес. Если в данный момент нет процедуры трансляции, то сегмент TCP с интерфейса Serial 0/0 (внешний интерфейс), адрес назначения которого удовлетворяет условиям списка доступа, транслируется в один из адресов адресного пула.

```
    ip nat pool real-hosts 192.168.15.2
192.168.15.15 prefix-length 28 type rotary
    ip nat inside destination list 2 pool real-
hosts
    !
```

```
interface serial 0/0
ip address 192.168.15.129 255.255.255.240
ip nat outside
!
interface ethernet 0/0
ip address 192.168.15.17 255.255.255.240
ip nat inside
!
access-list 2 permit 192.168.15.1
```

Контрольные вопросы

1. Принципы настройки и проверки статического преобразования NAT.

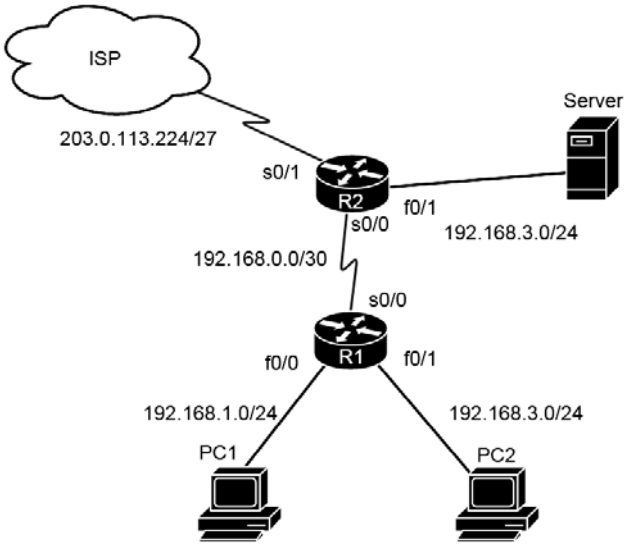
2. Принципы настройки и проверки динамического преобразования NAT.

3. Принципы настройки и проверки перегруженного преобразования NAT (PAT).

4. Принципы поиска и устранения ошибок конфигурации преобразования NAT.

Задача для самостоятельного решения

1. Собрать физическую топологию согласно схеме, изображённой на рисунке.



2. Настроить адресацию устройств согласно таблице:

Устройство	Интерфейс	IP адрес	Маска	Шлюз
R1	S0/0	192.168.0.1	255.255.255.252	–
	Fa0/0	192.168.1.1	255.255.255.0	–
	Fa0/1	192.168.2.1	255.255.255.0	–
R2	S0/0	192.168.0.2	255.255.255.252	–
	S0/1	203.0.113.225	255.255.255.224	–
	Fa0/0	192.168.3.1	255.255.255.0	–
ISP	S0/1	203.0.113.226	255.255.255.224	–
PC1	NIC	192.168.1.11	255.255.255.0	192.168.1.1
PC2	NIC	192.168.2.11	255.255.255.0	192.168.2.1
Server	NIC	192.168.3.254	255.255.255.0	192.168.3.1

3. Настроить трансляцию адресов:

– на маршрутизаторе R2 настроить статическую трансляцию адреса сервера в адрес 203.0.113.254;

– на маршрутизаторе R2 настроить динамическую трансляцию адресов сетей 192.168.1.0/24 и 192.168.2.0/24 в пул адресов 203.0.113.240/29;

– на маршрутизаторе R2 настроить перезагрузку адресов сетей 192.168.1.0/24 и 192.168.2.0/24 в интерфейс Serial 0/1.

ЗАКЛЮЧЕНИЕ

С каждым годом сетевые технологии усложняются. Из-за чего усложняется структура и принципы организации сетей передачи данных, растёт потребность в эффективной передаче трафика в условиях быстро меняющихся способов сетевого взаимодействия. Современные инфокоммуникационные системы и сети представляют сложный комплекс разнообразных технических средств, обеспечивающих передачу различных сообщений на любые расстояния с заданными параметрами качества. Основу инфокоммуникационных систем и сетей составляют многоканальные системы передачи по электрическим, волоконно-оптическим кабелям и радиолиниям, предназначенные для формирования типовых каналов и трактов.

Объем информации, передаваемой через информационно-телекоммуникационную инфраструктуру мира, удваивается каждые 2-3 года. Появляются и успешно развиваются новые отрасли информационной индустрии, существенно возрастает информационная составляющая экономической активности субъектов рынка и влияние информационных технологий на научно-технический потенциал. Начало XXI века рассматривается как эра информационного общества, требующегодля своего эффективного развития создания глобальной информационно-телекоммуникационной инфраструктуры, темпы развития которой должны быть опережающими по отношению к темпам развитияэкономики в целом.

