Ахборот тизимлари протоколлари (АТП)

(Протоколы Информационных систем)

Кувнаков А.Э.

Содержание

- Декомпозиция управления
- Архитектура TCP/IP
- Протокол IP
- Транспортные протоколы
- Литература

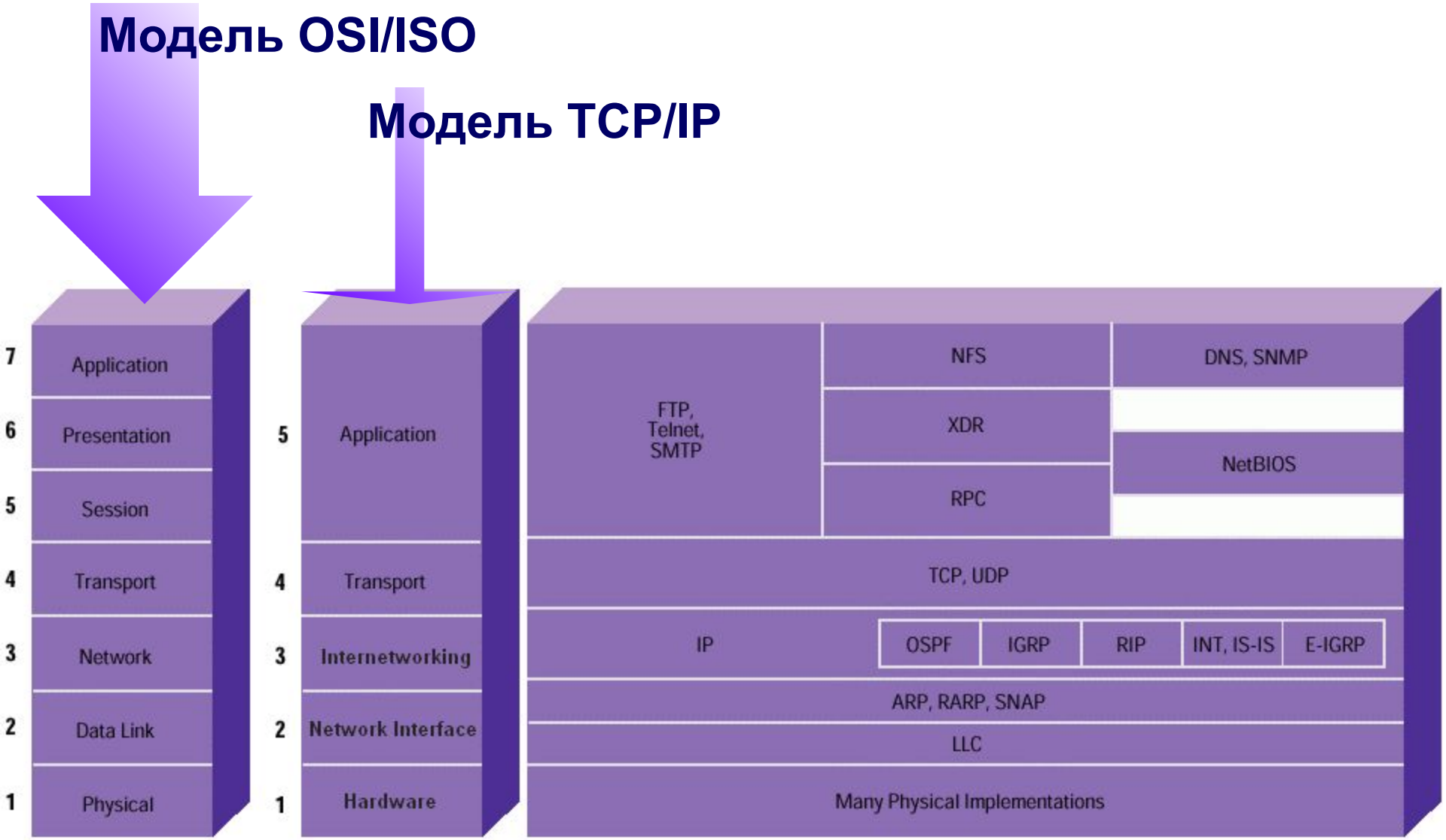
Понятие протокола

- [Сетевой] протокол – это распределенный алгоритм, выполняемый на нескольких компьютерах, и в основном является языком и правилом общения.
  - основной и первичной задачей сетевых протоколов было обеспечение передачи данных
    - собственно организация сервиса передачи данных
    - организация маршрутизации
    - сигналлинг, мониторинг, управление
    - прикладные протоколы: электронная почта терминальный доступ и т.п.
  - сейчас понятие сетевого протокола в целом сохраняется, но
    - просматривается явная тенденция к усложнению протоколов
    - число протоколов стремительно растет, причем наибольшие «темпы роста» показывают протоколы прикладного уровня

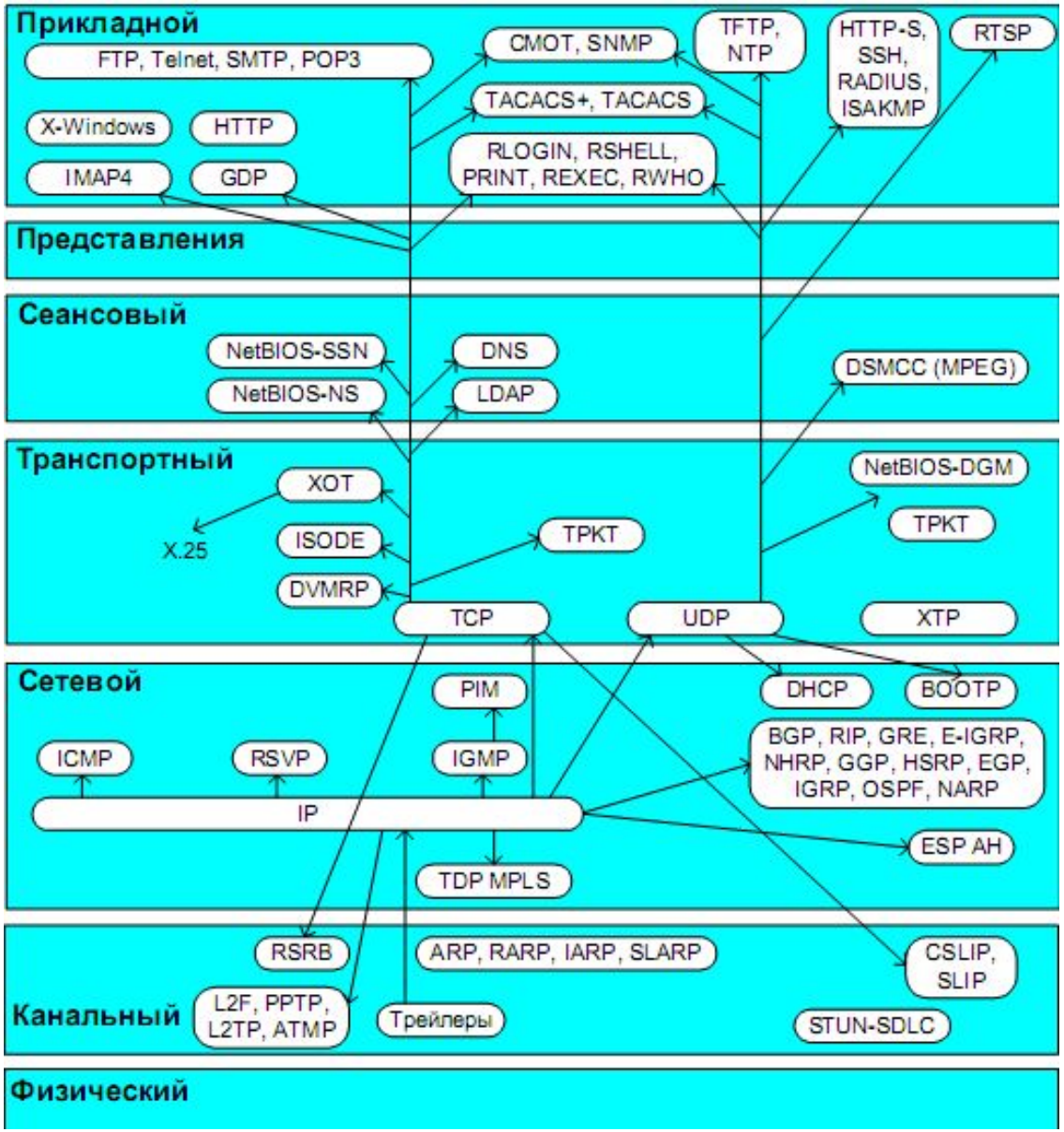
Зачем нужна декомпозиция управления?

1. Задачи сетевого управления сложны
2. Сеть гетерогенна, состоит из множества устройств, выпускаемых различными производителями; без стандартизации функциональности и способов взаимодействия этих компонент сеть не будет работать

Декомпозиция сетевого управления



Наиболее важные протоколы

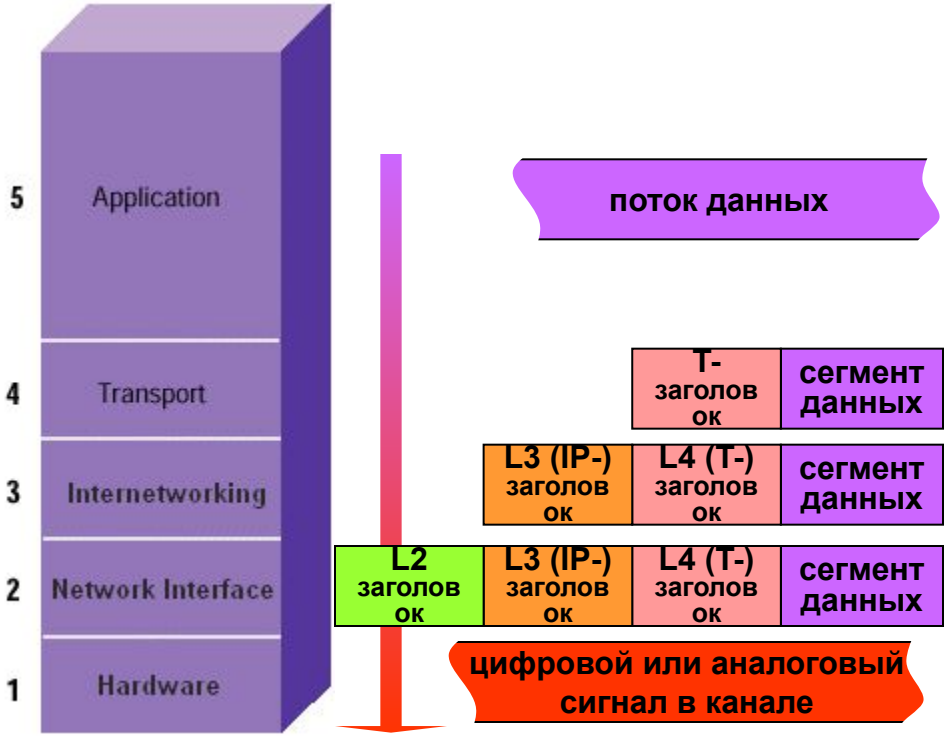


- состав протоколов и их взаимодействия

Задачи протоколов различных уровней

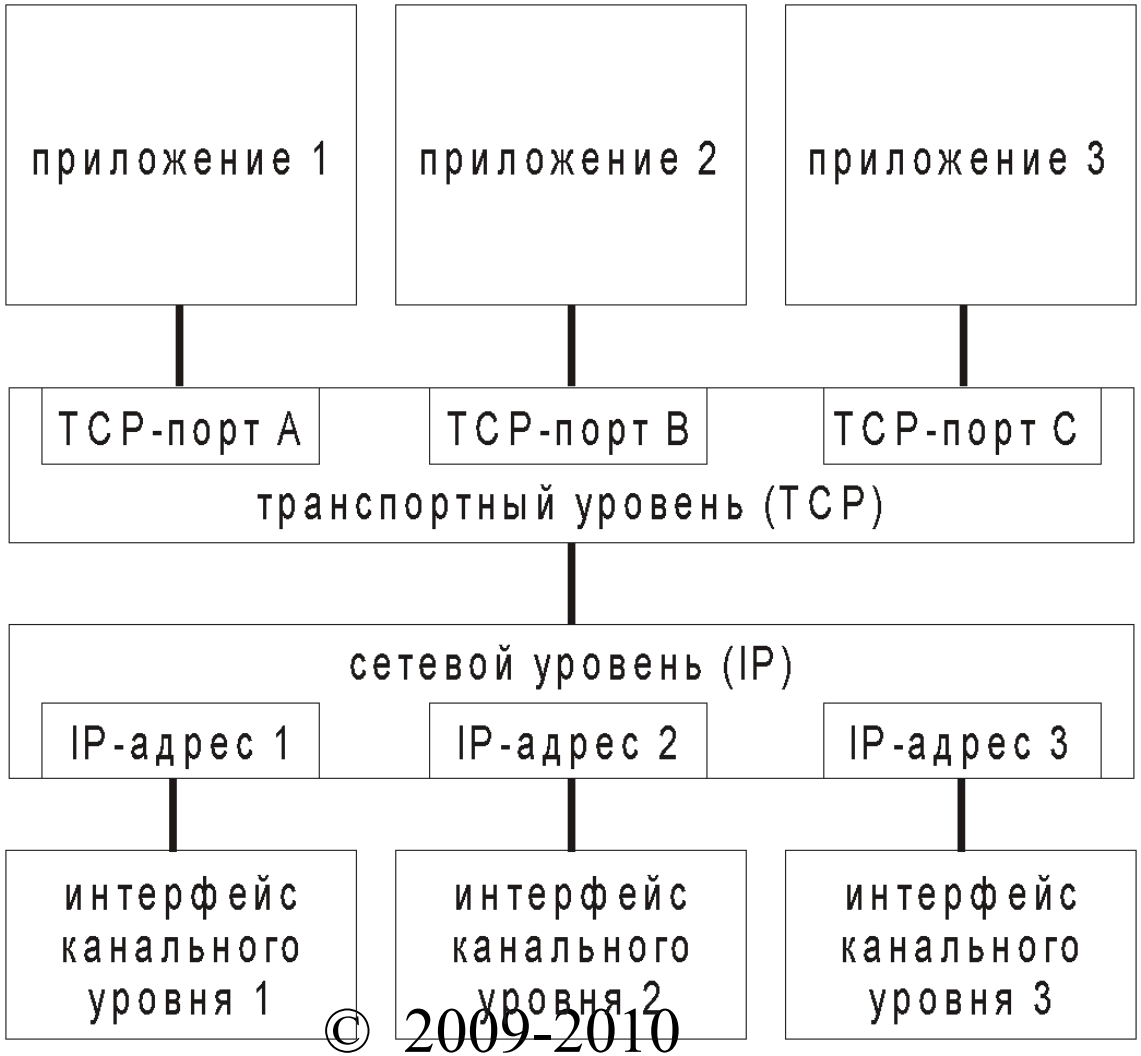
1. Физический уровень
   - обеспечивает стандартизацию сред передачи, (носителей, частотой, контактов, типов модуляции, сигналов)
2. Канальный уровень
   - обеспечивает передачу данных
   - передает *фреймы* (frame), иногда говорят - кадры
3. Сетевой уровень
   - обеспечивает передачу данных из конце в конец сети; делает сеть связной
   - передает *пакеты*
   - решает необходимую для передачи пакетов из конца в конец задачу маршрутизации
4. Транспортный уровень
   - обеспечивает взаимодействие приложений, определяет какому приложению доставить поток данных или сообщение
   - передает *сегменты данных* (TCP) или *дейтаграммы* (UDP)
5. Прикладной уровень
   - решает специфические, утилитарные, необходимые скорее человеку, чем системе, задачи

Инкапсуляция протоколов



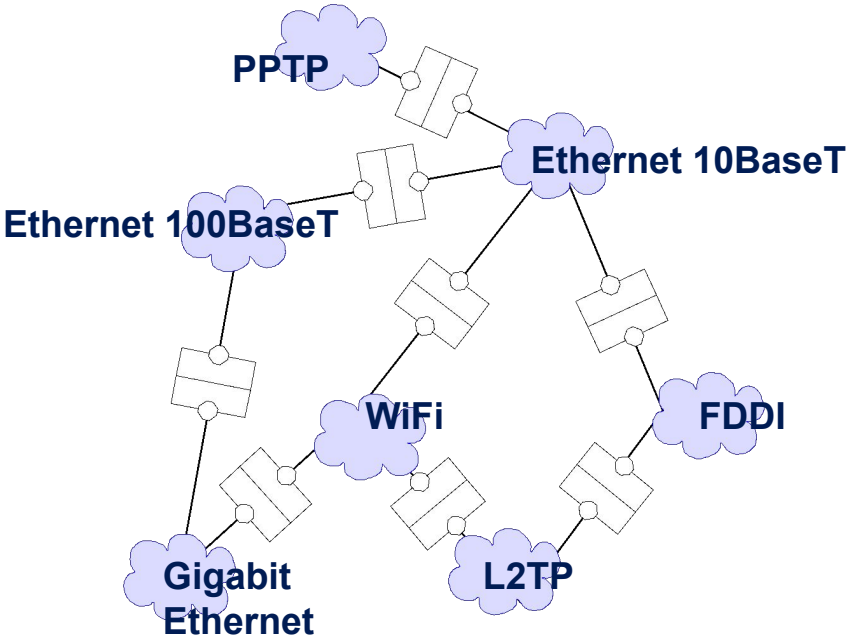
- При передаче данных от приложения в сеть транспортный, сетевой и канальный уровень последовательно упаковывают (инкапсулируют) данные «внутрь» своего пакета
  - такое включение протокола в протокол часто повторно применяется и в рамках одного и того же уровня управления
  - эта техника передачи одного протокола под заголовком («под видом») другого называется *туннелированием*

Техника туннелирования трафика имеет фундаментальное значение в сетевой информационной безопасности. На ней построена архитектура IPsec и многие другие решения


приложение 1
приложение 2
приложение 3
TCP-порт A
TCP-порт B
TCP-порт C
транспортный уровень (TCP)
сетевой уровень (IP)
IP-адрес 1
IP-адрес 2
IP-адрес 3
интерфейс канального уровня 1
интерфейс канального уровня 2
интерфейс канального уровня 3

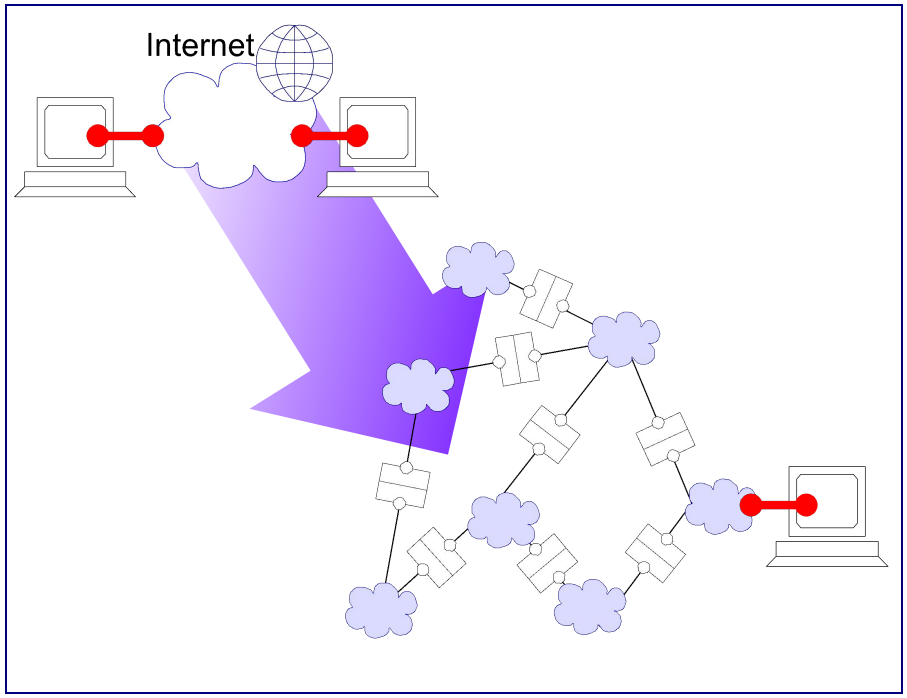
Архитектура TCP/IP

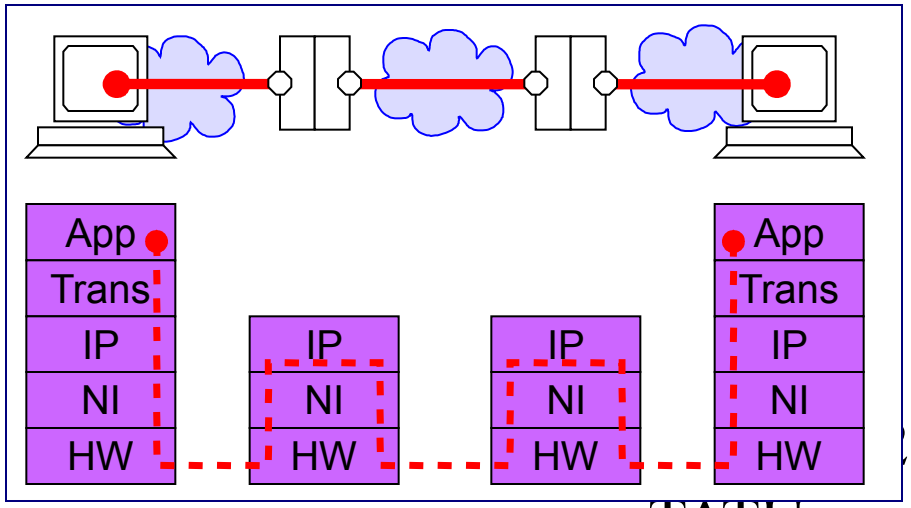
TCP/IP: независимость от среды передачи



- Физические подсети могут иметь различную природу и различные системы адресации канального уровня
- Стек TCP/IP устроен так, что от физической природы линии связи зависят только протоколы физического и канального уровней
- IP и вышележащие протоколы абстрактны и «обязаны» работать «поверх» всех физических сетей, независимо от их природы

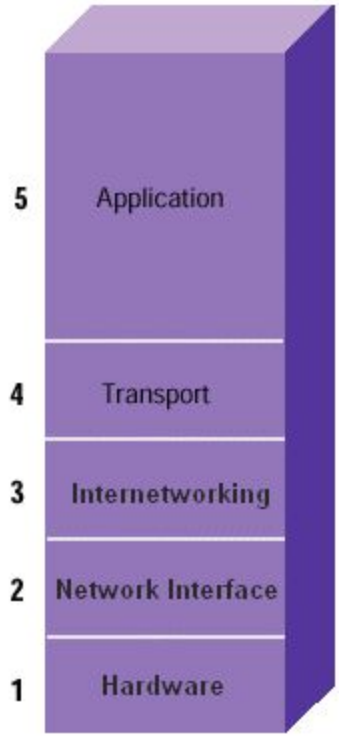
IP-сети





- Сеть – это совокупность *подсетей*, соединенных *шлюзами* (маршрутизаторами)
  - *подсеть* – это целостное адресуемое пространство (в терминах IP-адресов)
  - IP-адрес – уникальное число, приписываемое сетевому интерфейсу; по IP-адресу находится получатель пакета (детали позднее)
  - *шлюз* – машина с 2мя (или более) сетевыми интерфейсами, «смотрящими» в разные подсети
- Поток данных передается от приложения к приложению га оконечных устройствах, но на промежуточных устройствах (шлюзах) используются только три нижних уровня сетевого стека

Функциональная декомпозиция TCP/IP



- Канальный уровень – обеспечивает *двухточечную связность*
- IP (RFC791, 950, 919, 922, 2474) – обеспечивает *негарантированную дейтаграммную* доставку пакетов по сети; 3 главных задачи IP и вспомогательных протоколов:
  - адресация сетевых объектов (включая конфигурирование адресов)
  - маршрутизация
  - обмен служебной информацией, разрешение конфликтных ситуаций, диагностика
- 2 главных задачи транспортных протоколов:
  - обеспечение заданного сервиса доставки данных
  - мультиплексирование/демультиплексирование трафика приложений

Мосты, маршрутизаторы и коммутаторы

Оборудование сетей

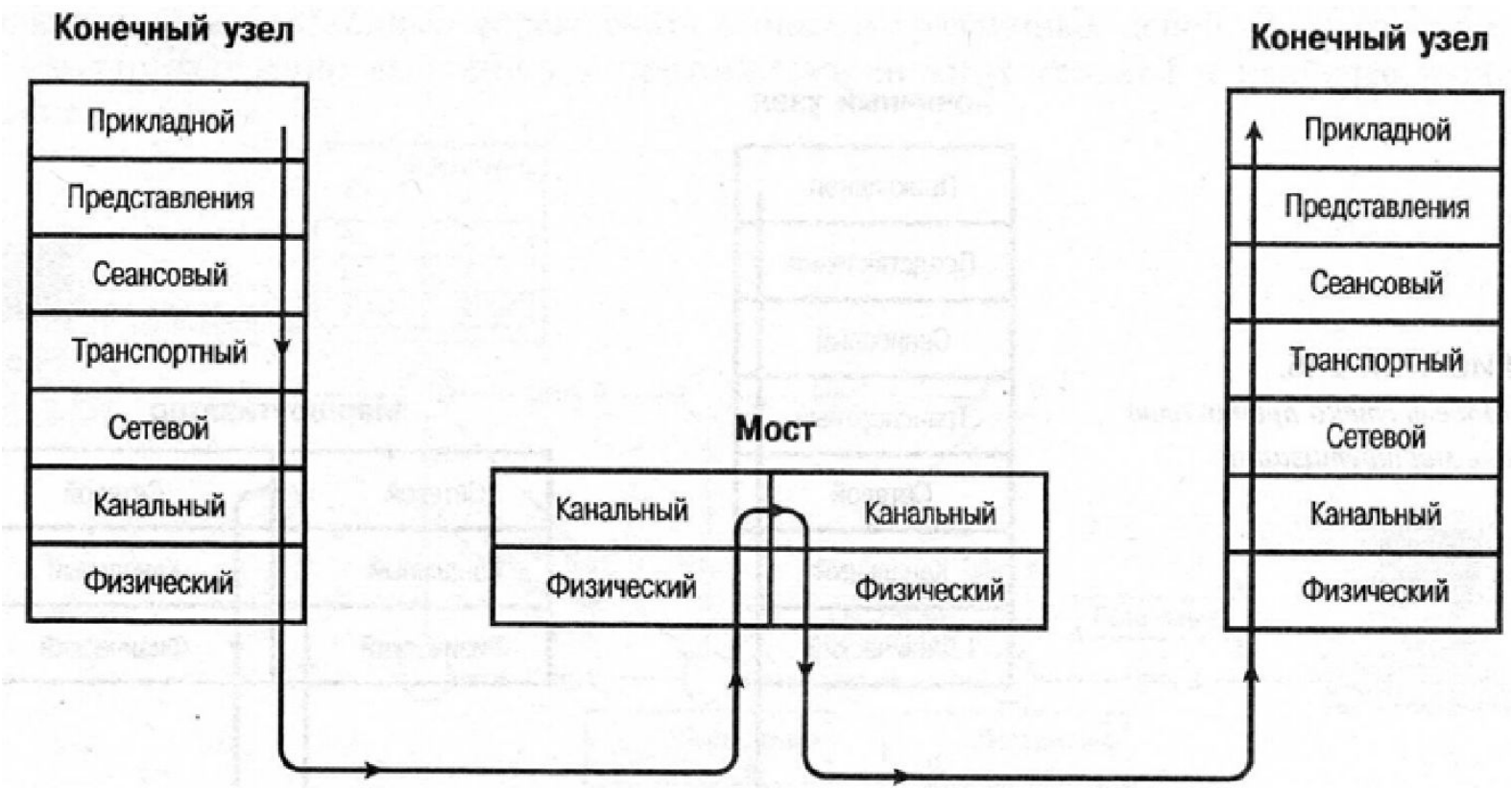
Данные могут передаваться в объединенной сети с помощью следующих трех типов информации:

- **Физический адрес устройства назначения, находящийся на канальном уровне.** Устройства, направляющие сообщения исходя из физических адресов, называются *мостами (brigdes)*

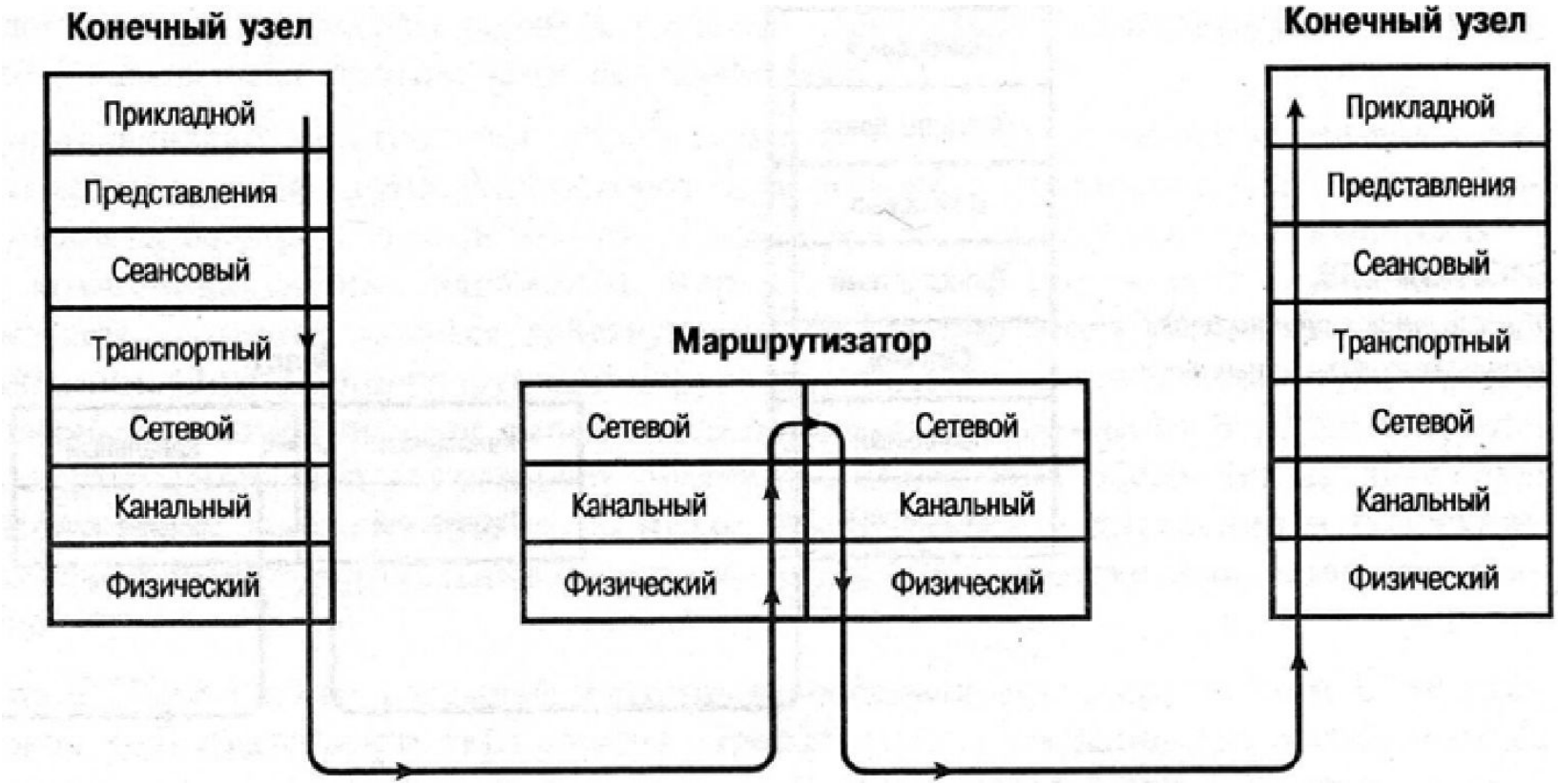
- **Адрес сети назначения, находящийся на сетевом уровне.** Устройства, использующие для направления сообщения адрес сети, обычно называются *маршрутизаторами (router),* хотя первоначальным их названием, все еще используемым в мире TCP/IP, является *шлюз (gateway).*

- **Цепь, устанавливаемая для определенного соединения.** Устройства, направляющие сообщения по выделенным цепям, называются *коммутаторами (switches).*

Мосты


Конечный узел
Прикладной
Представления
Сеансовый
Транспортный
Сетевой
Канальный
Физический
Мост
Канальный
Физический
Канальный
Физический
Конечный узел
Прикладной
Представления
Сеансовый
Транспортный
Сетевой
Канальный
Физический

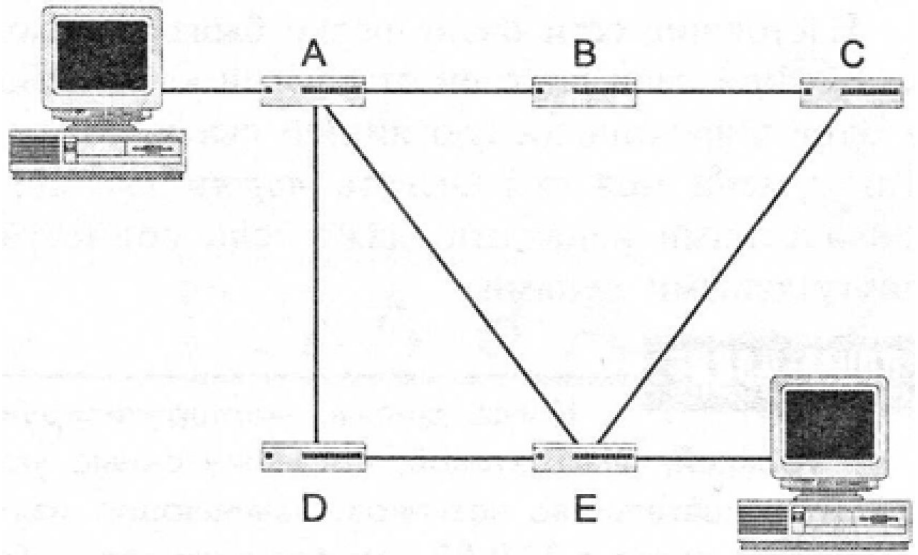
Маршрутизация



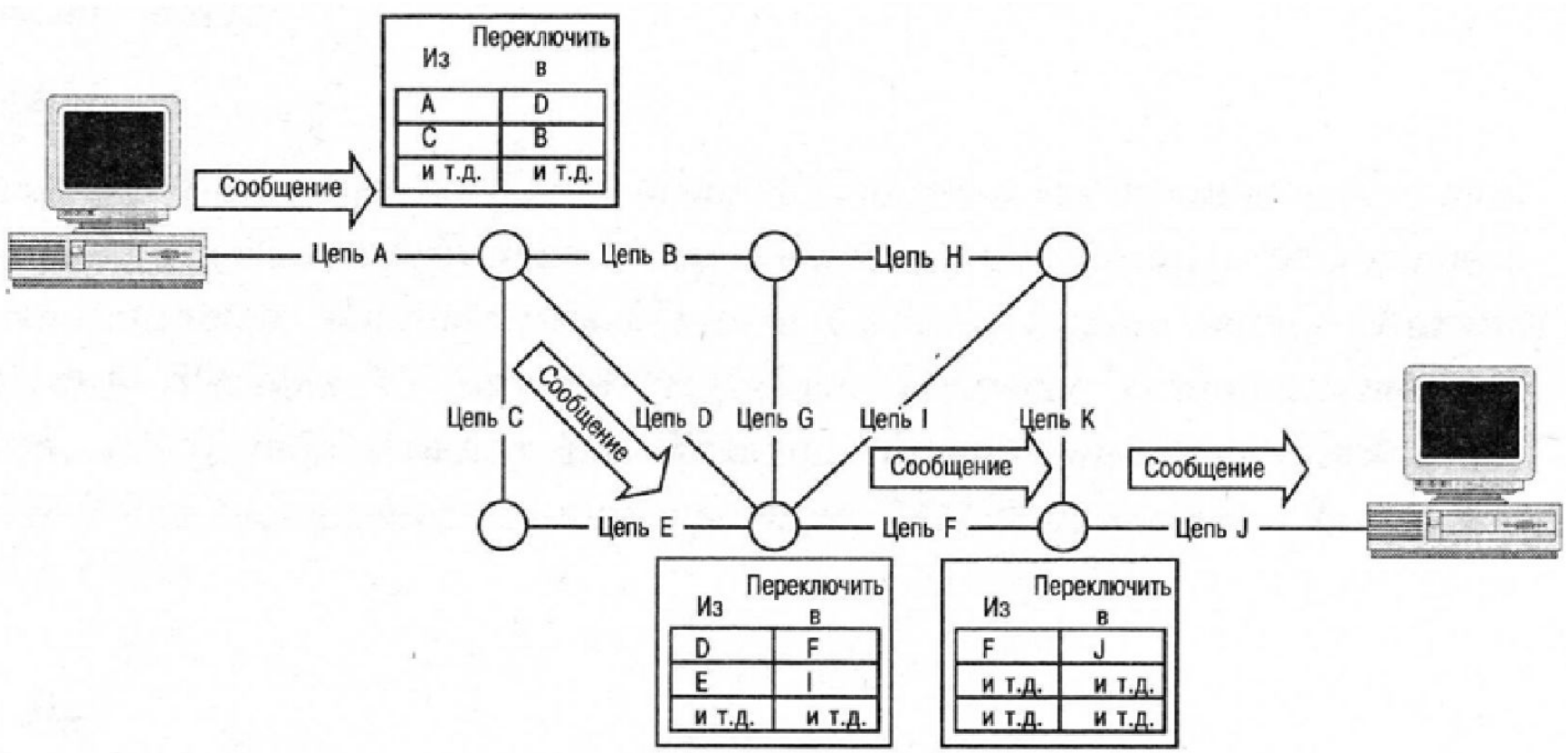
Маршрутизация

- Счетчик пересылок определяет количество сетей, которые необходимо пересечь, чтобы перейти из одного конечного узла в другой. Между узлами A и E можно определить несколько маршрутов:

- A-B-C-D-E (пять пересылок)
- A-E (три пересылки)
- A-D-E (четыре пересылки)



Коммутация

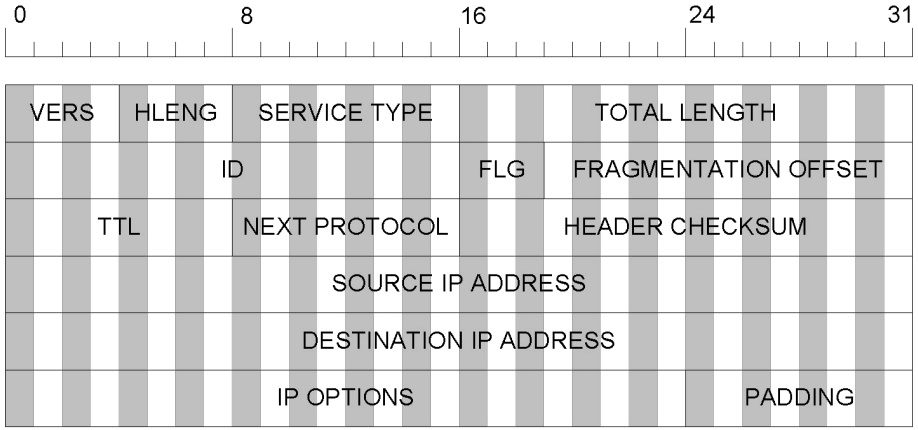

Переключить
Из
в
A
D
C
B
и т.д.
и т.д.
Сообщение
Цепь A
Цепь B
Цепь H
Цепь C
Сообщение
Цепь D
Цепь G
Цепь I
Цепь K
Сообщение
Сообщение
Цепь E
Цепь F
Цепь J
Переключить
Из
в
D
F
E
I
и т.д.
и т.д.
Переключить
Из
в
F
J
и т.д.
и т.д.
и т.д.
и т.д.

Протокол IP: IP-пакет

IP пакет

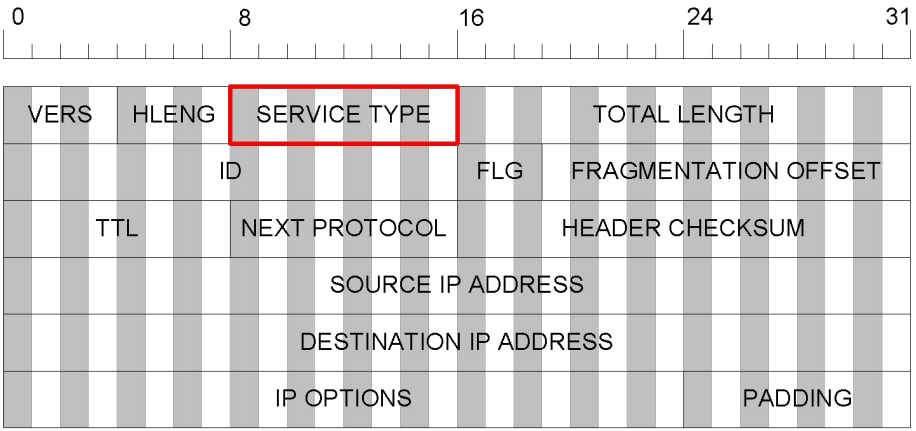
Структура IP-пакета

Заголовок IP-пакета



- VERS, version – версия IP (4, 0010)
- HLENG, header length – длина заголовка
- TOTAL LENGTH – полная длина пакета
- ID, identification – номер (идентификатор) фрагмента
- FLG, flags - флаги
- FRAGMENTATION OFFSET – начало фрагмента
- TTL, time to live – время жизни
- NEXT PROTOCOL – следующий протокол
- HEADER CHECKSUM – контрольная сумма заголовка
- SOURCE IP ADDRESS – адрес отправителя
- DESTINATION IP ADDRESS – адрес получателя
- IP OPTIONS – варианты
- PADDIND – заполнение

Тип сервиса



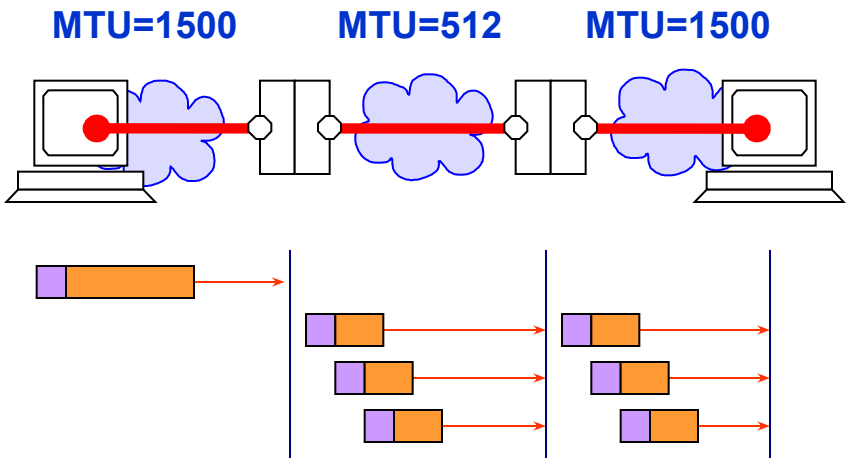
Структура поля SERVICE TYPE

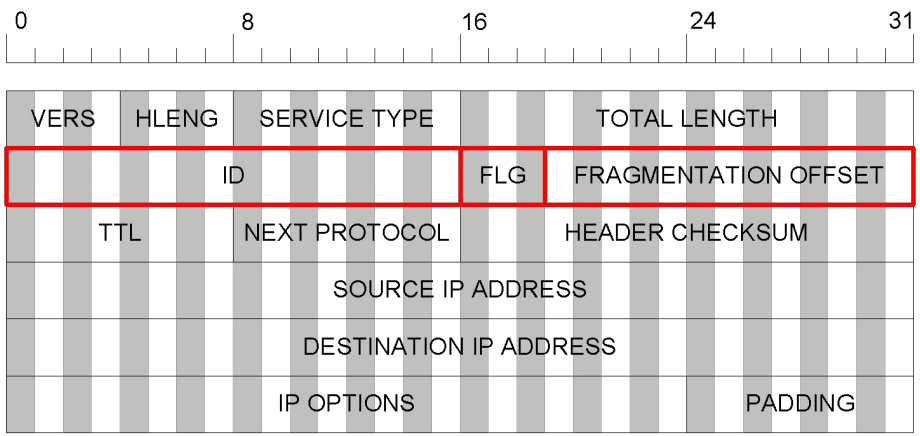
На обработке битов поля TOS строятся современные механизмы управления качеством сервиса (Quality of Service, QoS) для передачи голосового и видеотрафика

- Поле SERVICE TYPE используется для управления приоритетом (качеством сервиса)
  - PRED, predecence – приоритет:
    - 000: Routine
    - 001: Priority
    - 010: Immediate
    - 011: Flash
    - 100: Flash override
    - 101: Critical
    - 110: Internetwork control
    - 111: Network control
  - TOS, type of service – тип сервиса:
    - 1000: Minimize delay
    - 0100: Maximize throughput
    - 0010: Maximize reliability
    - 0001: Minimize monetary cost
    - 0000: Normal service
  - MBZ – зарезервировано для последующего использования

© 2009-2010 TATU

Фрагментация





Структура поля FLG:

- Физические сети могут иметь различные размеры кадров (minimal transfer unit, MTU)
  - если на пути пакета встречается сеть с MTU менее его размера, пакет фрагментируется
  - фрагменты «собирает» в исходный пакет получатель
- Управляют фрагментацией поля ID, FLG, FRGMTTN OFFGSET
  - ID –уникальный идентификатор, единый для всех фрагментов серии
  - поле FLG:
    - 1й бит – резерв, всегда 0
    - 2й бит – DF, Do not Fragment – запрещает фрагментацию
    - бит MF – More Fragments – 0 для нефрагментированного или последнего пакета в серии, 1 – в противном случае

© 2009-2010 TATU

Пример: Размер пакета

- Имеется e-mail с размером 3500 байт (3,5Кбайт). Сеть которая вы подключены поддерживает фиксированную размер пакета равное 1024 байт (1Кбит), 1024-96-32=896. т.е размер данных равно 896 байтам.
- 3500/896=896+896+896+812

Packet - E-mail Example

Header	Sender's IP address Receiver's IP address Protocol Packet number	96 bits
Payload	Data	896 bits
Trailer	Data to show end of packet Error correction	32 bits

©2000 How Stuff Works

MTU: Maximum Transfer Unit

- Каждая сеть отправляет информацию в виде пакета. MTU показывает размер пакета. Варьированием этого параметра мы можем получить различные скорости.
  - Чем выше значение MTU, тем меньше заголовков передаётся по сети — а значит, выше пропускная способность.
    - Но увеличение MTU приводят к большим задержкам для других пользователей.

Пример: Передача пакета с размерностью 12000 байт

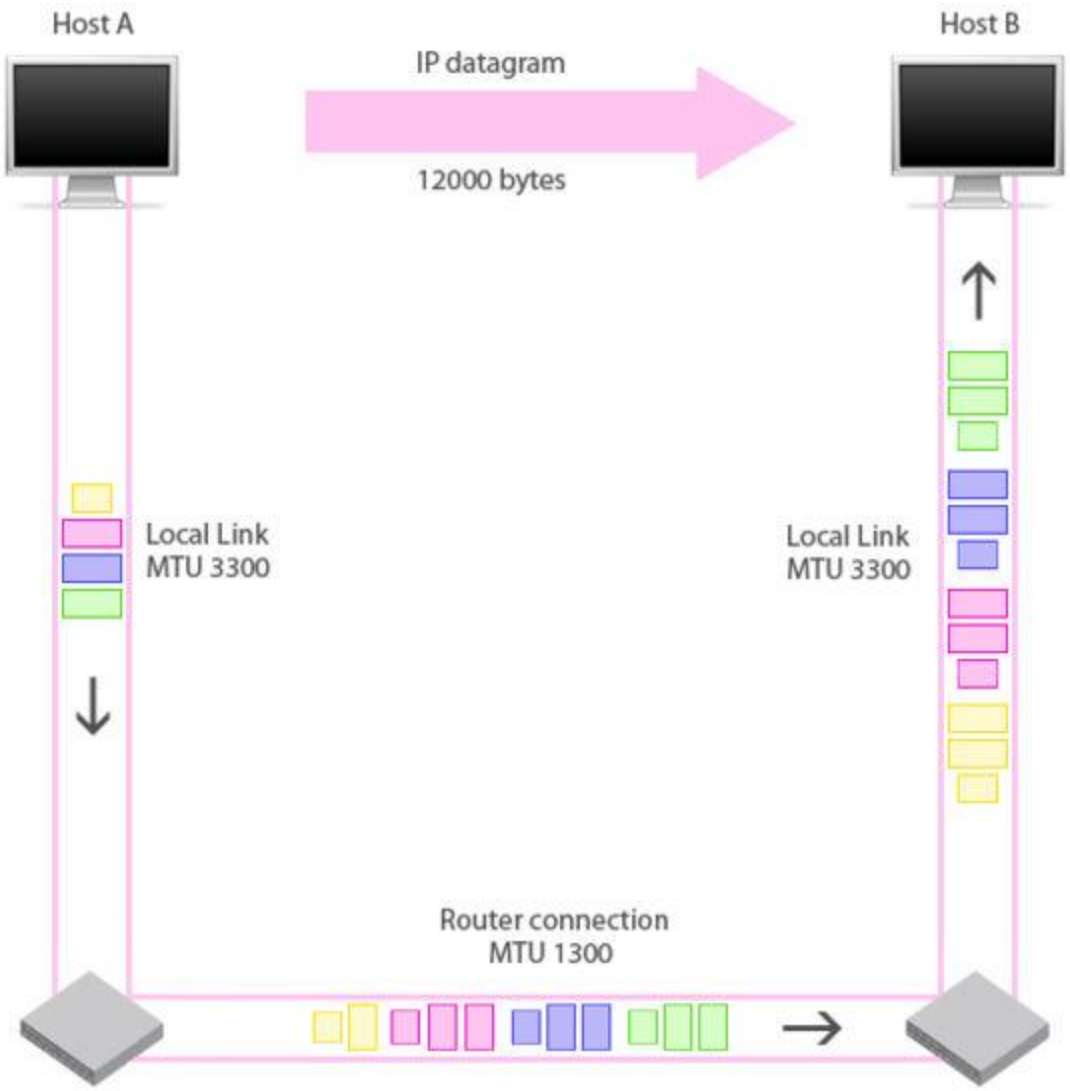
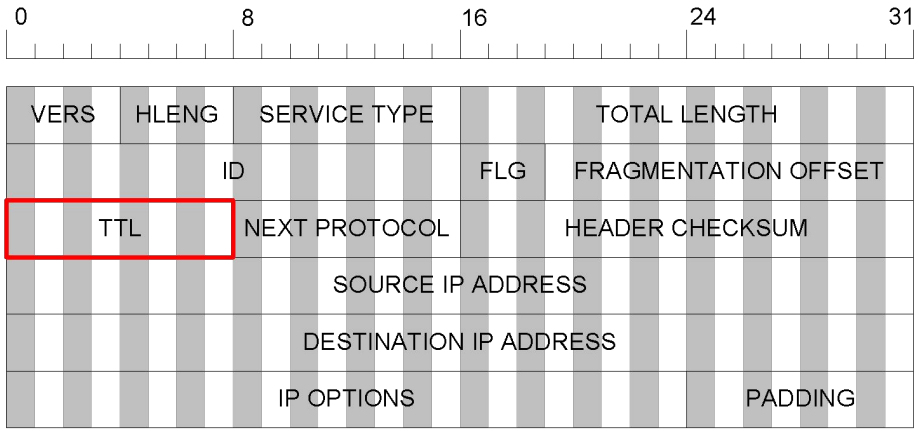


Таблица основных значений MTU

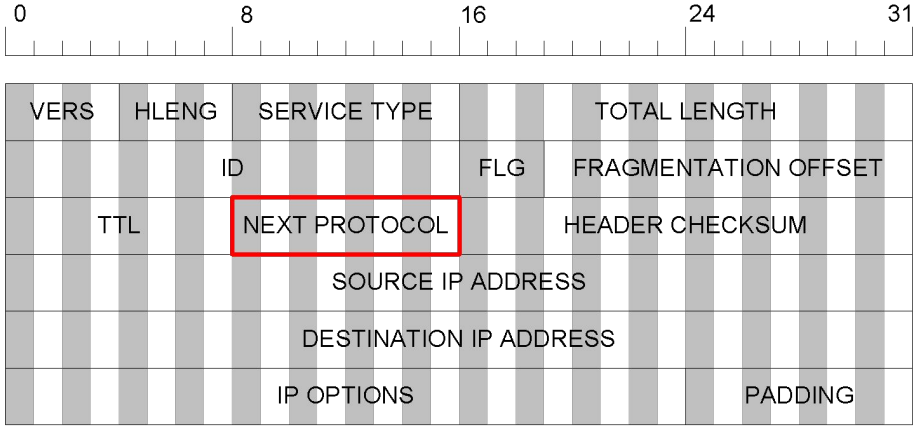
Интерфейс	Значение MTU, байт
Internet **Path MTU (***RFC 879***)**	576
802.3 **(***RFC 1191***)**	1492
802.11	2272
802.5 ***(расформировано)***	4464
FDDI (*RFC 1191***)**	4500

Время жизни IP пакета



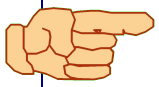
- В силу ошибок маршрутизации или по другим причинам пакет может бесконечно циркулировать по некоторому пути в сети
  - поскольку маршрутизатор обрабатывает IP в дейтаграммном режиме, т.е. «забывает» о всех переданных пакетах (не хранит предысторию) – такие пакеты могут «бродить по сети» вечно
  - чтобы устранить перегрузку сети такими пакетами, введено поле TTL
    - хост-отправитель устанавливает TTL в некоторое заданной значение, отличное от нуля
    - при всякой переретрансляции промежуточные маршрутизаторы уменьшают значение TTL на единицу
    - когда поле TTL принимает значение 0 – пакет изымается из сети

Механизм IP-инкапсуляции



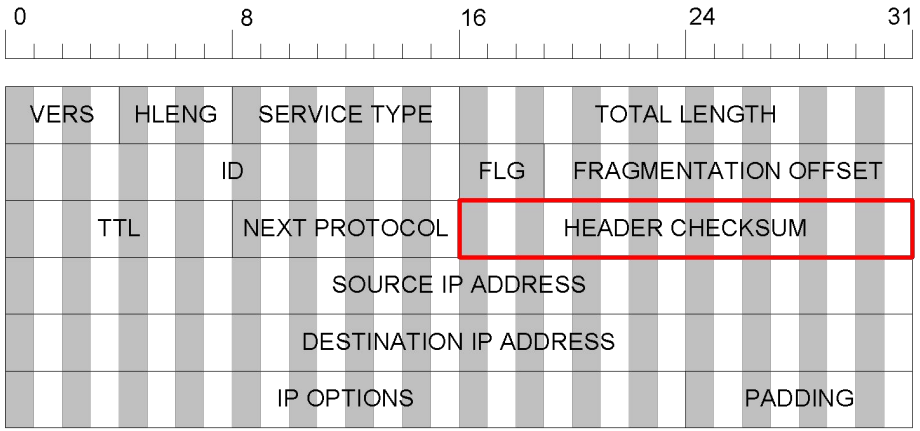
- IP может «нести» данные различных протоколов, номер «вложенного» протокола кодируется в поле NEXT PROTOCOL:
  - 0: Reserved
  - 1: Internet Control Message Protocol (ICMP)
  - 2: Internet Group Management Protocol (IGMP)
  - 3: Gateway-to-Gateway Protocol (GGP)
  - 4: IP (IP encapsulation)
  - 5: Stream
  - 6: Transmission Control Protocol (TCP)
  - 8: Exterior Gateway Protocol (EGP)
  - 9: Private Interior Routing Protocol
  - 17: User Datagram Protocol (UDP)
  - 41: IP Version 6 (IPv6)
  - 50: Encap Security Payload (ESP)
  - 51: Authentication Header (AH)
  - 89: Open Shortest Path First (OSPF)