В данном учебном пособии излагаются общие сведения по статистической теории передачи информации, принципы построения и характеристики инфокоммуникационных систем, отдельные вопросы функционирования информационных сетей. Последний раздел дополнен основными задачами администрирования информационных сетей, построенных с использованием оборудования компании CiscoSystems. В пособии представлены способы решения задач оптимизации передачи трафика, с которыми часто сталкиваются сетевые инженеры при проектировании, реализации и поддержке современных инфокоммуникационных систем.

Учебное пособие предназначено для студентов высших учебных заведений, обучающихся по направлениям подготовки 09.03.02 «Информационные системы и технологии» и 10.05.03 «Информационная безопасность автоматизированных систем».

СПИСОК ЛИТЕРАТУРЫ

К части I

1. **Громов, Ю. Ю.** Теоретические основы передачи сигналов : учебное пособие / Ю. Ю. Громов, И. Г. Карпов, Г. Н. Нурутдинов. – Тамбов : Изд-во МИНЦ «Нобелистика», 2010. – Ч. 1. – 138 с.
2. **Громов, Ю. Ю.** Теоретические основы передачи сигналов : учебное пособие / Ю. Ю. Громов, И. Г. Карпов, Г. Н. Нурутдинов. – Тамбов : Изд-во МИНЦ «Нобелистика», 2010. – Ч. 2. – 140 с.
3. **Компьютерные** телекоммуникации [Электронный ресурс] : учебное пособие / Ю. Ю. Громов, В. Е. Дидрих, И. В. Дидрих и др. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 224 с.
4. **Информационные** сети : учебное пособие / Ю. Ю. Громов и др. – Тамбов : Изд-во ТГТУ, 2008. – 120 с.

Громов, Ю. Ю. Системы и сети передачи информации [Электронный ресурс] : учебное пособие / Ю. Ю. Громов, И. Г. Карпов, Г. Н. Нурутдинов. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2012. – 140 с.

К части II

1. **Олифер, В. Г.** Компьютерные сети. Принципы, технологии, протоколы : учебник / В. Г. Олифер, Н. А. Олифер – СПб. : Изд-во «Питер», 2014 – 992 с.
2. **Одом, Уэнделл.** Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101 / Уэнделл Одом. – М. : Вильямс, 2015. – 912 с.
3. **Одом, Уэнделл.** Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-101. Маршрутизация и коммутация / Уэнделл Одом. – М. : Вильямс, 2015. – 736 с.
4. **RFC 2453.** Malkin G. / RIP Version 2 / G. Malkin. – Network Working Group, 1998. – 39 p.
5. **RFC 2328.** Moy J. / OSPF Version 2 / J. Moy. – Network Working Group, 1998. – 244 p.
6. **IP Routing:** EIGRP Configuration Guide, Cisco IOS Release 15M&T / Cisco Systems, Inc. – 2015. – 268 p.

7. **IP Routing:** OSPF Configuration Guide, Cisco IOS Release 15M&T / Cisco Systems, Inc. – 2014. – 440 p.
8. **IP Routing:** Protocol-Independent Configuration Guide, Cisco IOS Release 15M&T / Cisco Systems, Inc. – 2014. – 248 p.
9. **IP Routing:** RIP Configuration Guide, Cisco IOS Release 15M&T / Cisco Systems, Inc. – 2014. – 86 p.
10. **Security** Configuration Guide: Access Control Lists, Cisco IOS Release 15M&T/ Cisco Systems, Inc. – 2014. – 278 p.

ОГЛАВЛЕНИЕ

Учебное издание

КАРПОВ Иван Георгиевич
НУРУТДИНОВ Геннадий Нурисламович
ЯКОВЛЕВ Алексей Вячеславович
ЕЛИСЕЕВ Алексей Игоревич
ПОЛЯКОВ Дмитрий Вадимович
ОДНОЛЬКО Валерий Григорьевич

ИНФОКОММУНИКАЦИОННЫЕ СИСТЕМЫ И СЕТИ.

ПРАКТИКУМ

Учебное пособие

Редактор З. Г. Чернова
Инженер по компьютерному макетированию Т. Ю. Зотова

ISBN 978-5-8265-1597-6

Подписано в печать 30.06.2016.
Формат 60×84 /16. 13,72 усл. печ. л.
Тираж 100 экз. Заказ № 318

Издательско-полиграфический центр
ФГБОУ ВО «ТГТУ»
392000, г. Тамбов, ул. Советская, д. 106, к. 14
Тел. 8(4752) 63-81-08;
E-mail: izdatelstvo@admin.tstu.ru