Обратите внимание на множественность механизмов туннелирования трафика, заложенных в IP: IP может «нести» не только транспортные (TCP, UDP), служебные (ICMP, IGMP, GGP, EGP, OSPF), протоколы сетевой защиты (AH и ESP), но также нести «себя» (IP-IP инкапсуляция), IPv6



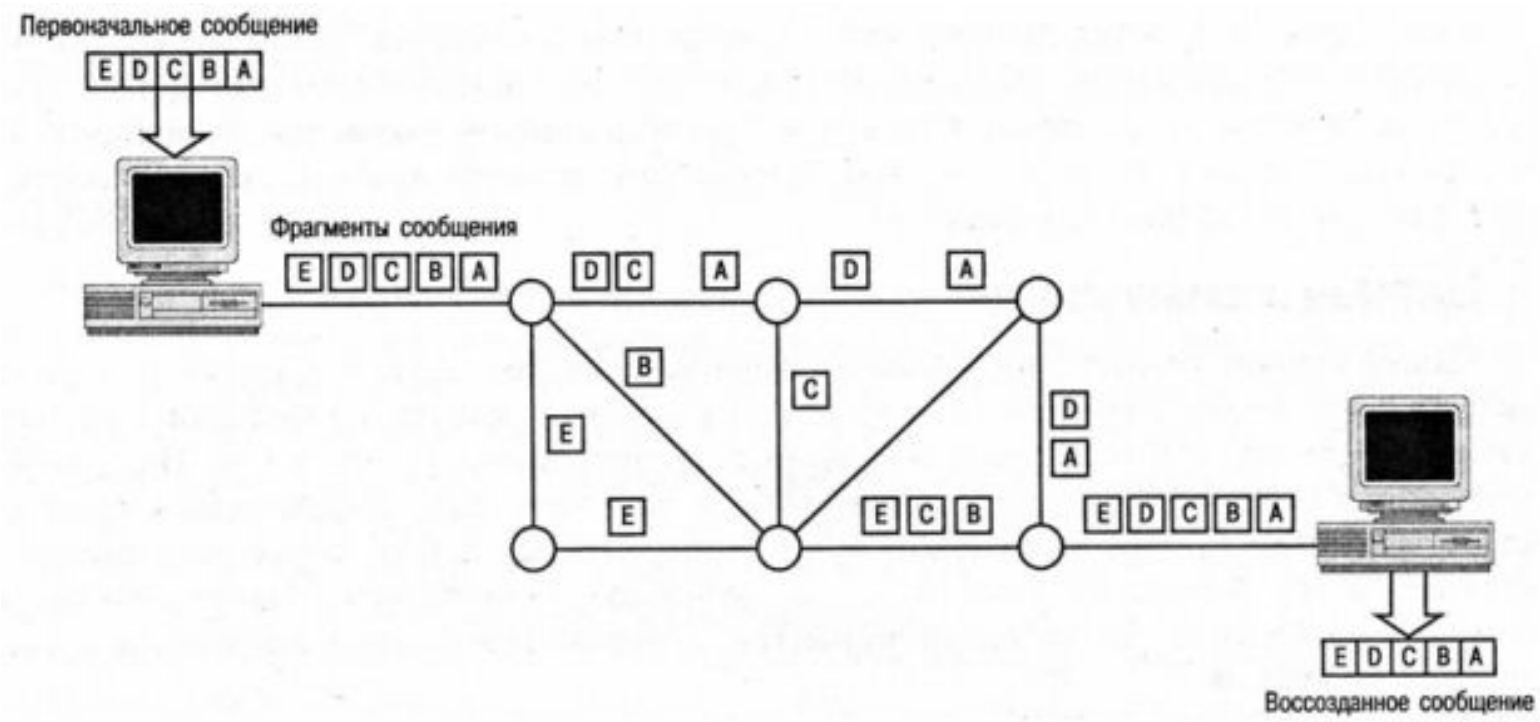
© 2009-2010 TATU

Целостность IP-пакетов



- IP вообще не следит за целостностью IP-пакетов
  - за целостностью данных «следят» протоколы канального и транспортного уровня, IP ни к чему дублировать их функциональность
  - единственная проверка, которую обеспечивает IP – проверка целостности собственной служебной информации (контрольная сумма заголовка пакета)

Пакетная коммутация



Протокол IP: адресация

Системы адресации

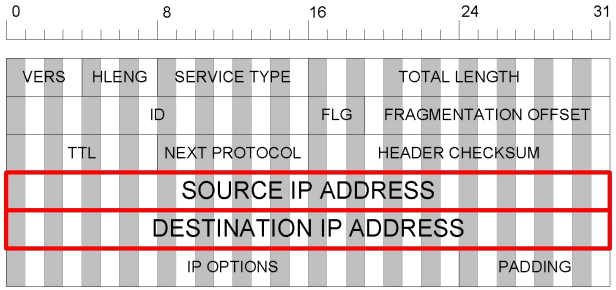
- Адресации:

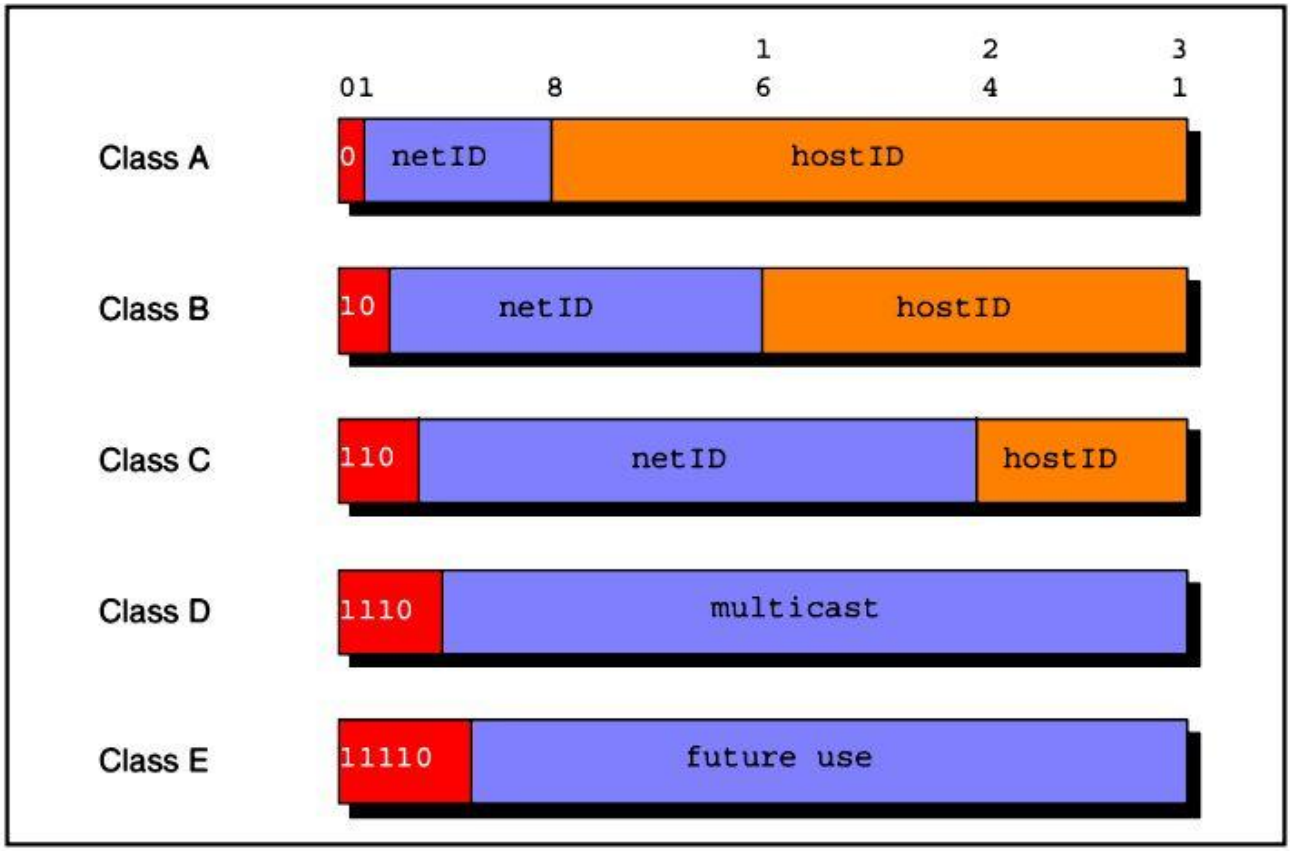
- 1. Прикладной уровень (служба DNS, имя компьютеров в рабочих группах Windows, др. системы символьной адресации), адресация в глобальных и локальных сетях
- 2. сетевой уровень (IP, IPX адреса), адресация в глобальных сетях
- 3. канальный уровень (MAC адрес), адресация в локальных сетях

- Службы:

- ARP/RARP (Address Resolution Protocol) – 2-3
- DNS (Domain Name Service) – 1-2
- Возможны службы для связи систем адресации 1-3

IP-адрес





- Уникальный в масштабах Интернет 32х-битный идентификатор, обычно в dotted-decimal notation
  - aaa.bbb.ccc.ddd
- Адресуемые объекты:
  - хосты (их сетевые интерфейсы)
  - сети
- Классификация сетей:
  A. до 16777214 хостов
  B. до 65534 хостов
  C. до 255 хостов
- Групповые (multicast) адреса:
  - Internet group management protocol (IGMP), RFC1112, 2236

Маска подсети

- Сеть (вернее, адресное пространство сети, определяемое полем netID) можно разделить на несколько частей (подсетей)
  - возможные причины для такого деления:
    - территориальное распределение адресного пространства
    - использование различных физических сред распространения данных в одной сети
    - необходимость логического деления пространства сети
  - инструмент деления: маска подсети
    - логические деление IP-адреса:
      [netID] [subnetID] [HostAddressSpace]
    - маска подсети: aaa.bbb.ccc.ddd, где битовое пространство [netID] и [subnetID] устанавливается в 1, а [HostAddressSpace] – в 0
  - маска подсети администрируется и используется локально в только в данной подсети

Специальные адресные пространства

00000000 00	00000000 00

00000000 00	xxxxxxxx xx

xxxxxxsx xx	00000000 00

11111111 11	11111111 11

xxxxxxxx xx	11111111 11

11111 10	xxxxxxxxxxx xx

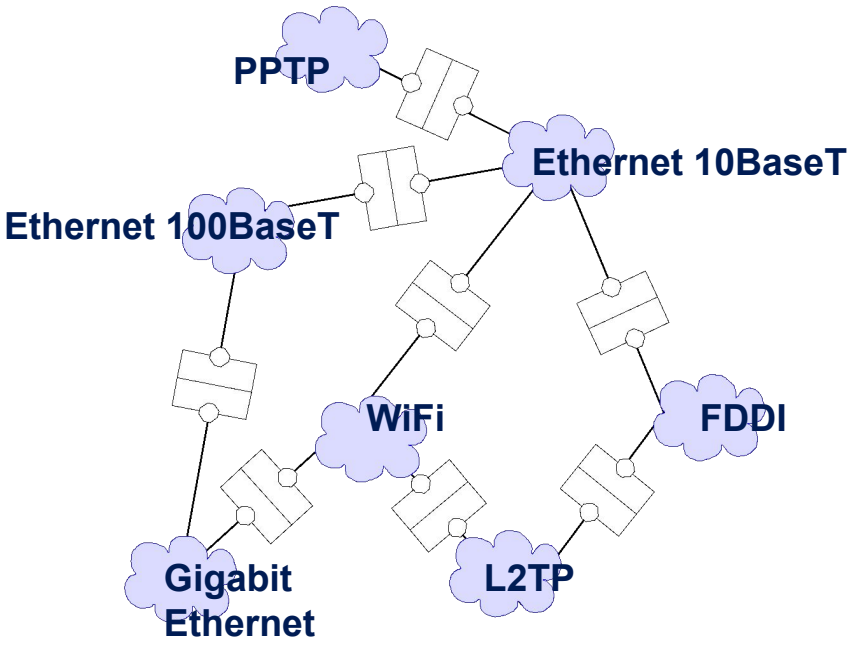
1. Адрес «этот хост», «пустышка»
   - может использоваться в инициализационной процедуре, когда рабочая станция не знает (или хочет согласовать) свой IP-адрес
   - этот адрес может использоваться только как адрес отправителя и никогда как адрес получателя пакета
2. NetID заполнен нулями, а hostID имеет осмысленное значение – есть адрес конкретного хоста в сети, из которой он получил пакет
   - это адрес используется только как адрес получателя и никогда как адрес отправителя
3. NetID имеет некоторое значение, а hostID заполнен нулями – адрес некоторой сети (но ни одного из хостов данной сети)
4. Limited или local broadcast address – полезный, предположительно, когда идентификатор сети по каким-либо причинам неизвестен
   - использование этого адреса не рекомендуется
5. Direct broadcast address, широковещательный адрес, обращенный ко всем хостам в данной подсети
6. Тестовый адрес (loopback address), в котором первый байт имеет значение 127, а прочее поле не специфицировано (обычно заполняетсяединицами)
   - используется для задач отладки и тестирования
   - не является адресом никакой сети и роутеры никогда не обрабатывают его

© 2009-2010 IATU

Частные адресные пространства

- Уникальное (регистрируемое) адресное пространство Интернет почти исчерпано; поэтому для пользования внутри собственных сетей (без выхода в Интернет) организации могут использовать т.н. частные (private), незарегистрированные адреса
  - RFC1918 рекомендует использовать для этих целей следующие адресные пространства:
    - сеть класса A – 10.0.0.0
    - 16 сетей класса B – от 172.16.0.0 до 172.31.0.0
    - 256 сетей класса C – от 192.168.0.0 до 192.168.255.0
  - когда хостам с этими адресами все-таки необходимо работать в Интернет – применяются технологии трансляции адресов (NAT, Network Address Translation)

Привязка и конфигурирование адресов



- Поскольку IP является независимым от природы (и внутренней адресации) физических подсетей, возникает задача сопоставления адресов физических подсетей (канального уровня) и IP-адресов (сетевого уровня)
  - эту задачу решают протоколы ARP и RARP
- В отдельных случаях требуется присвоить хосту IP-адрес
  - такое конфигурирование выполняют протоколы BOOTP и DHCP

Протокол ARP (RFC826)

- ARP (Address Resolution Protocol, протокол определения адресов)
  - сопоставляет 32-разрядные IP-адреса физическим адресам подсети, например, в 48-разрядные адреса Ethernet
- Идея протокола ARP:
  - если узлу A необходимо связаться с узлом B, узел A знает IP-адрес узла B, но не знает его физического адреса, узел A шлет широковещательное сообщение, в котором запрашивает физический адрес узла B
  - все узлы принимают это сообщение, однако только узел B отвечает на него, высылая в ответ свой физический адрес узлу A
  - узел A, получив физический адрес B, кэширует его, с тем, чтобы не запрашивать его повторно при следующих обращениях к узлу B

140.1.1.3
140.1.1.4
140.1.1.5
Фрейм запроса ARP (широковещательная трансляция)
Фрейм ответа ARP хосту 140.1.1.3

Протокол RARP (RFC1293)

- RARP (Reverse Address Resolution Protocol, протокол обратного определения адресов)
  - сопоставляет IP адрес физическому
  - применяется, если узел А из предыдущего примера «не знает» собственного IP-адреса
- Идея протокола RARP:
  - узел А широковещательно вызывает RARP-сервер, закладывая в запрос свой физический адрес
  - RARP-сервер распознает запрос узла А, выбирает из некоторого списка свободный IP-адрес и шлет узлу А сообщение, включающее: динамически выделенный узлу А IP-адрес, физический и IP-адрес RARP-сервера
  - *отказ RARP-сервера становится очень критичен, поэтому применяется резервирование RARP-серверов*

© 2009-2010

Протоколы BOOTP (RFC951), DHCP (RFC2131, 2132)

- Результатом работы протоколов BOOTP (Bootstrap) и DHCP (Dynamic host configuration protocol) является конфигурирование IP-адресов, но применение этих протоколов – шире, и они не являются, строго говоря, протоколами сетевого уровня
- Протокол BOOTP обеспечивает начальную загрузку бездисковых рабочих станций, сетевых принтеров и т.п.
- Протокол DHCP базируется на BOOTP, но расширяет его возможности в двух отношениях:
  1. DHCP может выдавать IP адрес «во временной пользование» на ограниченное время; эта функция важна для эффективного использования адресного пространства, когда в сети появляются и исчезают некоторые хосты
  2. DHCP снабжает конфигурируемый хост не только IP-адресом, но и полным набором параметров стека (включая наборы параметров канального, сетевого и транспортного уровня)

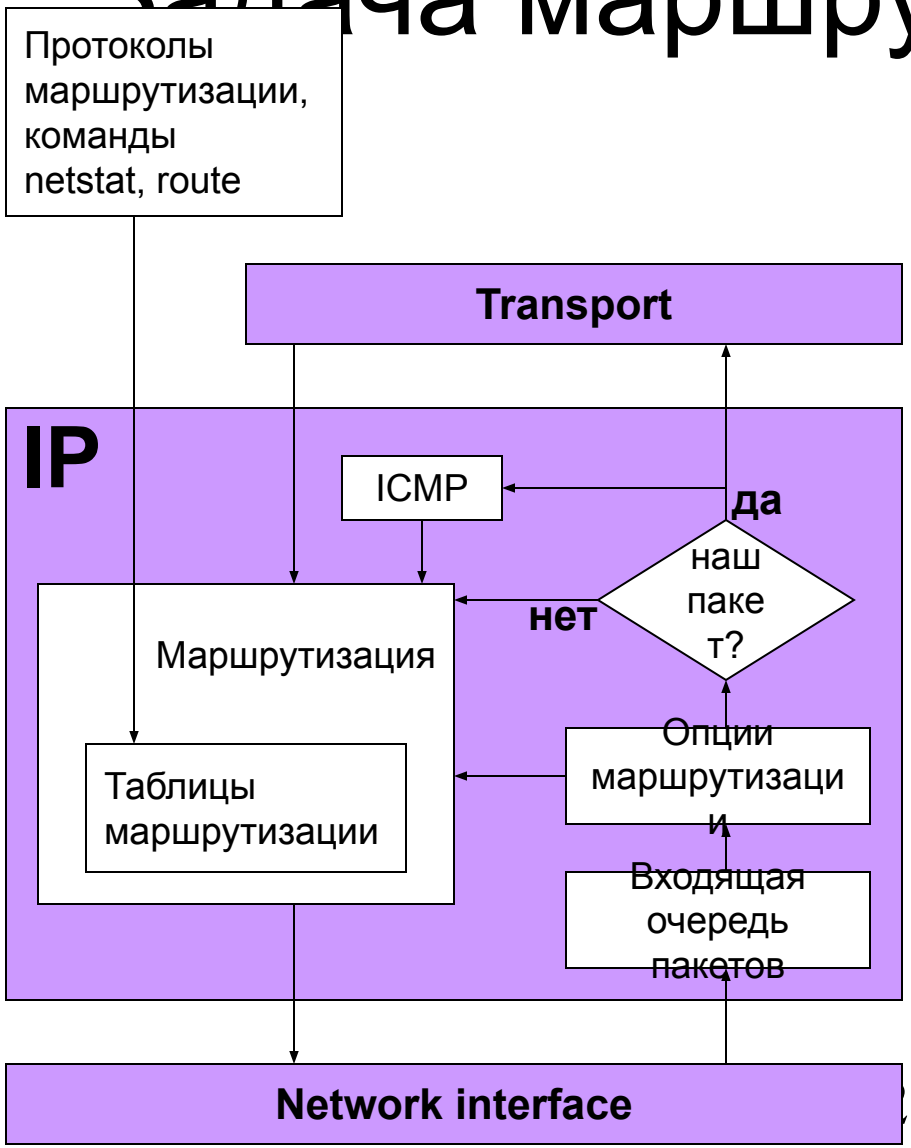
Справка. Параметры DHCP-настройки стека

- Параметры протокола IP
  - на уровне хоста
    - Be a router (on/off)
    - Non-local source routing (on/off)
    - Policy filters for non-local source routing (list)
    - Maximum reassembly size
    - Default TTL
    - PMTU aging timeout
    - MTU plateau table
  - на уровне интерфейса
    - IP address
    - Subnet mask
    - MTU
    - All-subnets-MTU (on/off)
    - Broadcast address flavor (0x00000000/0xffffffff)
    - Perform mask discovery (on/off)
    - Be a mask supplier (on/off)
    - Perform router discovery (on/off)
    - Router solicitation address
    - Default routers, list of:
      - router address
      - preference level
    - Static routes, list of:
      - destination (host/subnet/net)
      - destination mask (address mask)
      - type-of-service
      - first-hop router
      - ignore redirects (on/off)
      - PMTU
      - perform PMTU discovery (on/off)

- Параметры канального уровня (поинтерфейсно):
  - Trailers (on/off)
  - ARP cache timeout
  - Ethernet encapsulation
- Параметры протокола TCP:
  - TTL
  - Keep-alive interval
  - Keep-alive data size

Протокол IP: маршрутизация

Задача маршрутизации

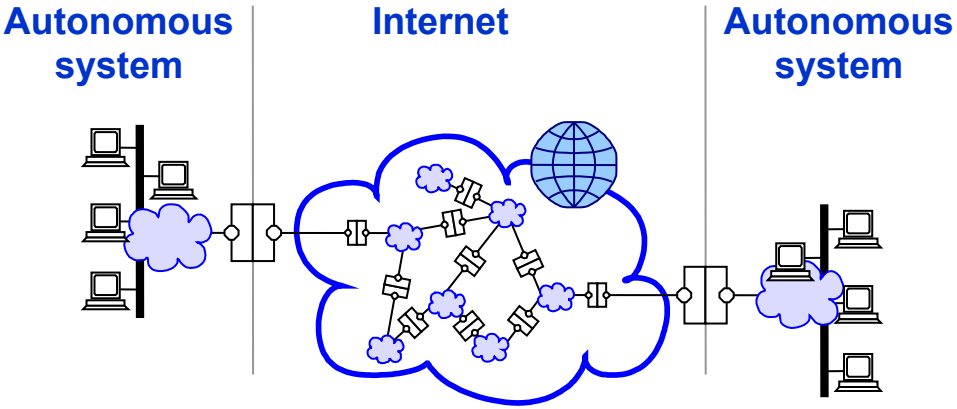


- Маршрутизационная задача решается локально на каждом хосте/маршрутизаторе
  - постановка задачи: на какой сетевой интерфейс нужно передать пакет, чтобы он дошел до получателя?
- Маршрутизационные таблицы (routing table) указывают, куда нужно перенаправлять пакет с заданным адресом
  - отдать некоторому хосту (прямая маршрутизация)
  - некоторому маршрутизатору (непрямая маршрутизация),
  - или передать в некоторую подсеть

Статическая и динамическая маршрутизация

- *Статическая маршрутизация* основывается на жестко заданных маршрутизационных таблицах
  - наиболее приемлема для хостов, работающих в небольших сетях, поскольку хост не должен тратить много сил на задачи маршрутизации
  - часто маршрутную таблицу на хосте конфигурируют как небольшой список хостов-соседей и маршрутизатор по умолчанию (default gateway), которому хост и отдает все пакеты, маршрутизация которых у него не определена
  - логика default gateway работает и на уровне маршрутизаторов: маршрутизатор обязан знать все о своей локальной сети, однако информацией о внешних сетях он может и не владеть, обращаясь, при необходимости, к владеющим этой информацией внешним маршрутизаторам
- *Динамическая маршрутизация* – позволяет хосту (маршрутизатору), взаимодействуя со смежными узлами по *протоколам маршрутизации*, обновлять и корректировать информацию в маршрутных таблицах

Архитектура маршрутизации Интернет

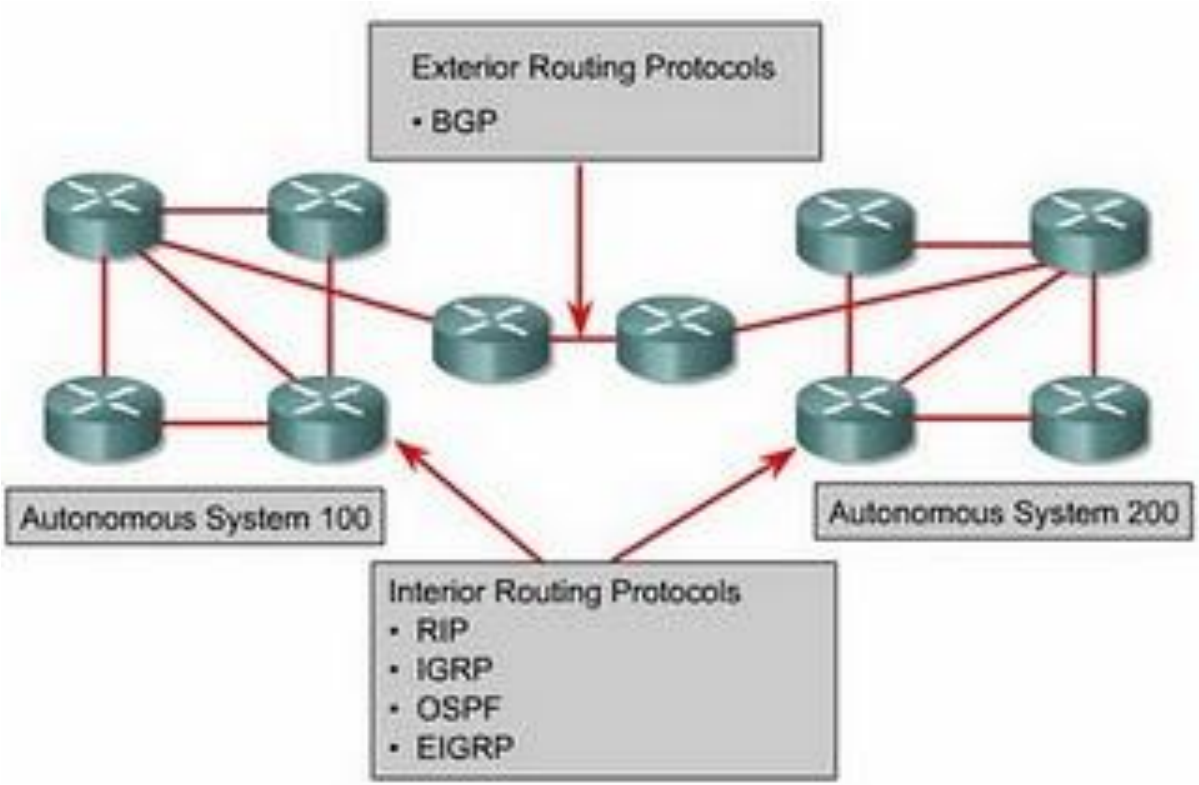


- Архитектурно задача динамической маршрутизации в Интернет делится на два уровня:
  - маршрутизацию внутри локальных (или целостных многосегментных ведомственных сетей) – такие системы называют автономными системами, (autonomous system)
  - межсетевую маршрутизацию
- Протоколы маршрутизации, осуществляющие межсетевую маршрутизацию называют *exterior routing protocols*, маршрутизацию внутри автономных систем осуществляют *interior routing protocols*

Interior routing protocols

- Routing Information Protocol (RIP), Xerox Corp., начало 1980х
  - прост (минимизирует путь только по числу хопоов), ограничен (максимальная длина пути – 16 хопов) получил широкое распространение в малых сетях
- Interior Gateway Routing Protocol (IGRP), Cisco Systems, нач. 1980х
  - определяет путь с учетом скорости линий и суммарной задержки
  - развитие – Enhanced IGRP, более эффективен, использован для маршрутизации не только IP (IPX, AppleTalk)
- Open Shortest Path First (OSPF), IETF
  - развитой междоменный иерархический (делит автономные системы на магистраль и подсети) протокол, разработанный взамен RIP
  - гибок, эффективен, поддерживает маски сетей переменной длины
- Integrated intermediate System to Intermediate System protocol (IS-IS), ISO 10589
  - междоменный иерархический протокол маршрутизации, похож на OSPF
  - работает через множество LAN- и WAN-подсетей, двухточечные соединения, поддерживает протоколы OSI

Пример использования различных протоколов в сети



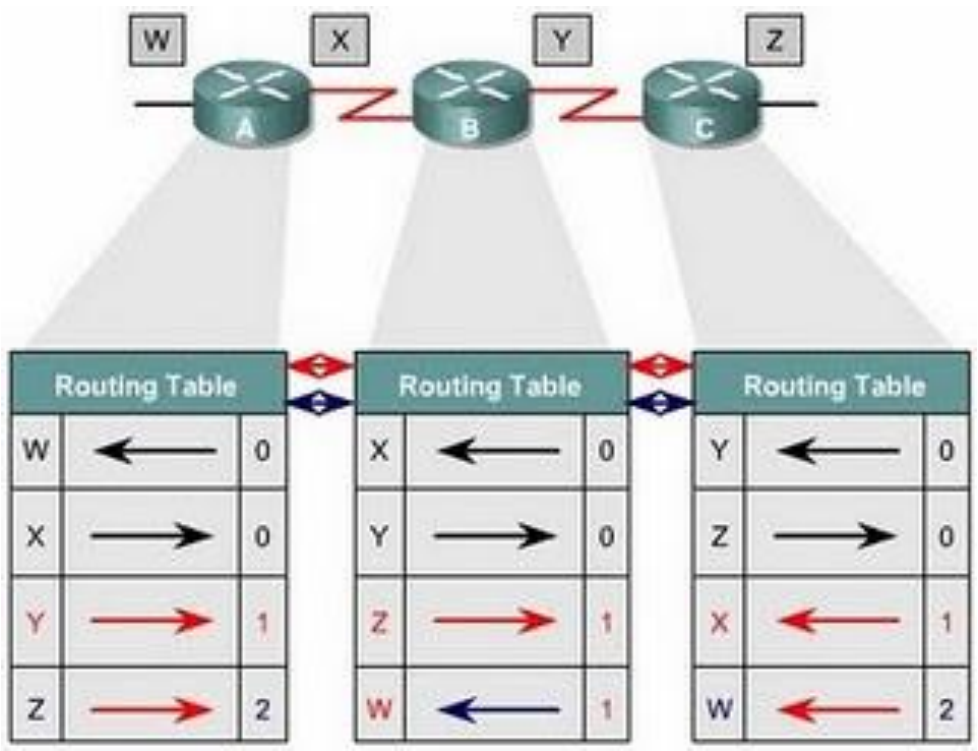
Exterior routing protocols

- Exterior Gateway Protocol, EGP
  - обеспечивает динамическую маршрутизацию, очень прост, ограничен, исходит из предположения, что автономные системы подключены к древесной топологии, не использует метрик
- Border Gateway Protocol, BGP
  - работает в произвольных топологиях, исключает циклы, использует метрики, высоко масштабируем

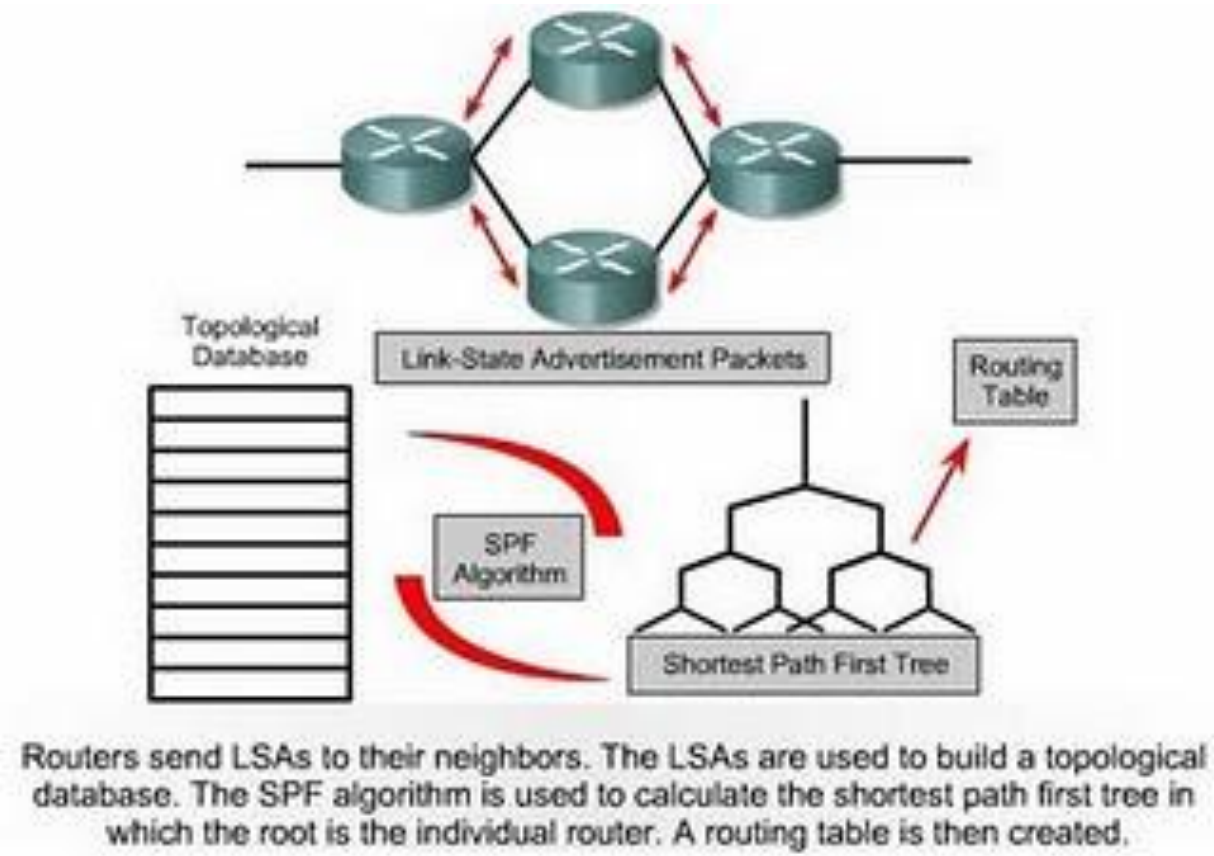
Классификация маршрутизируемых протоколов

- Многие маршрутизируемых протоколов можно разделить на следующие категории:
  - Distance vector
  - Link-state

Distance vector routing protocol



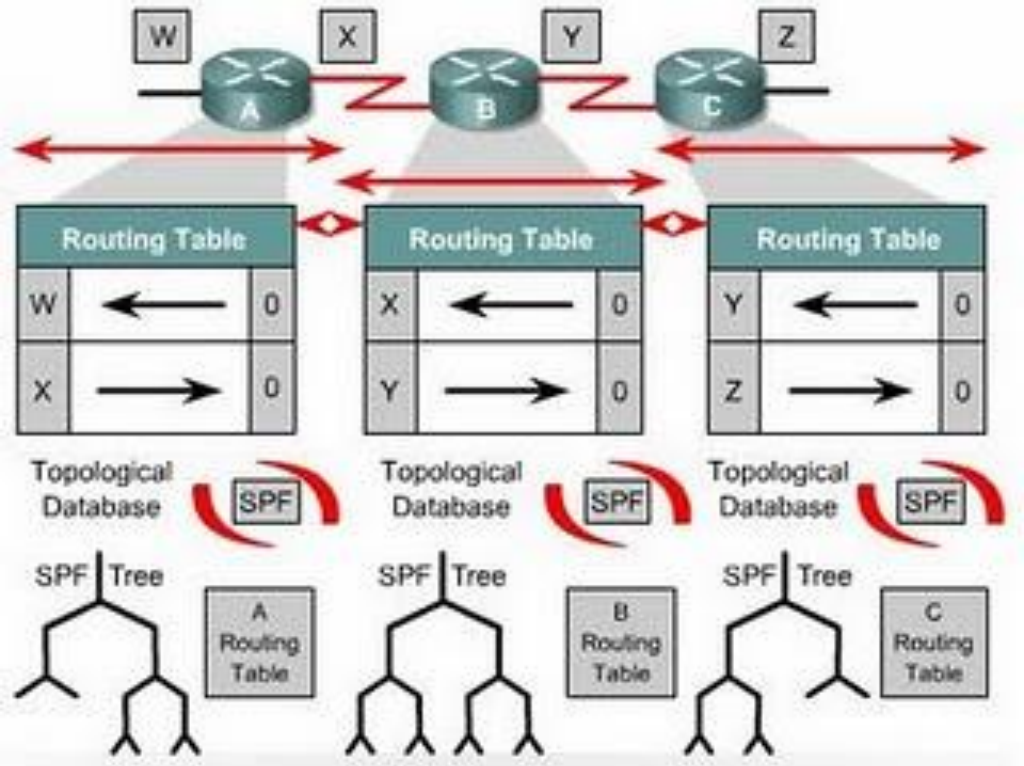
Алгоритм маршрутизации Link-state



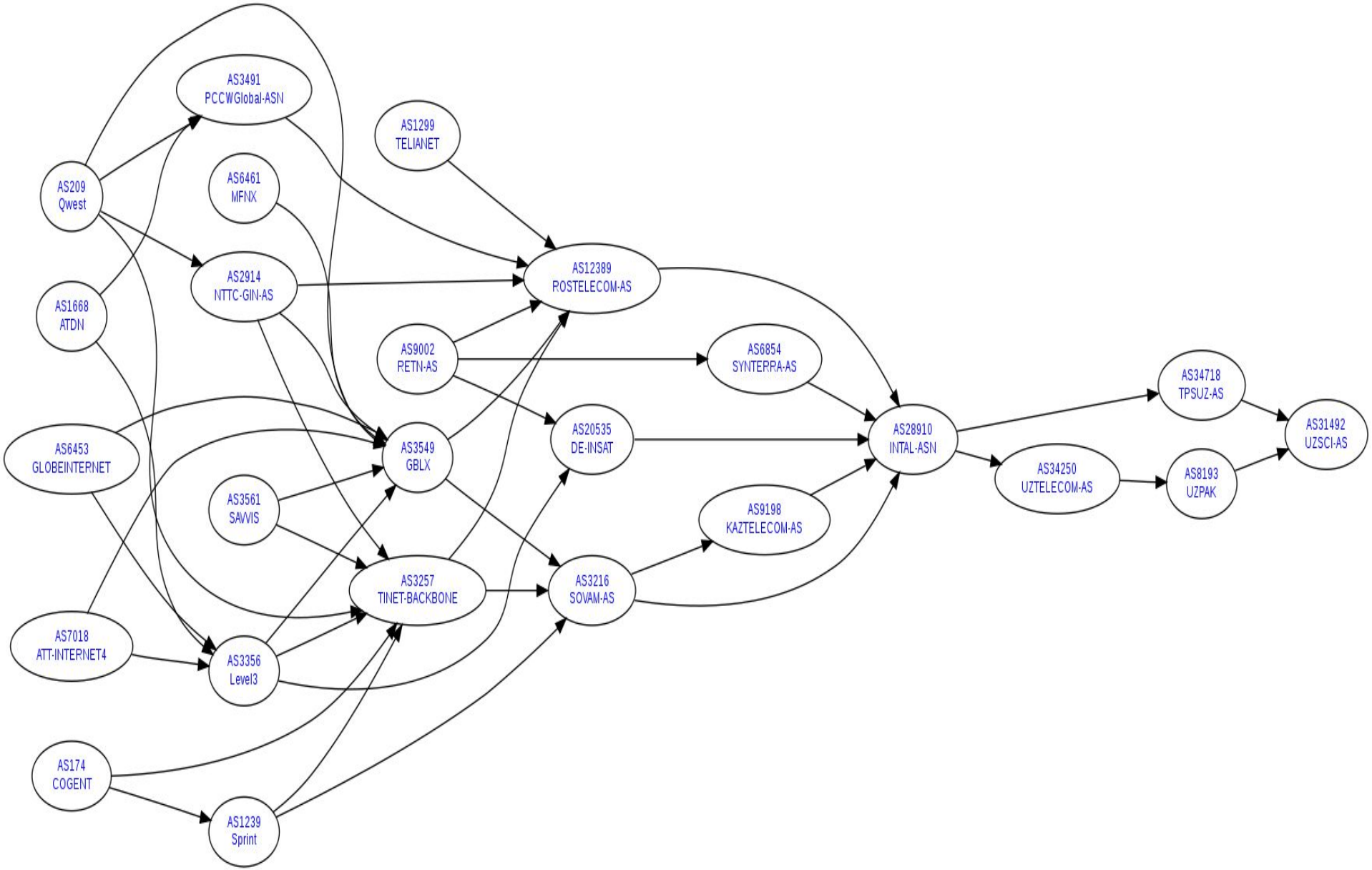
Используется Алгоритм Дейкстры

shortest path first (SPF).

Link-state



Each router has its own topological database on which the SPF algorithm is run.


AS3491
PCCWGlobal-ASN
AS1299
TELIANET
AS209
Qwest
AS6461
MFNX
AS2914
NTTC-GIN-AS
AS12389
ROSTELECOM-AS
AS1668
ATDN
AS9002
RETN-AS
AS6854
SYNTERRA-AS
AS34718
TPSUZ-AS
AS6453
GLOBEINTERNET
AS3549
GBLX
AS20535
DE-INSAT
AS28910
INTAL-ASN
AS31492
UZSCI-AS
AS3561
SAVVIS
AS34250
UZTELECOM-AS
AS8193
UZPAK
AS9198
KAZTELECOM-AS
AS3257
TINET-BACKBONE
AS3216
SOVAM-AS
AS7018
ATT-INTERNET4
AS3356
Level3
AS174
COGENT
AS1239
Sprint

Вывод

- **Conclusion**

- - Routing is the process of how routers pass the packets to the destination network

- - Routing protocols are used between the communication routers

- - Routing protocol allows a router to share information with other routers on the network and he knows the best path to the network

- - Routing algorithm can be classified as one of two categories, distance vector or link-state

- - Autonomous systems (AS) is a collection of networks within a single administrative control

- RIP - using interior routing protocols with distance vector algorithms
  - IGRP - interior routing protocol with Cisco distance vector algorithm
  - OSPF - interior routing protocol with link-state algorithm
  - EIGRP - interior routing protocol with Cisco advanced algorithm distance vector
  - BGP - using exterior routing protocols with distance vector algorithm

  Basic RIP described in RFC 1058, with the following characteristics:
  - Distance vector routing protocol
  - Metric based on the number of jumps (hop count) for route selection
  - If the hop count for more than 15, packet discarded
  - Update routing is done in a broadcast every 30 seconds

  IGRP is a routing protocol developed by Cisco, with the following characteristics:
  - Distance vector routing protocol
  - Using a composite metric consisting of bandwidth, load, delay and reliability
  - Update routing is done in a broadcast every 90 seconds

- Routing protocol OSPF uses link-state, with the following characteristics:
  - Protocol link-state routing
  - It is open standard routing protocol described in RFC 2328
  - Using the SPF algorithm to calculate the lowest cost
  - Update routing floaded done when the network topology changes

  EIGRP protocol enhanced distance vector routing, with the following characteristics:
  - Using the protocol enhanced distance vector routing
  - Using the cost load balancing is not the same
  - Using a combination algorithm distance vector and link-state
  - Using the Diffusing Update Algorithm (DUAL) to calculate the shortest path
  - Update routing is done using multicast address 224.0.0.10 caused by changes in network topology

- Border Gateway Protocol (BGP) is an exterior routing protocol, with the following characteristics:
  - Use distance vector routing protocol
  - Used between the ISP and the ISP clients
  - Used to route internet traffic between autonomous systems

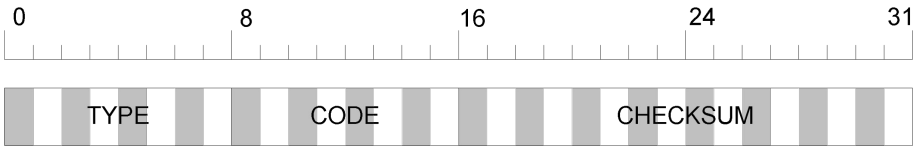
© 2009-2010

Протокол IP: служебные обмены и диагностика

Протокол ICMP

- Internet Control Message Protocol (ICMP) используется в случаях доставки информации об ошибках, причем
  - ICMP, как и IP, – работает с негарантированной доставкой, сообщения об ошибках могут теряться, как всякие IP-пакеты
  - сообщения об ошибках в протоколе ICMP не выдаются с тем, чтобы избежать генерации бесконечных повторов
  - в случае ошибок фрагментации ICMP-сообщение генерируется только для первого фрагмента
  - уведомление направляется обычно отправителю, ICMP-сообщения никогда не высылаются в ответ на групповые сообщения и в случаях, когда адрес отправителя не определяет хост-источник однозначно

Формат ICMP-сообщения



- ICMP-сообщение инкапсулируется в обычный IP-пакет, снабжается стандартным заголовком
  - TYPE – определяет тип сообщения (ICMP-приложения)
  - CODE – содержит код ошибки код ошибки (специфичный для ICMP-приложения)
  - CHECKSUM – содержит контрольную сумму по ICMP-сообщению, начиная с поля тип и включая данные, которые могут следовать за заголовком сообщения

- **Поддерживаемые типы сообщений:**

0: Echo reply
3: Destination unreachable
4: Source quench
5: Redirect
8: Echo
9: Router advertisement
10: Router solicitation
11: Time exceeded
12: Parameter problem
13: Timestamp request
14: Timestamp reply
15: Information request (устарел)
16: Information reply (устарел)
17: Address mask request
18: Address mask reply
30: Traceroute
31: Datagram conversion error
32: Mobile host redirect
33: IPv6 Where-Are-You
34: IPv6 I-Am-Here
35: Mobile registration request
36: Mobile registration reply
37: Domain name request
38: Domain name reply
39: SKIP
40: Photuris

Важнейшие ICMP-приложения

- Ping – диагностика связи с хостом

```
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] destination-list

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet.
    -i TTL         Time To Live.
    -v TOS         Type Of Service.
    -r count       Record route for count hops.
    -s count       Timestamp for count hops.
    -j host-list   Loose source route along host-list.
    -k host-list   Strict source route along host-list.
    -w timeout     Timeout in milliseconds to wait for each reply.
```

- Traceroute – трассировка пути к удаленному хосту

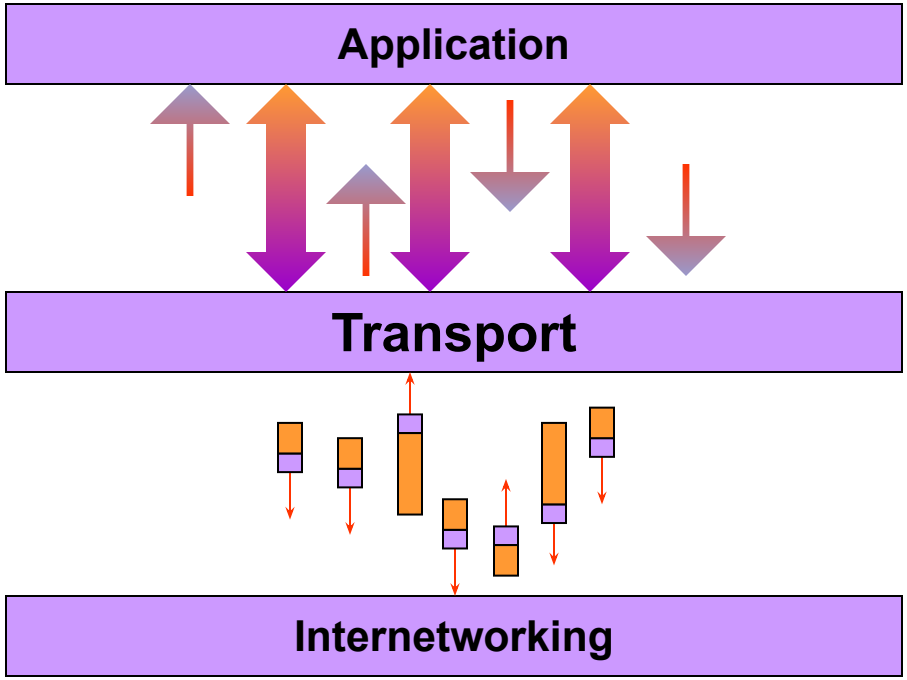
```
Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list.
    -w timeout         Wait timeout milliseconds for each reply.
```

© 2009-2010 TATU

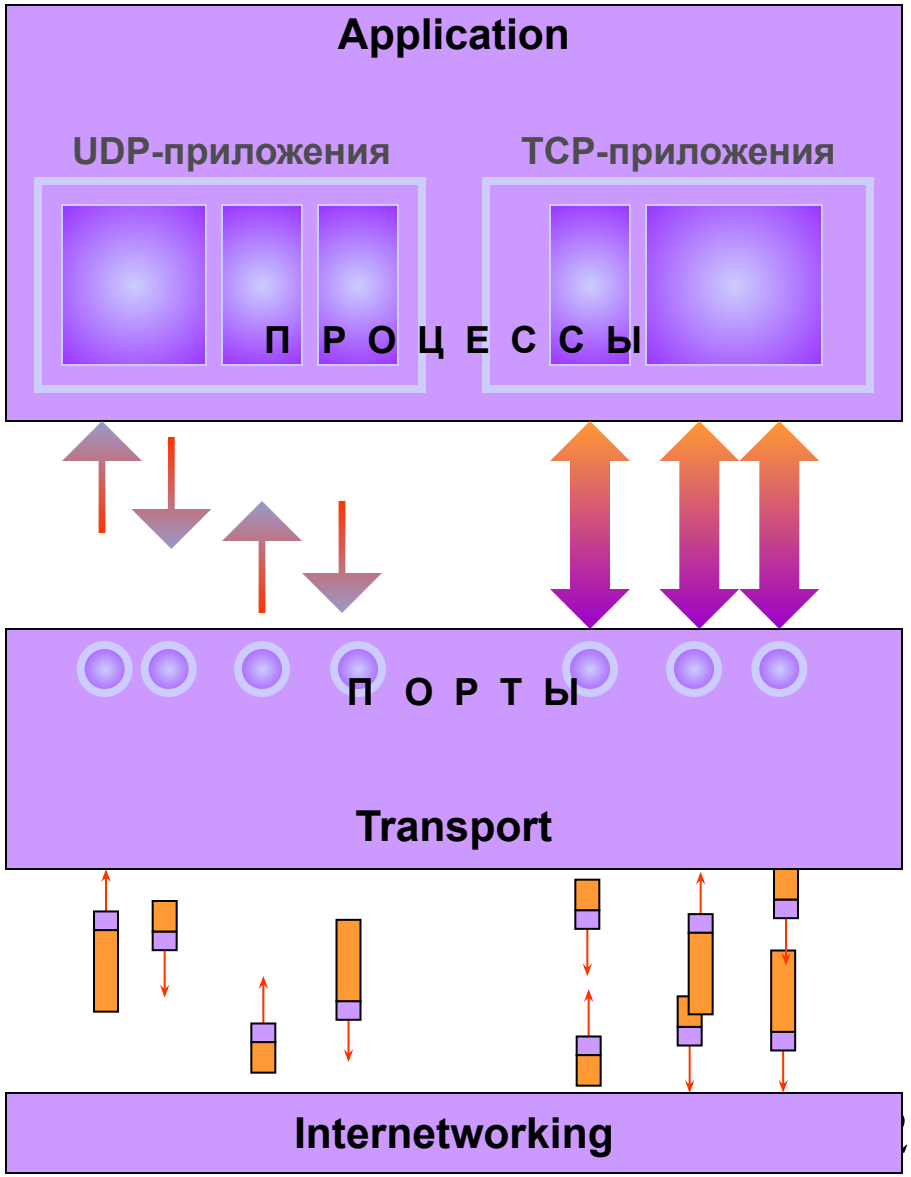
- Транспортные протоколы: сервис

Сервис транспортного уровня



- Приложения не формируют IP-пакеты
  - для этого разработчикам приложений пришлось бы разбираться с неспецифичными для них сетевыми задачами и приложения стали бы зависимы от типа сети
- Транспортный уровень принимает потоки данных или сообщения, «упаковывает» данные приложений в IP-пакеты и передает в сеть
  - сервис негарантированной доставки единичных сообщений обеспечивает транспортный протокол UDP
  - потоковый транспортный сервис с надежной доставкой обеспечивает протокол TCP

Идентификация приложений



- Транспортный уровень принимает из сети пакеты для множества приложений, возникает проблема разобраться – где чьи данные
- Сетевые приложения идентифицируются 16-разрядным числом – портом (port)
  - одно приложение может использовать несколько портов
  - сетевое соединение (между приложениями) однозначно определяется набором параметров:

(T-protocol, SRC-IP, SRC-port, DST-IP, DST-port)

Немного о портах

- Порты транспортных протоколов бывают предписанные (well-known) и динамически назначаемые
  - номера предписанных портов лежат в диапазоне от 1 до 1023
    - распределением (предписанием) well-known портов занимается специальная организационная структура Интернет – IANA
    - well-known порты приписываются серверам известных (широко распространенных) приложений; клиенты обычно используют эфемеридные, динамически назначаемые порты
  - динамически назначаемые порты используются либо известными приложениями для установления временных соединений, либо нераспространенными приложениями
    - когда нераспространенному приложению необходимо получить номер порта, либо когда возникает конфликт использования номеров портов (например, когда на сервере работают два однотипных известных приложения), приложения запрашивают динамический номер порта у TCP/IP-стека

© 2009-2010 TATU

Прикладной интерфейс socket

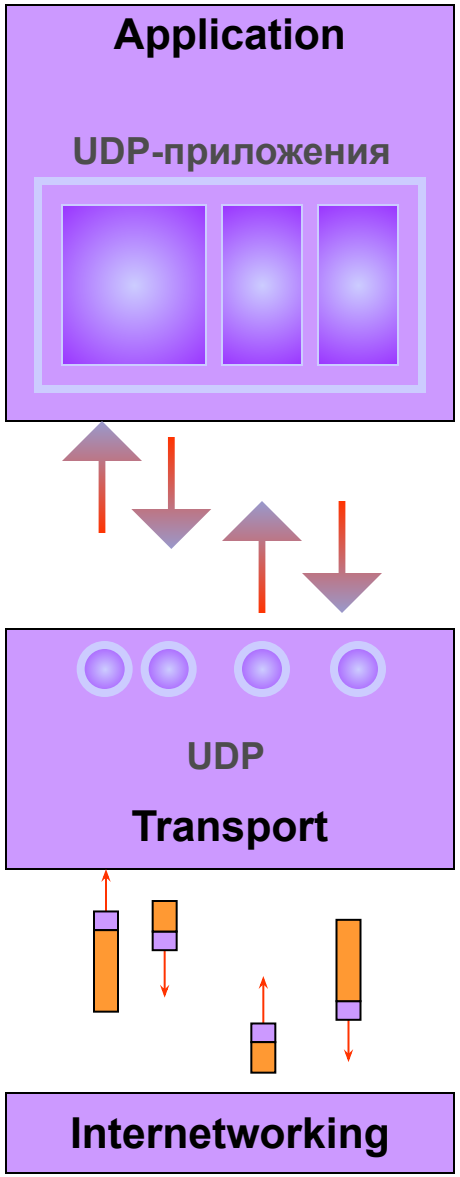
- Наиболее распространенным прикладным интерфейсом для передачи данных по сети является предложенный в BSD Unix интерфейс socket
  - socket описывает сетевое соединение, как файл ввода-вывода
  - окончание соединения socket идентифицируется триадой

    (T-protocol, IP-address, process port),

    которую также называют *транспортным адресом* (socket или transport address)

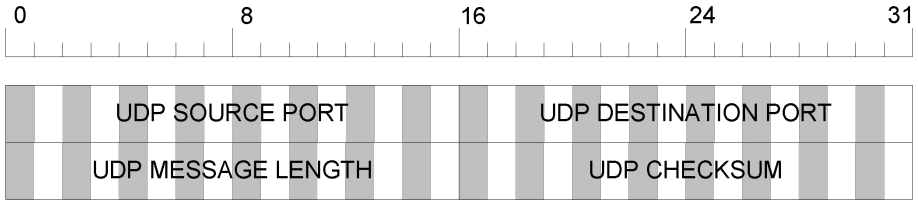
Транспортные протоколы: UDP

Сервис протокола UDP



- User Datagram Protocol, UDP (RFC 768) обеспечивает обмен единичными сообщениями между приложениями
  - UDP очень прост, это прямая ретрансляция сервиса протокола IP приложениям
  - UDP - дейтаграммный протокол, не гарантирующий доставку (может как терять, так и дуплицировать сообщения) и не сохраняющий порядка следования сообщений
- Сообщение протокола UDP называют *пользовательской дейтаграммой* (user datagram)

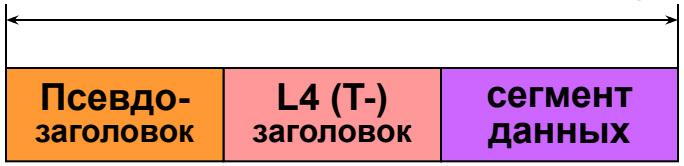
Дейтаграмма UDP



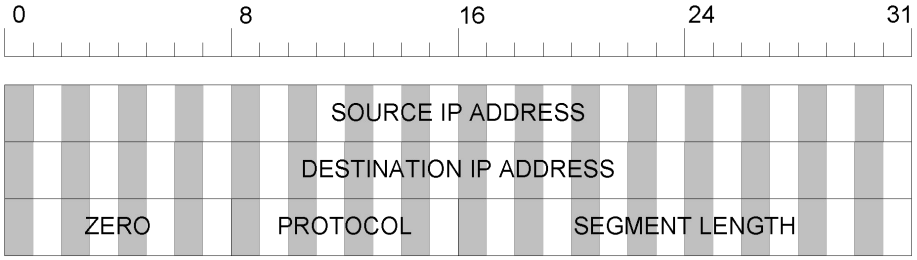
- UDP SOURCE PORT и UDP DESTINATION PORT – порты процесса-отправителя и процесса-получателя
  - source port имеет ненулевое заполнение, если процесс-отправитель должен получить ответное сообщение
- UDP MESSAGE LENGTH – полная длина заголовка и сегмента данных
- UDP CHECKSUM – контрольная сумма

Контрольная сумма Т-протоколов

Пространство расчета контрольной суммы



Структура псевдозаголовка

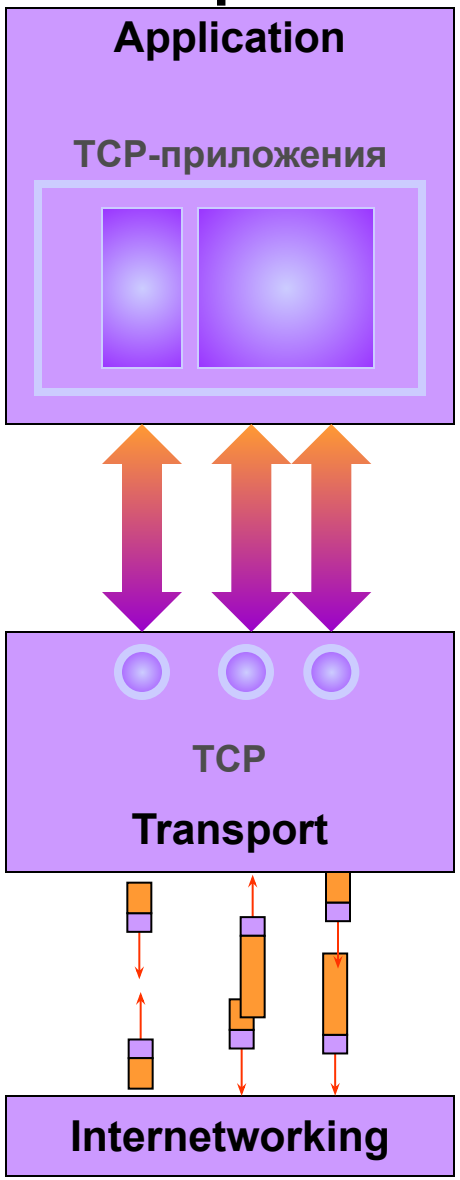


- Расчет контрольной суммы в TCP обязателен, а в UDP опционален: заполнение поля CHECKSUM нулями означает в UDP отказ от расчета контрольной суммы
  - при отказе от расчета контрольной суммы в UDP следует иметь ввиду, что сохранность блока данных не гарантирована ничем, кроме канального протокола
- Расчет контрольной суммы производится по трем структурам данных:
  - псевдозаголовку
  - транспортному заголовку
  - сегменту данных транспортного сообщения

Транспортные протоколы: TCP

ВВЕДЕНИЕ В ТЕХНОЛОГИИ TCP/IP

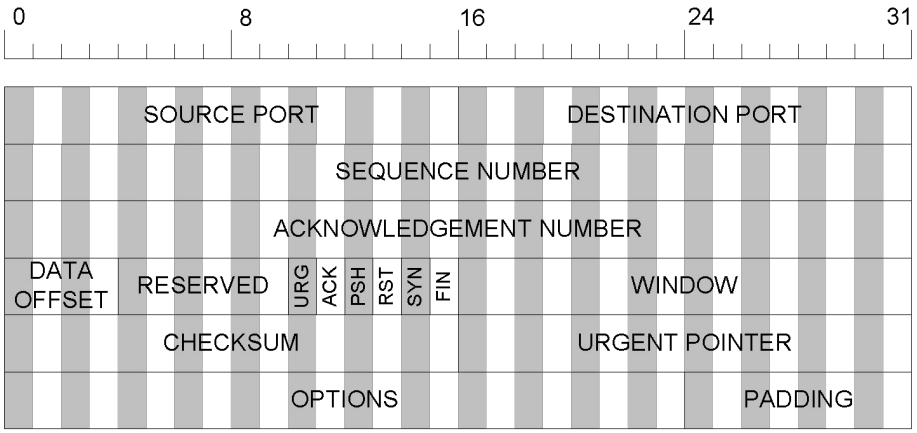
Сервис протокола TCP



- Transfer Control Protocol, TCP (RFC793)
  - обеспечивает транспорт потоков (stream) т.е. приложение, передающее данные, не заботится о том, чтобы передавать транспортному протоколу информацию порциями
  - обрабатывает неструктурированные потоки данных, т.е. не накладывает никаких ограничений на состав потока и взаимосвязи между его элементами
  - буферизует данные, передаваемые в сеть
  - организует т.н. виртуальные соединения посредством предварительной согласовательной процедуры
  - обеспечивает полнодуплексное соединение при этом обеспечивается
    - управление потоком (в зависимости от пропускной способности и загрузки сети)
  - обеспечивает целостность потока и гарантирует доставку данных

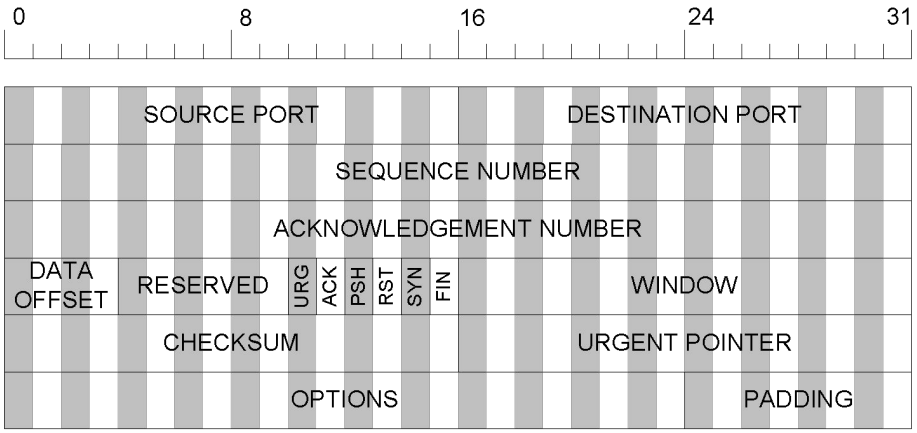
© 2009-2010 TATU

Сегмент TCP



- TCP передает данные порциями (сегментами), каждый из которых включается затем в IP пакет
- Заголовок сегмента (транспортный заголовок TCP) обеспечивает возможность для передачи управляющей информации протокола вместе с трафиком (piggybacking)

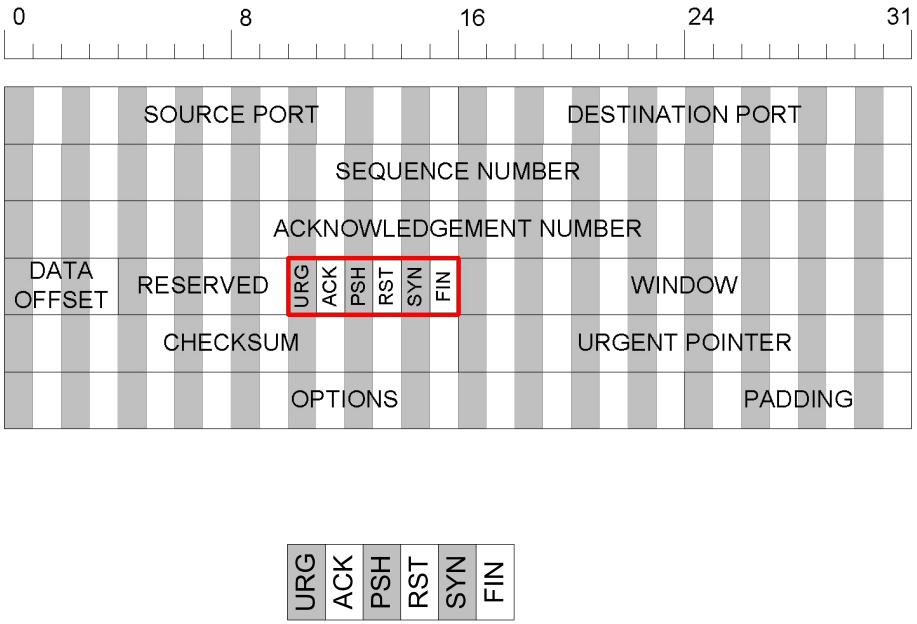
Поля заголовка TCP



- SOURCE PORT, DESTINATION PORT – номера портов отправителя и получателя сообщения
- SEQUENCE, ACKNOLEDGEMENT NUMBER, WINDOW, URGENT POINTER – поля для управления потоком
- DATA OFFSET – указатель на конец заголовка (начало блока данных)
- CHECKSUM – контрольная сумма по сегменту данных
- OPTIONS – варианты
- PADDING - заполнение

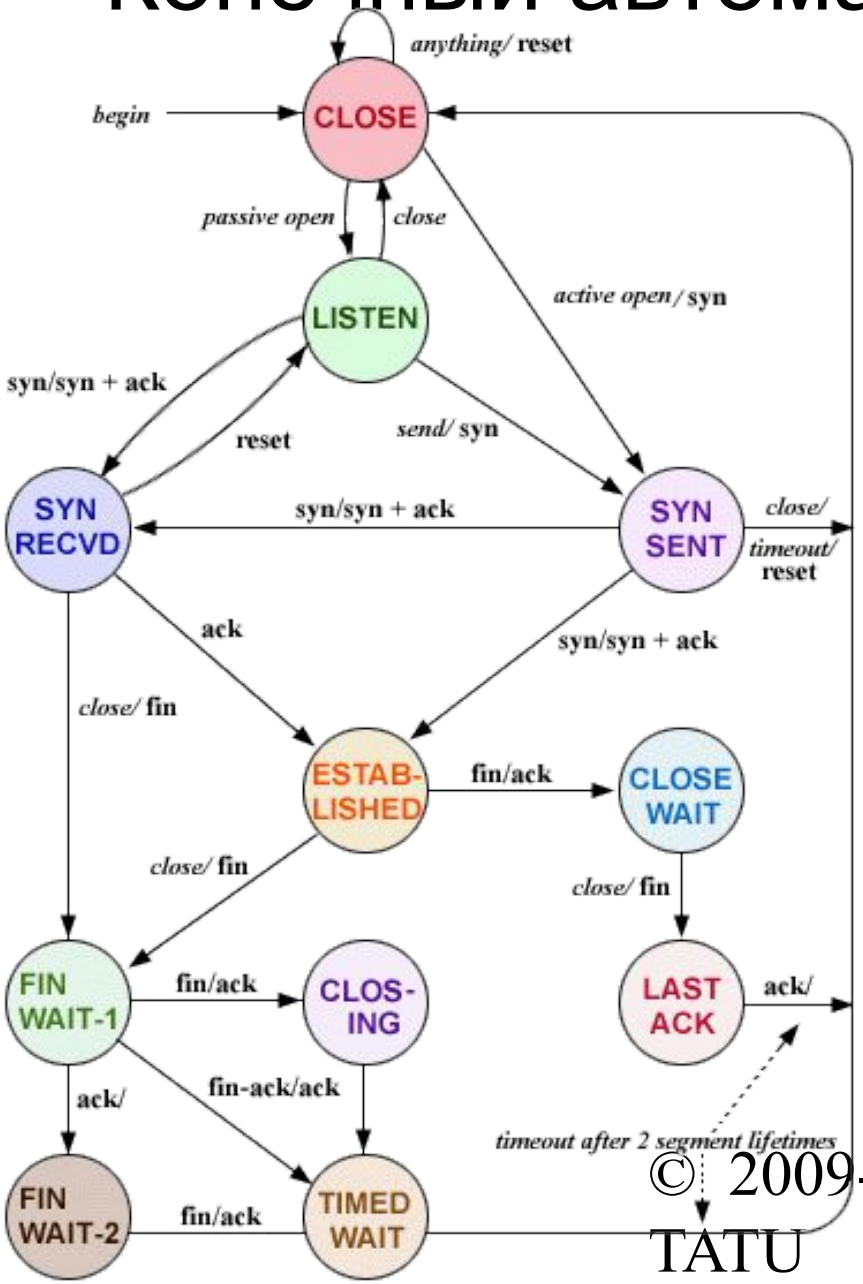
© 2009-2010 TATU

Биты управления



- URG, urgent – срочная передача данных
- ACK, acknowledgement – подтверждение приема
- PSH, push – очистка буфера
- RST, reset – переустановление соединения
- SYN, synchronize – синхронизация потоков
- FIN, finish – окончание потока данных

Конечный автомат протокола TCP



- Работу протокола TCP удобно пояснить на основе конечного автомата; состояния:
  - CLOSE – холостое состояние, отсутствие соединения
  - LISTEN, SYN RECVD, SYN SENT – промежуточные состояния фазы установления соединения
  - ESTABLISHED – соединение установлено, передача данных
  - CLOSE WAIT, LAST ACK, FIN WAIT 1,2, CLOSING, TIMED WAIT – промежуточные состояния фазы завершения соединения

© 2009-2010 TATU

Установление TCP-соединения

Инициатор

Запрос от ОС на установление соединения

Active open:

1. Выделение порта
2. Инициализация счетчика выходного потока (SQNC=X), где X – случайное число
3. Отправка сегмента SYN

Прием SYN/ACK:

1. Выходной счетчик = Y
2. Квитирование (ACK SQNC=Y+1)

CLOSE

SYN SENT

ESTABLISHED

SYN=1, ACK=0, SQNC=X

SYN=1, ACK=1, SQNC=Y ACK SQNC=X+1

ACK=1, ACK SQNC=Y+1

Адресат

CLOSE

LISTEN

SYN RECVD

ESTABLISHED

Запрос от ОС на прием

Passive open: инициализация серверного порта, переход в состояние ожидания

Прием сообщения SYN:

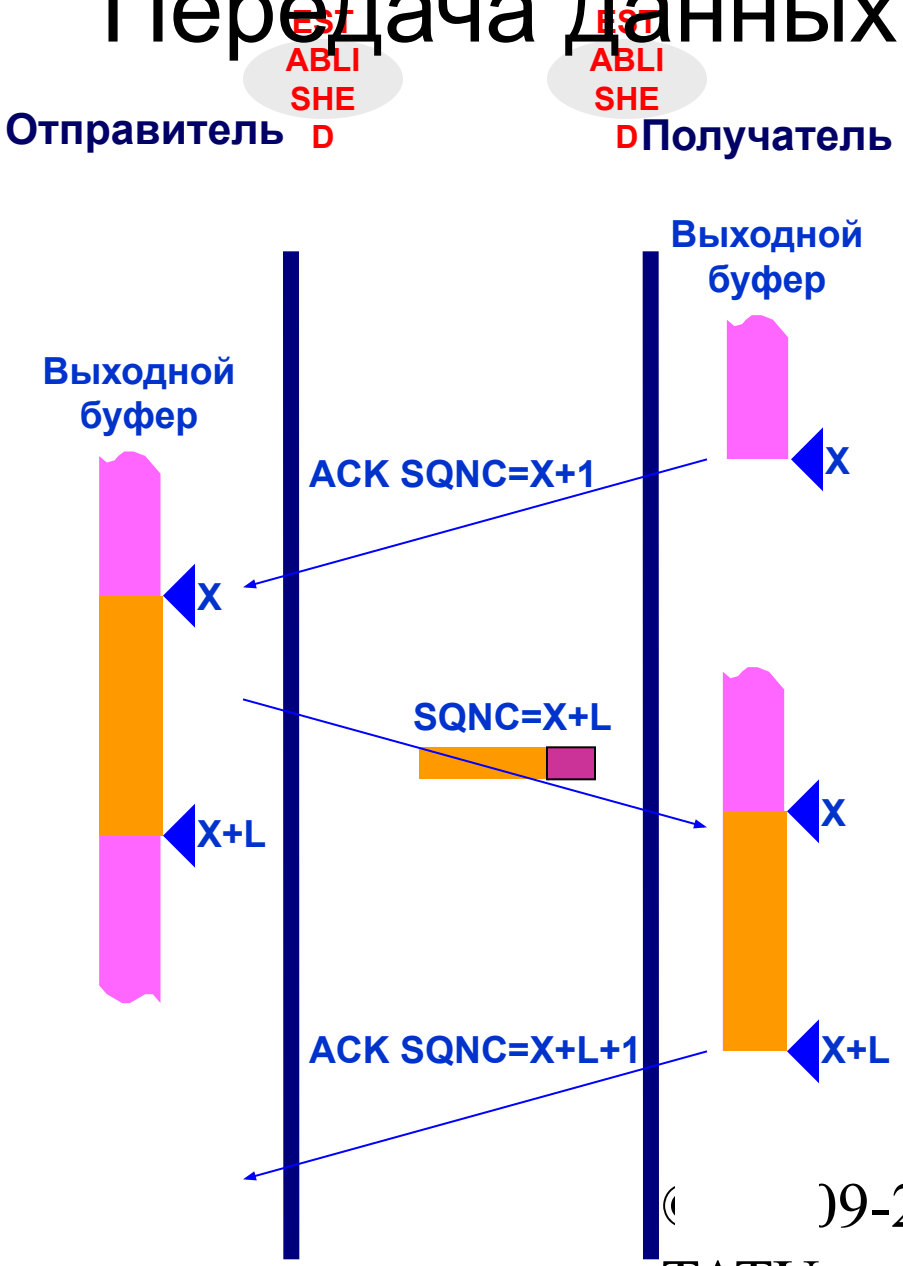
1. Входной счетчик = X
2. Квитирование (ACK SQNC=X+1)
3. Инициализация выходного счетчика (SQNC=Y), где Y – случайное число
4. Отправка SYN/ACK

Результаты:

1. **стороны готовы к приему/передаче и уведомили друг друга об этом**
2. **счетчики потоков установлены в начальные состояния**

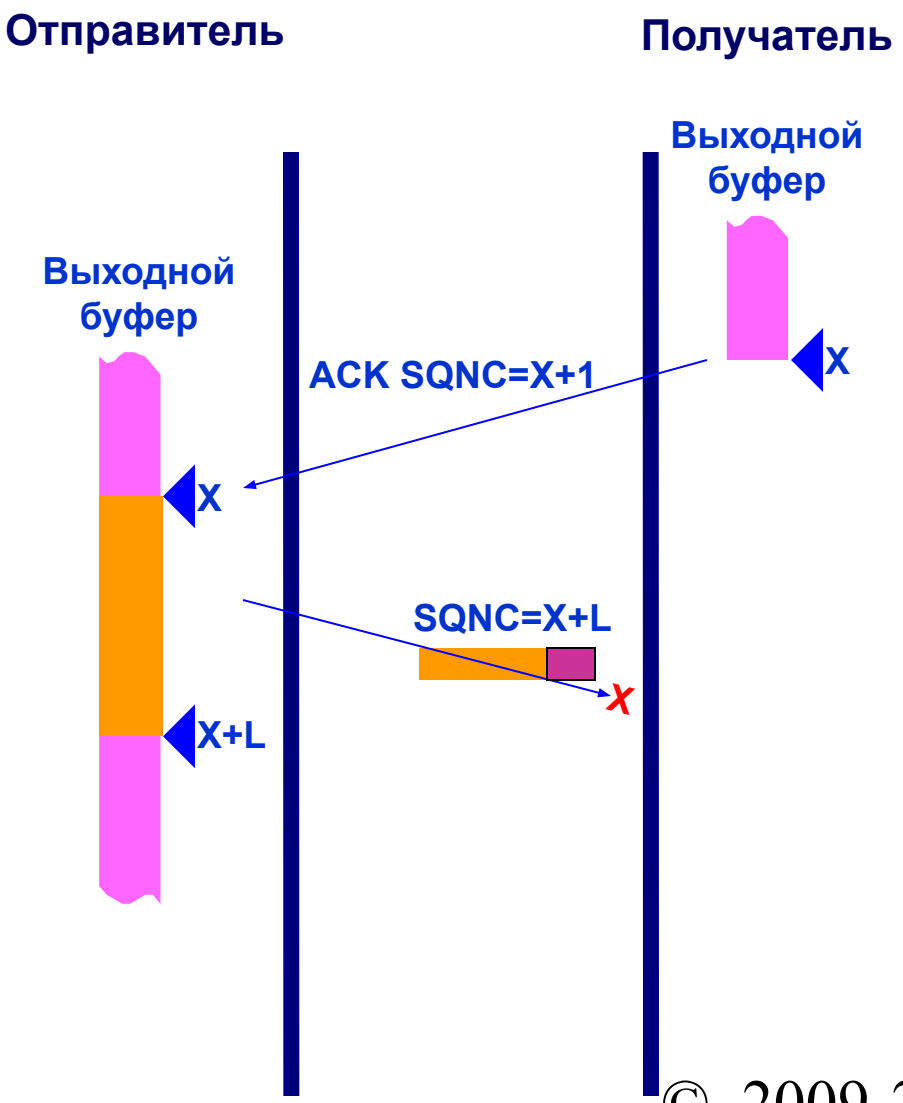
© 2009-2010 ГАТИ

Передача данных. Квитирование



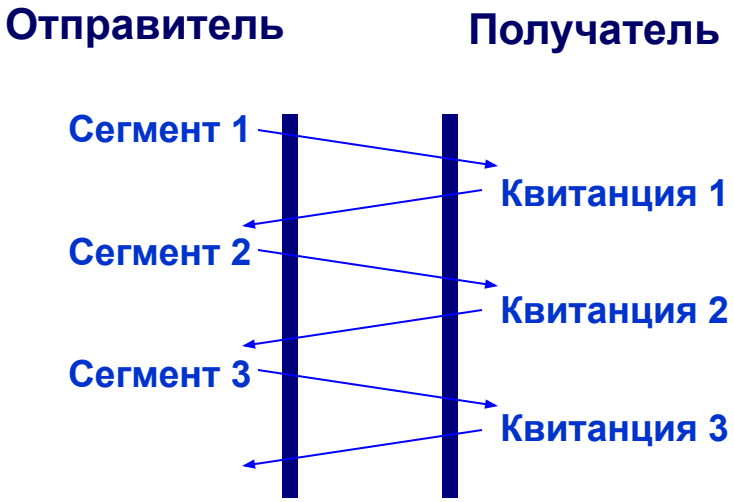
- Отправитель берет из выходного буфера очередную порцию данных, формирует TCP-сегмент, рассчитывает контрольную сумму (см. «Контрольная сумма Т-протоколов»), высылает сегмент, устанавливает тайм-аут на ожидание квитанции
- Получатель получает сегмент, сверяет контрольную сумму, выдает квитанцию
  - если контрольная сумма сошлась – ACK SQNC=X+L+1 (ожидание порции потока со следующего байта)
  - если сумма не сошлась – квитанция не высылается
- Получив квитанцию, отправитель перемещает счетчик переданного потока в позицию, соответствующую ACK SQNC

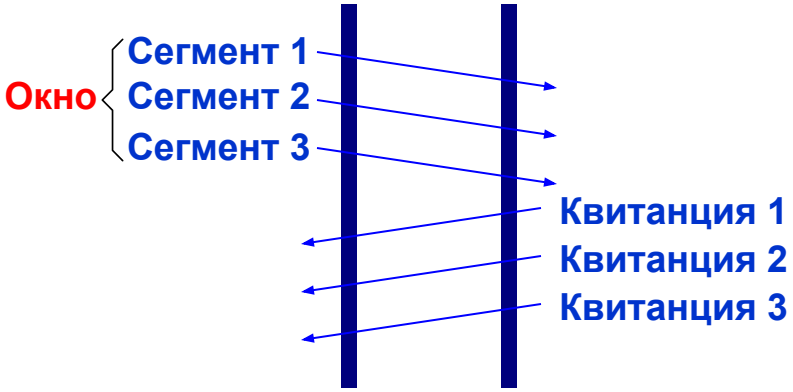
Настройка тайм-аута (очень упрощенно)



- Если сегмент (или квитанция) потеряны – отправитель по истечении тайм-аута повторяет передачу сегмента
- Длительность тайм-аута должна быть настроена для пары отправитель-получатель
  - если оба в одной локальной сети – тайм-аут м.б. несколько миллисекунд
  - если на разных концах земли – требуется тайм-аут 1-10 с
- TCP производит измерение времени до прихода квитанции (round trip time, RTT)
- Результаты измерений RTT усредняются с убывающим для более ранних измерений весом
- Длительность тайм-аута выбирается пропорциональной усредненному (с убывающим во времени весом) времени двойного прохода

Оконное управление потоком

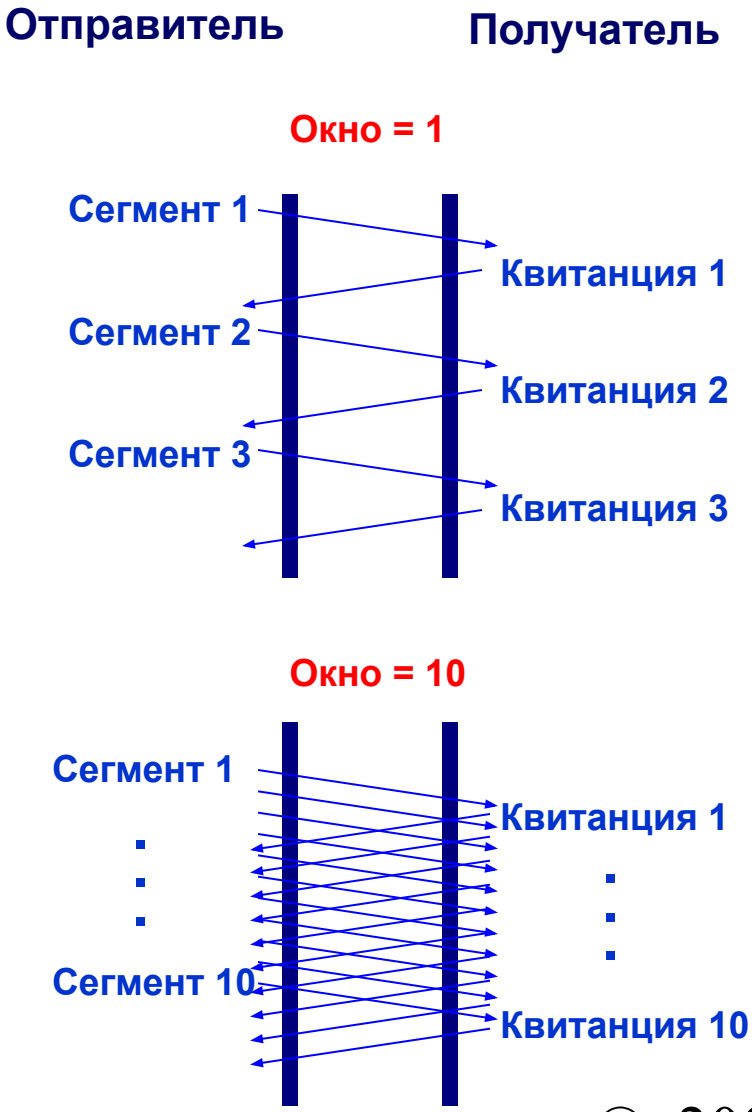




- Если отправлять сегменты только после поступления квитанций (верхний рисунок), пропускная способность линии сильно падает из-за больших времен ожидания квитанций
- Эффективность можно существенно поднять, если позволить отправителю высылать N сегментов до поручения квитанции на 1й сегмент из серии N (нижний рисунок)
  - число N называется [*скользящим*] *окном*, а этот механизм – *оконным управлением потоком*

© 2009-2010 TATU

Оконное управление потоком

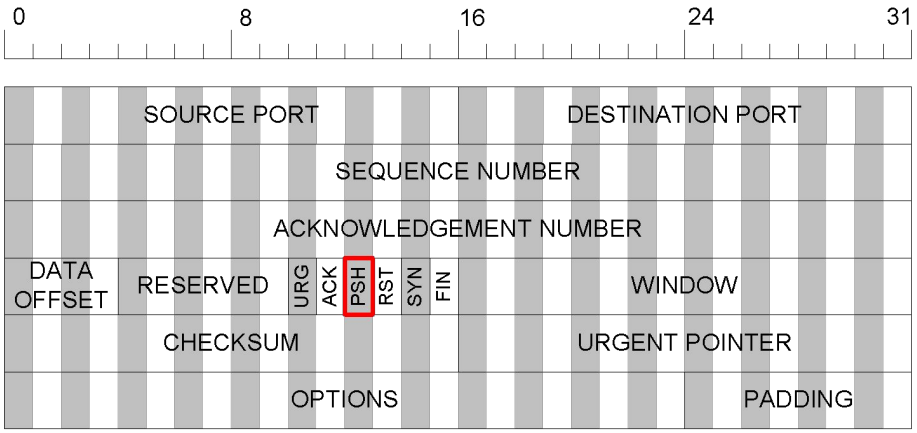


- Изменение размера окна позволяет эффективно управлять интенсивностью потока данных
  - при N=1 реализуется последовательная передача сегмент-квитанция
  - при больших N реализуется практически непрерывный дуплексный поток сегментов и квитанций
- Механизм оконного управления потоком используется в TCP/IP для управления загрузкой сети (при перегрузке производится уменьшение окон передающих трафик узлов)

Управление перегрузкой сети (упрощенно)

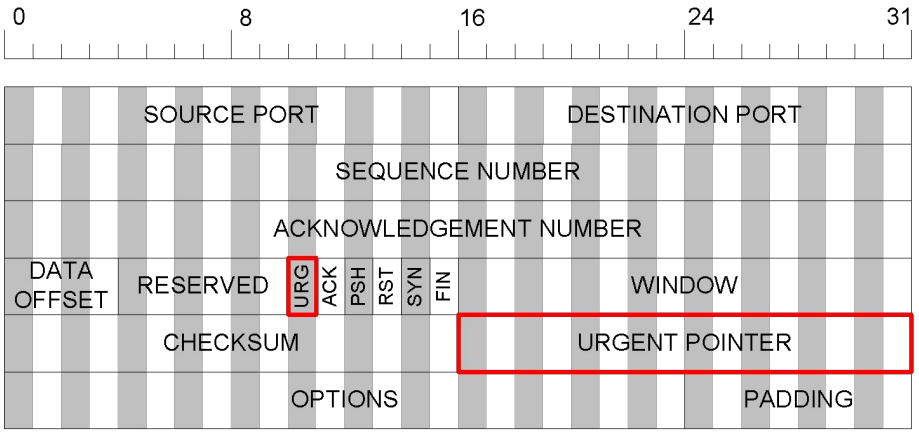
- Перегрузка на промежуточном устройстве диагностируется по увеличению задержки передачи пакетов (дополнительно – по ICMP-сообщениям от промежуточных маршрутизаторов)
- Методы управления перегрузкой:
  - на оконечных устройствах:
    - мультипликативное уменьшение окна (всякий раз вдвое, вплоть до 1) и увеличение тайм-аута
    - медленный старт: после восстановления работоспособности сети (устранение перегрузки) – увеличение окна вдвое (на 1 сегмент) по всякому факту подтверждения приема до размера окна получателя
  - на промежуточных устройствах:
    - усечение (сброс) хвоста очереди или, более поздний и оптимальный механизм – произвольный ранний сброс хвоста очереди

Принудительная передача данных



- Отправитель накапливает данные во входном буфере
  - иногда, например, после набора команды в терминальном режиме, требуется передать данные срочно, не ожидая наполнения буфера
  - для этого:
    - в прикладном интерфейсе TCP используется команда push, «выталкивающая» данные из выходного буфера в сеть
    - бит PSH устанавливается в значение 1, чтобы принимающий трафик узел немедленно произвел прием данных

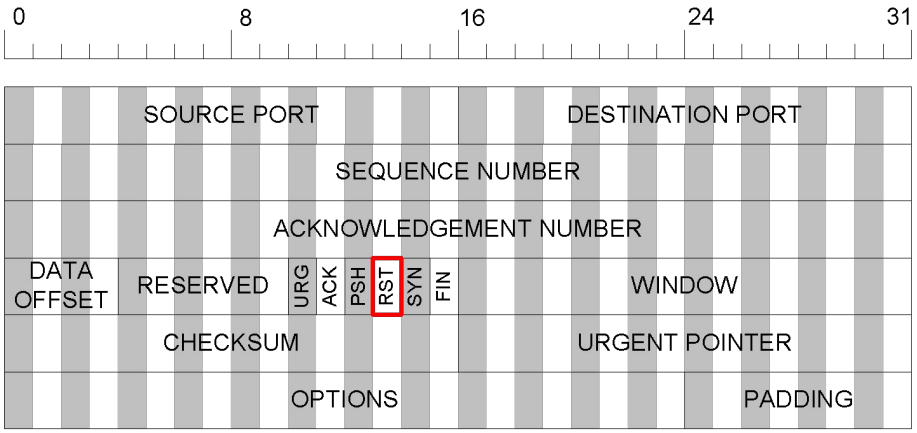
Передача вне [приемной] очереди



- В случае необходимости передать данные срочно, вне очереди (out of band), например, для передачи запроса на перезагрузку удаленного компьютера нужно указать получателю на подлежащие срочному приему данные:
  - бит URG=1
  - указатель срочных данных указывает на позицию срочных данных в сегменте
- Получатель примет данные, игнорируя необработанную входную очередь

Завершение TCP-соединения

Сброс TCP-соединения



- Когда следует прекратить связь, а штатное завершение TCP-соединения по каким-либо причинам невозможно, используется аварийный механизм сброса соединения
  - инициатор высылает сегмент с установленным битом RST
  - получатель немедленно разрывает соединение

Литература

1. TCP/IP protocol brief (Cisco Systems)
2. TCP/IP Tutorial and Technical Overview (IBM)
3. Стек TCP/IP (перевод World of Protocols, © RADCOM Ltd., 1999, публикация www.protocols.ru)
4. Internetworking with TCP/IP by Douglas E. Comer, Prentice Hall, 4th edition, January 18, 2000)
5. RFC 768, User Datagram Protocol
6. RFC 791, Internet Protocol
7. RFC 793, Transmission Control Protocol
8. RFC 826, An Ethernet Address Resolution Protocol
9. RFC 919, Broadcasting Internet Datagrams
10. RFC 922, Broadcasting Internet Datagrams In the Presence of Subnets
11. RFC 950, Internet Standard Subnetting Procedure
12. RFC 951, Bootstrap Protocol (BOOTP)
13. RFC 1112, Host Extensions for IP Multicasting
14. RFC 1293, Inverse Address Resolution Protocol
15. RFC 1918, Address Allocation for Private Internets
16. RFC 2131, Dynamic Host Configuration Protocol
17. RFC 2132, DHCP Options and BOOTP Vendor Extensions
18. RFC 2236, Internet Group Management Protocol, Version 2
19. RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers