



Введение в технологии TSP/IP

Рябко С.Д., к.ф.-м.н.

Декомпозиция управления
Архитектура TCP/IP
Протокол IP
Транспортные протоколы
Литература

Декомпозиция управления

s•terra

C S P

Cisco Solution Technology Integrator

● **Понятие протокола**

★ **[Сетевой] протокол – это распределенный алгоритм, выполняемый на нескольких компьютерах**

основной и первичной задачей сетевых протоколов было обеспечение передачи данных

- **собственно организация сервиса передачи данных**
- **организация маршрутизации**
- **сигналинг, 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**

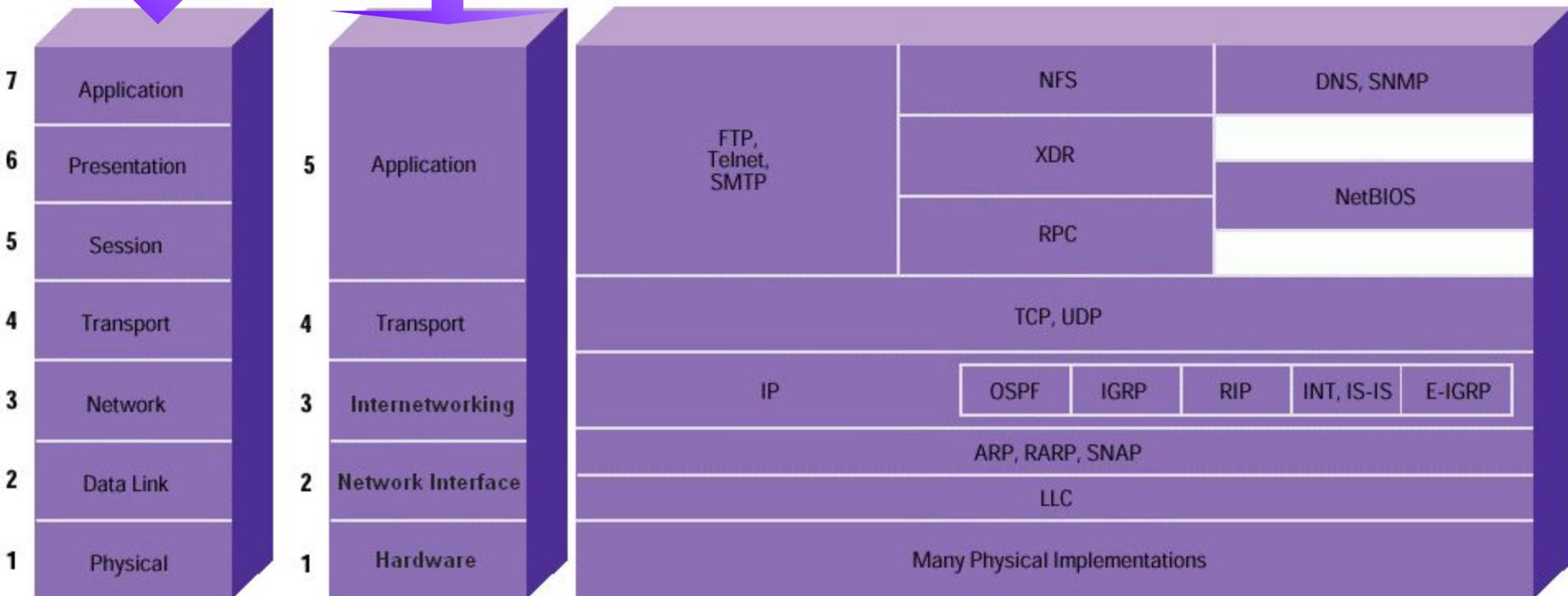
● Декомпозиция сетевого управления

Модель OSI/ISO

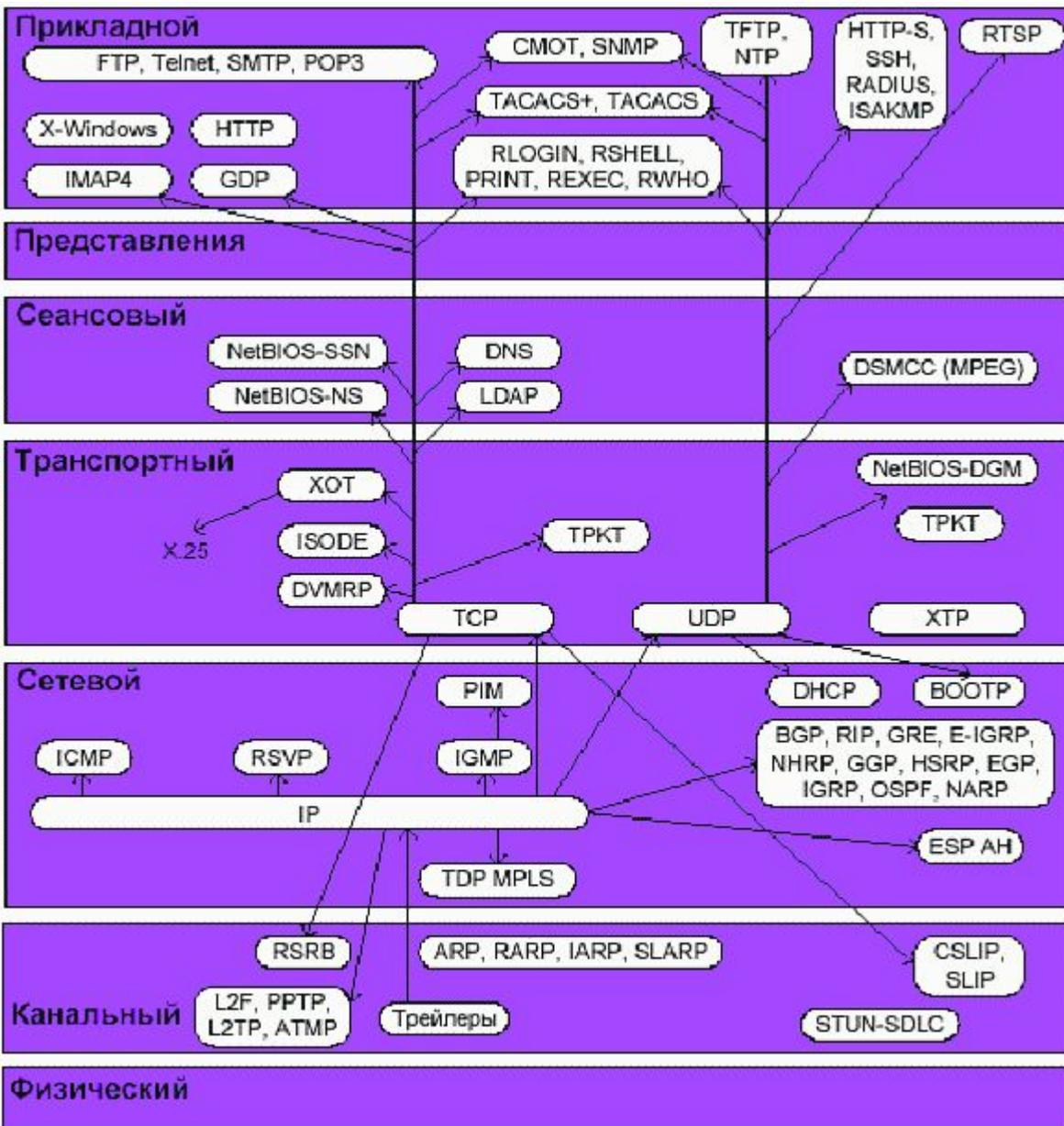
Модель TCP/IP



С точностью до незначительных различий можно считать, что функциональность второго (канального), третьего (сетевое) и четвертого (транспортного) уровней в моделях OSI/ISO и TCP/IP совпадают



Справка. Наиболее важные протоколы



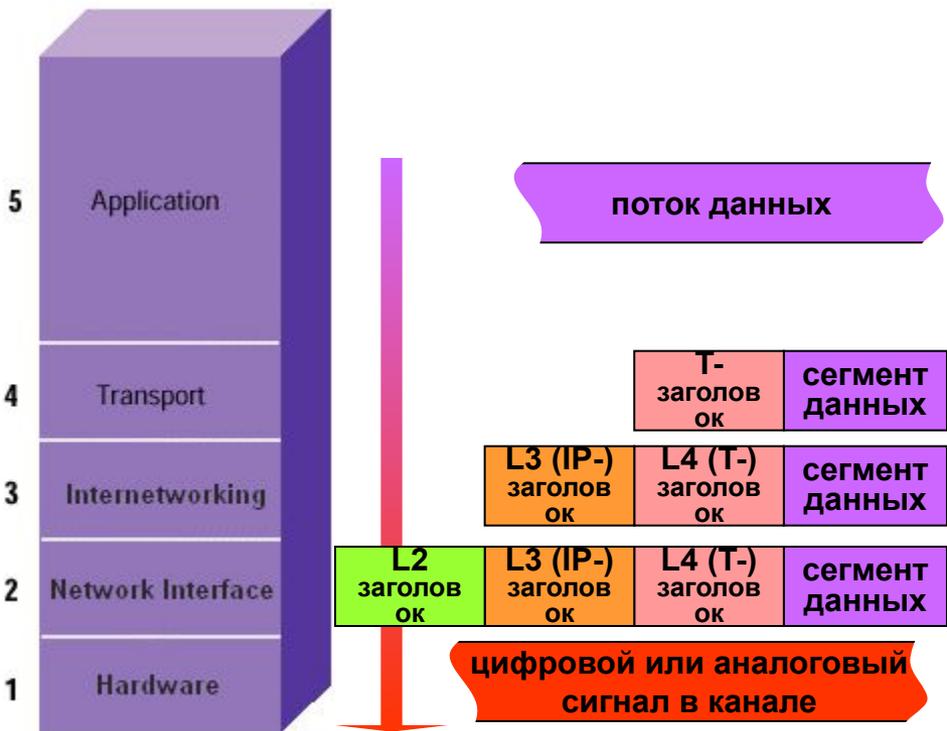
Эта картинка с www.protocols.ru - не бесспорна по декомпозиции и не совсем полна, однако уточняет состав протоколов и показывает их взаимодействия

приводится для справки

● Задачи протоколов различных уровней

1. **Физический уровень**
обеспечивает стандартизацию сред передачи, (носителей, частот, контактов, типов модуляции, сигналов)
2. **Канальный уровень**
обеспечивает передачу данных на однопролетном звене (связность смежных узлов)
передает *фреймы (frame)*, иногда говорят - кадры
3. **Сетевой уровень**
обеспечивает передачу данных из конца 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 уровни последовательно упаковывают (инкапсулируют) данные «внутри» 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



Декомпозиция управления

Архитектура TCP/IP

Протокол IP

Транспортные протоколы

Литература

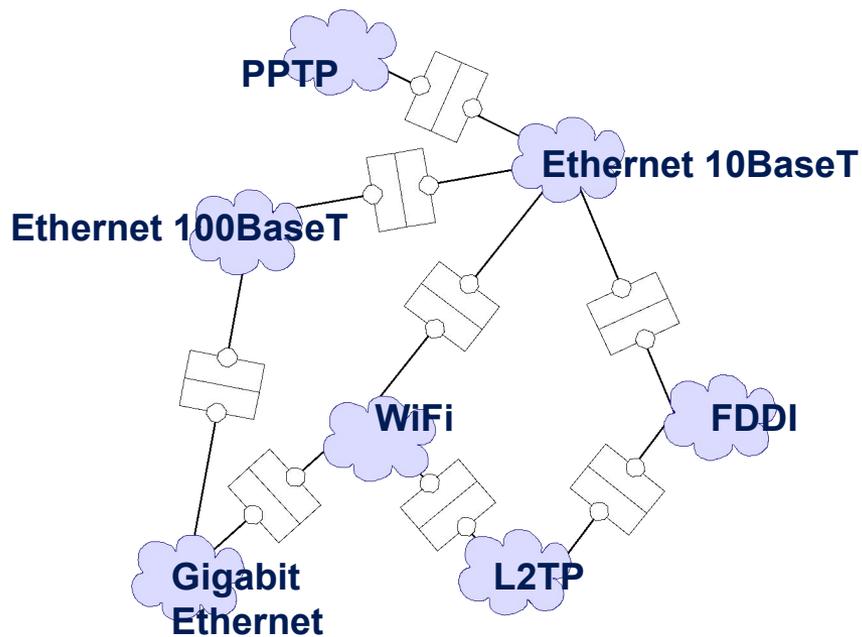
Архитектура TCP/IP

s•terra

C S P

Cisco Solution Technology Integrator

● ТСП/IP: независимость от среды передачи

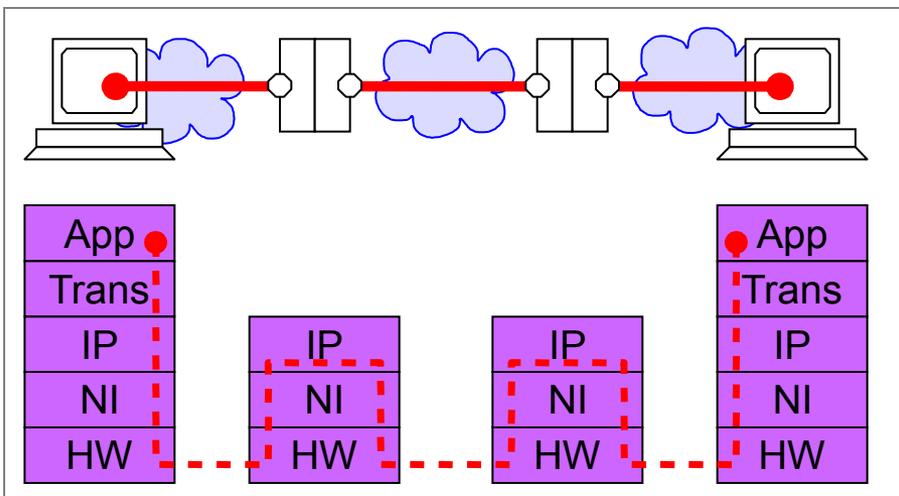
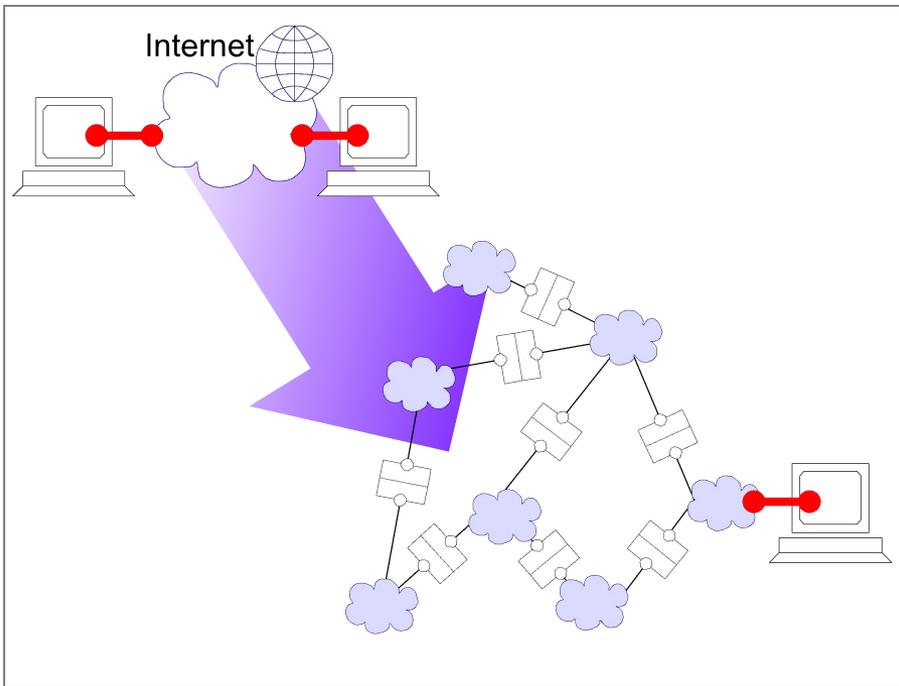


В сущности, «носителями протоколов ТСП/IP» являются не физические подсети, а стеки сетевых компьютеров (хостов) и шлюзов (маршрутизаторов)



- ★ Физические подсети могут иметь различную природу и различные системы адресации канального уровня
- ★ Стек ТСП/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

● Метафора 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 на конечных устройствах, но на промежуточных устройствах (шлюзах) используются только три нижних уровня сетевого стека**

● Функциональная декомпозиция ТСР/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:

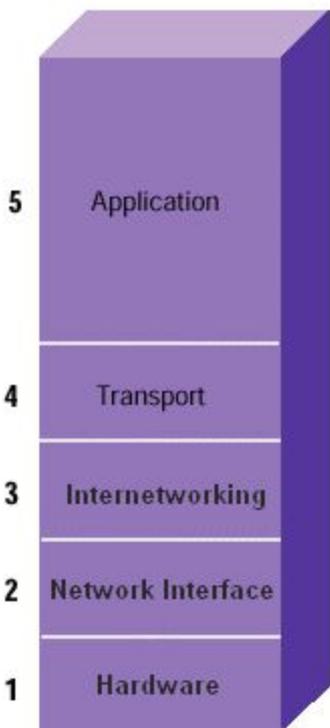
адресация сетевых объектов (включая конфигурирование адресов)

маршрутизация

обмен служебной информацией, разрешение конфликтных ситуаций, диагностика

- ✱ 2 главных задачи транспортных протоколов:

обеспечение заданного сервиса доставки данных
мультиплексирование/демультиплексирование трафика приложений



Декомпозиция управления

Архитектура TCP/IP

Протокол IP

- IP-пакет

- адресация

- маршрутизация

- служебные обмены

Транспортные протоколы

Литература

Протокол IP: IP-пакет

s•terra

C S P

Cisco Solution Technology Integrator

Структура IP-пакета

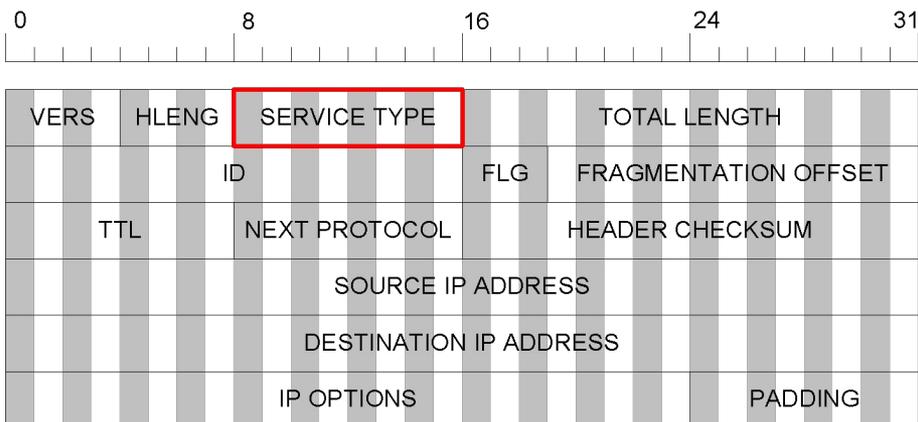


Заголовок IP-пакета



VERS	HLENG	SERVICE TYPE		TOTAL LENGTH
		ID	FLG	FRAGMENTATION OFFSET
TTL		NEXT PROTOCOL		HEADER CHECKSUM
SOURCE IP ADDRESS				
DESTINATION IP ADDRESS				
IP OPTIONS			PADDING	

- * **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



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, precedence – приоритет:

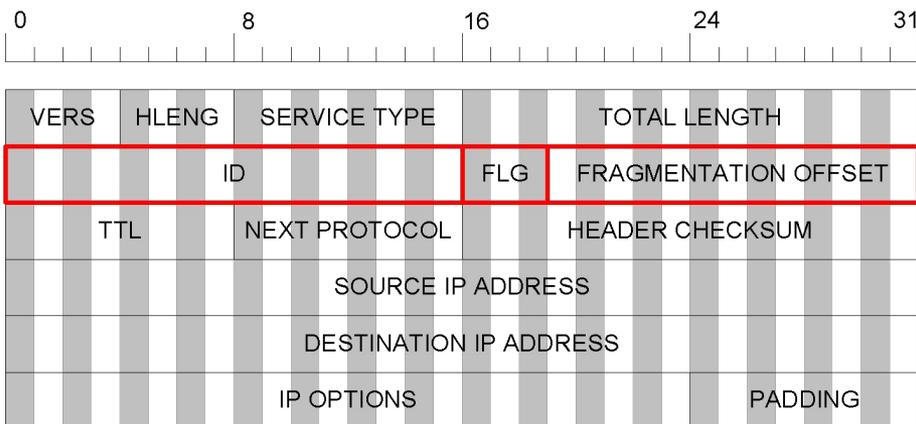
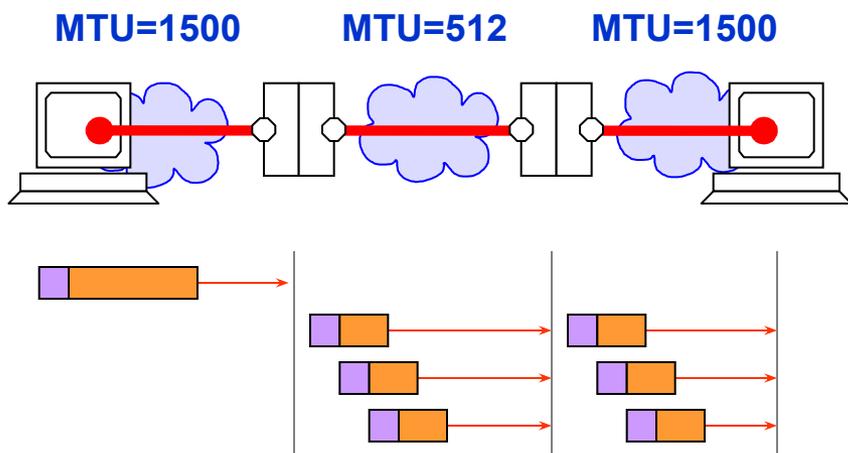
- 000: Routine
- 001: Priority
- 010: Immediate
- 011: Flash
- 100: Flash override
- 101: Critical
- 110: Internetwork control
- 111: Network control

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

● 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, FRGMENTTNOFFSET

ID – уникальный идентификатор, единый для всех фрагментов серии

поле FLG:

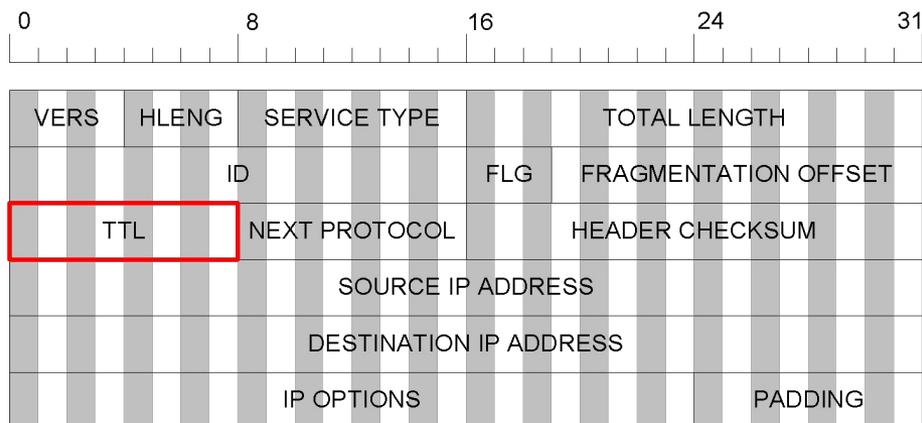
- 1й бит – резерв, всегда 0
- 2й бит – DF, Do not Fragment – запрещает фрагментацию
- бит MF – More Fragments – 0 для нефрагментированного или последнего пакета в серии, 1 – в противном случае

● Время жизни IP пакета

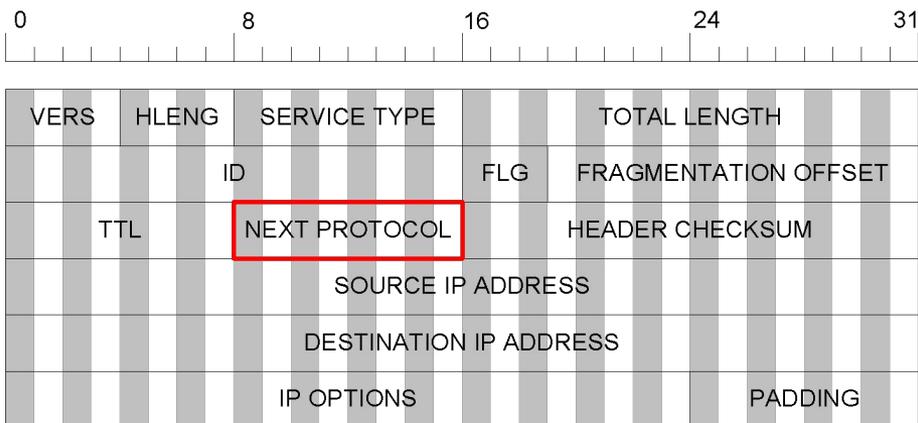
- ✦ В силу ошибок маршрутизации или по другим причинам пакет может бесконечно циркулировать по некоторому пути в сети

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



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



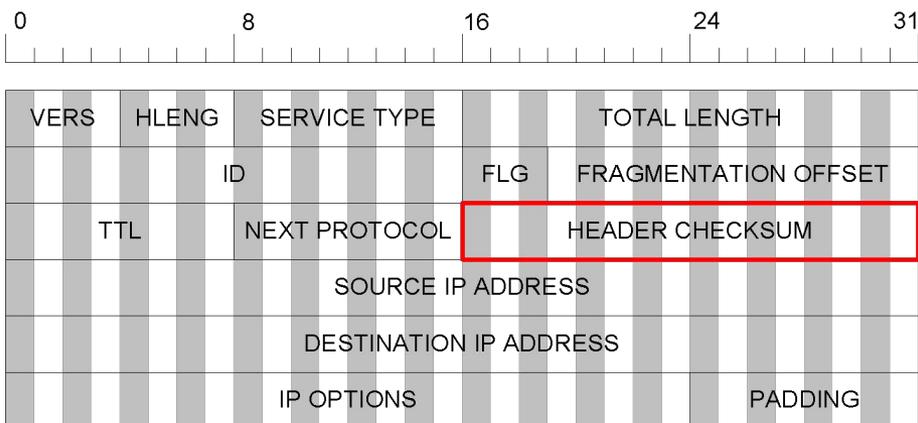
★ IP может «нести» данные различных протоколов, номер «вложенного» протокола кодируется в поле **NEXT PROTOCOL**:

- 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)

● Целостность IP-пакетов

- ✦ IP (если не применяются специальные протоколы защиты информации AH и ESP) вообще не следит за целостностью IP-пакетов

в этом есть резон, поскольку 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)



Декомпозиция управления

Архитектура TCP/IP

Протокол IP

- IP-пакет

- адресация

- маршрутизация

- служебные обмены

Транспортные протоколы

Литература

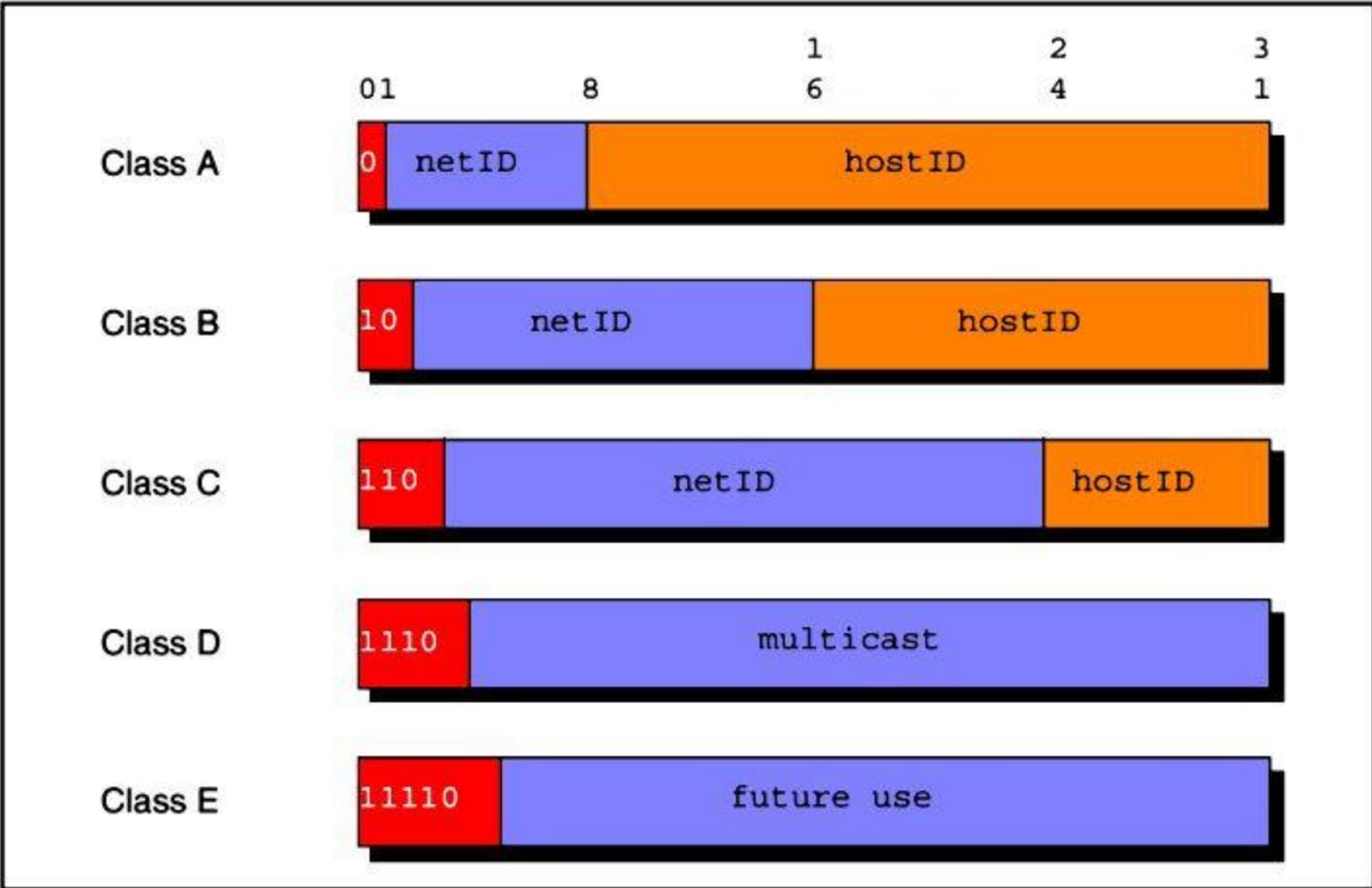
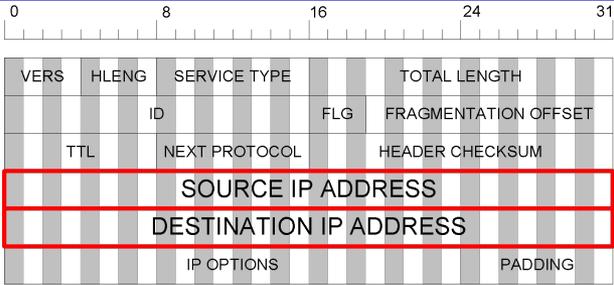
Протокол IP: адресация

s•terra

C S P

Cisco Solution Technology Integrator

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

- логическое 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

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, широкоэмитательный 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 заполняется единицами)

используется для задач отладки и тестирования не является адресом никакой сети и роутеры никогда не обрабатывают его



● Частные адресные пространства

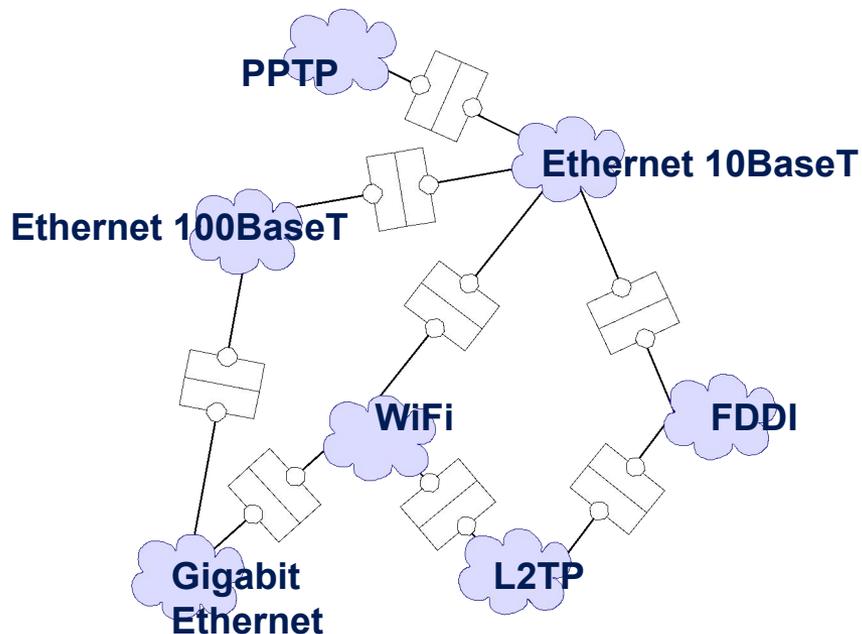
- ★ Уникальное (регистрируемое) адресное пространство Интернет почти исчерпано; поэтому для пользования внутри собственных сетей (без выхода в Интернет) организации могут использовать т.н. частные (private), незарегистрированные адреса

RFC1918 рекомендует использовать для этих целей следующие адресные пространства:

- сеть класса А – 10.0.0.0
- 16 сетей класса В – от 172.16.0.0 до 172.31.0.0
- 256 сетей класса С – 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 А необходимо связаться с узлом В, узел А знает IP-адрес узла В, но не знает его физического адреса, узел А шлет широковещательное сообщение, в котором запрашивает физический адрес узла В

все узлы принимают это сообщение, однако только узел В отвечает на него, высылая в ответ свой физический адрес узлу А

узел А, получив физический адрес В, кэширует его, с тем, чтобы не запрашивать его повторно при следующих обращениях к узлу В

- ✦ **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

● **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 DNSP-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

Декомпозиция управления

Архитектура TCP/IP

Протокол IP

- IP-пакет
- адресация
- маршрутизация
- служебные обмены

Транспортные протоколы

Литература

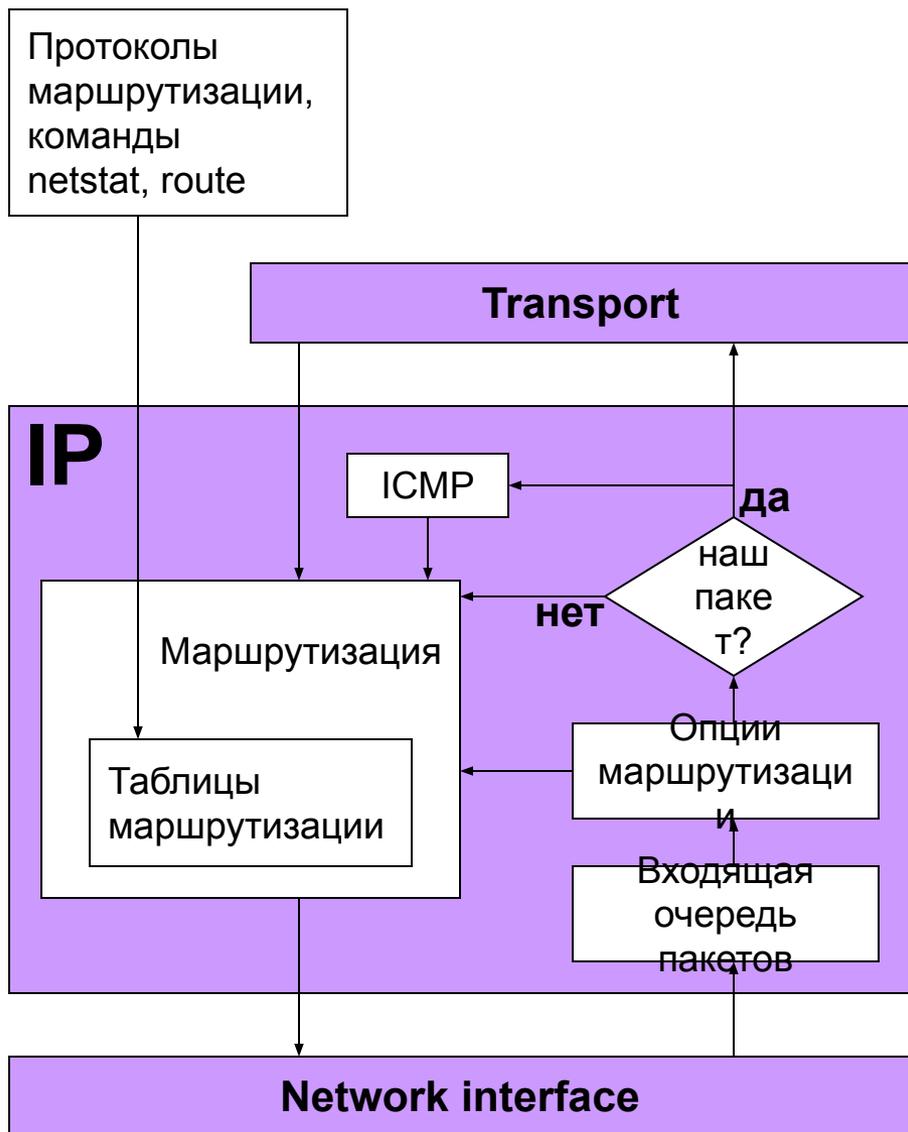
Протокол IP: маршрутизация

s•terra

C S P

Cisco Solution Technology Integrator

● Задача маршрутизации



- ✦ **Маршрутизационная задача решается локально на каждом хосте/маршрутизаторе**

постановка задачи: на какой сетевой интерфейс нужно передать пакет, чтобы он дошел до получателя?

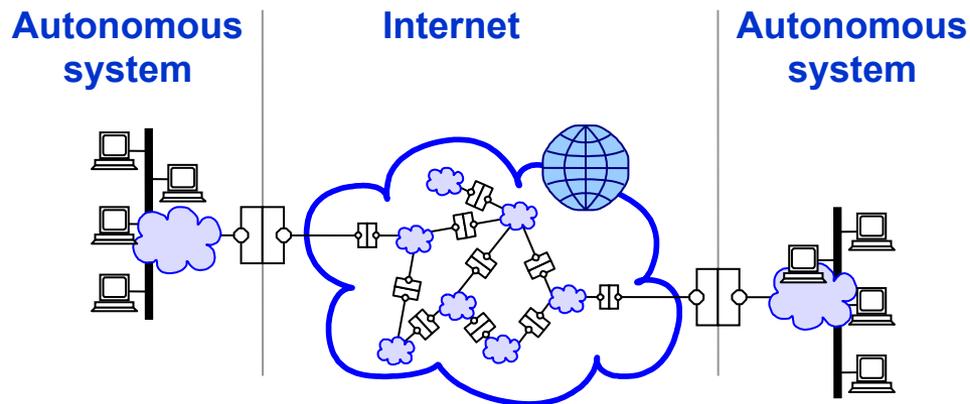
- ✦ **Маршрутизационные таблицы (routing table) указывают, куда нужно перенаправлять пакет с заданным адресом**

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 хопов), 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, м.б. 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

● 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

Декомпозиция управления

Архитектура TCP/IP

Протокол IP

- IP-пакет
- адресация
- маршрутизация
- служебные обмены

Транспортные протоколы

Литература

Протокол IP: служебные обмены и диагностика

s•terra

C S P

Cisco Solution Technology Integrator

★ 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

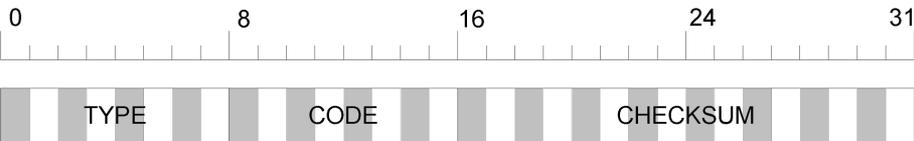
в случае ошибок фрагментации ICMP-сообщение генерируется только для первого фрагмента

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

☀ 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: traceroute [-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.

Декомпозиция управления

Архитектура TCP/IP

Протокол IP

Транспортные протоколы

- сервис

- UDP

- TCP

Литература

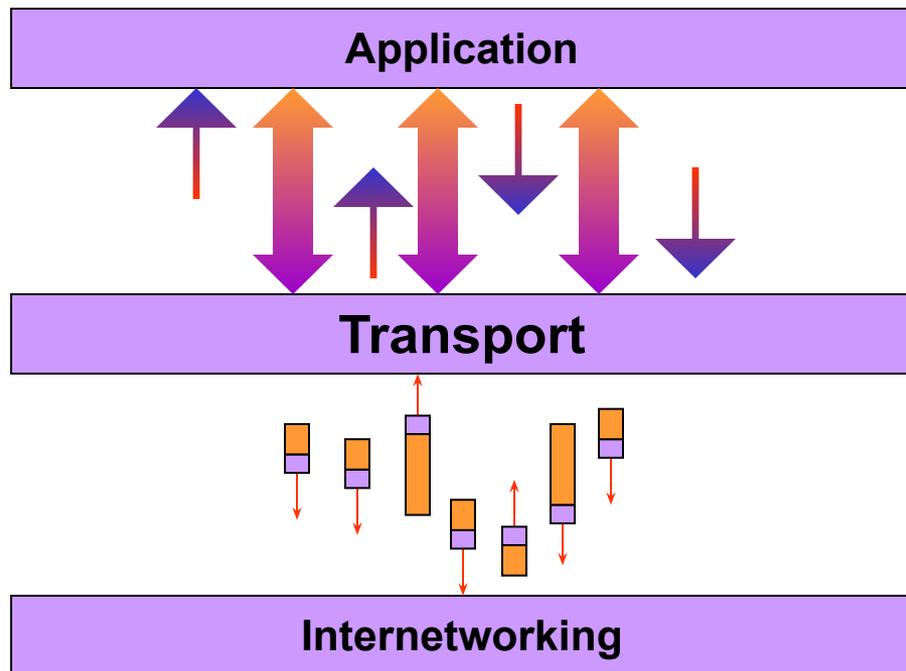
Транспортные протоколы: сервис

s•terra

C S P

Cisco Solution Technology Integrator

● Сервис транспортного уровня



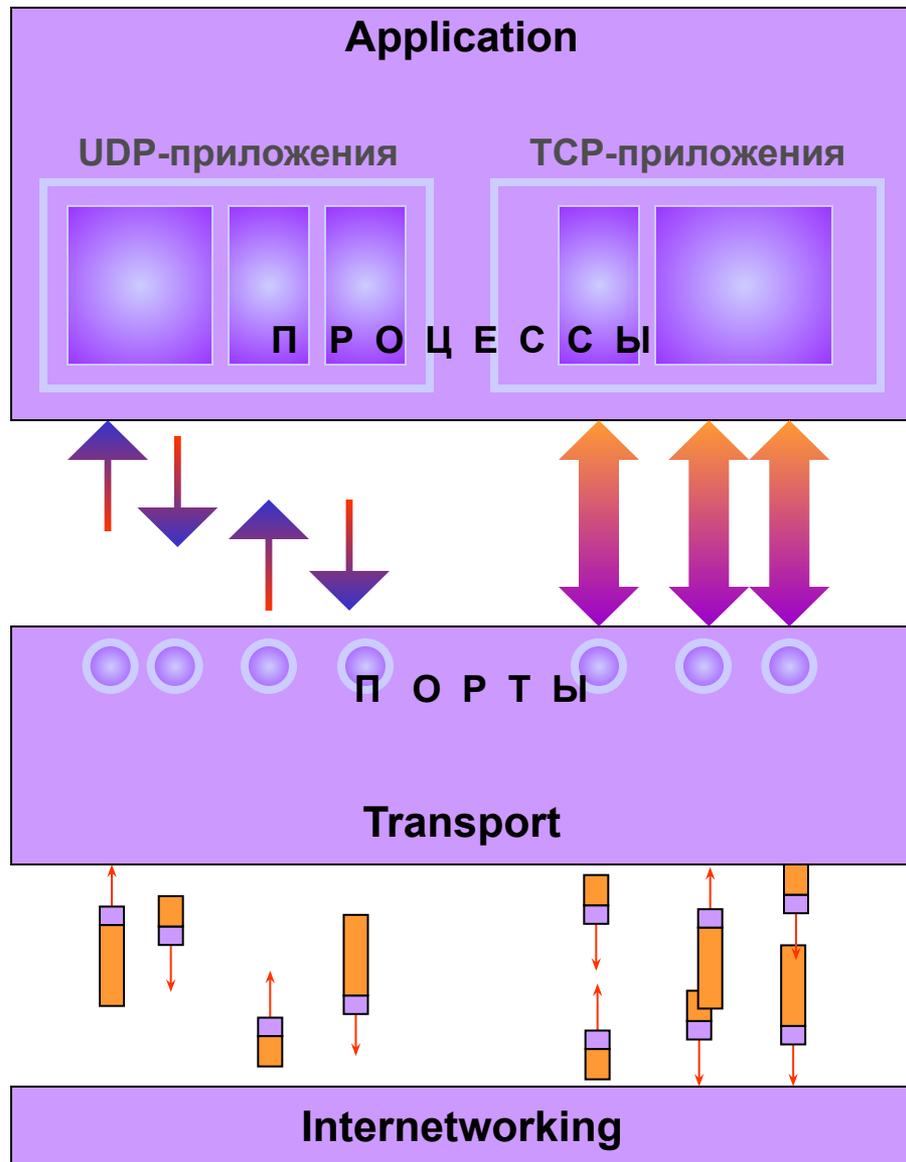
- ✱ Приложения не формируют IP-пакеты

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
поточный 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**

● 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)

Декомпозиция управления

Архитектура TCP/IP

Протокол IP

Транспортные протоколы

- сервис

- UDP

- TCP

Литература

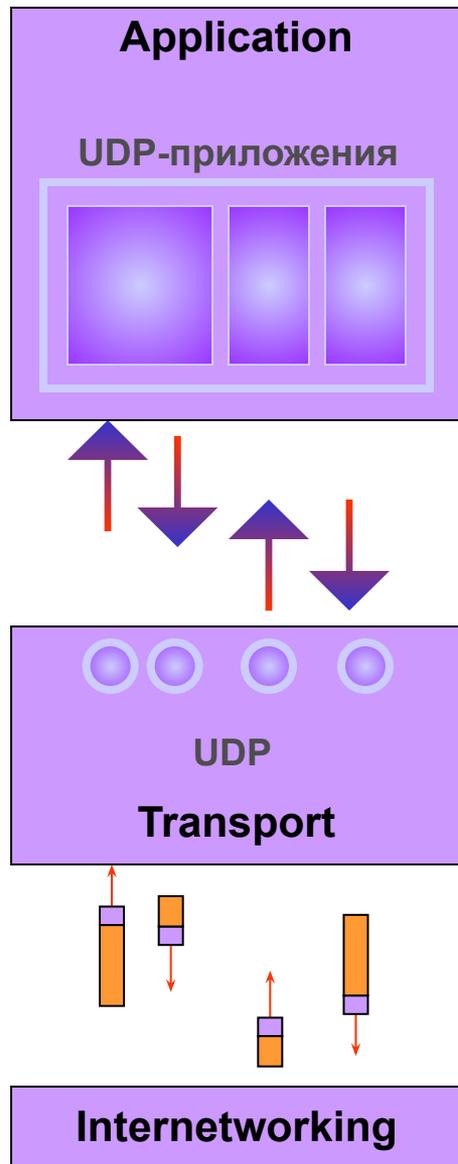
Транспортные протоколы: UDP

s•terra

C S P

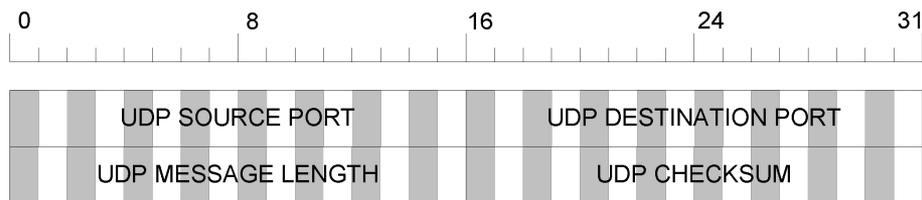
Cisco Solution Technology Integrator

● 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 дублировать сообщения) и не сохраняющий порядка следования сообщений
- ★ Сообщение протокола **UDP** называют *пользовательской дейтаграммой* (user datagram)

● Дейтаграмма UDP



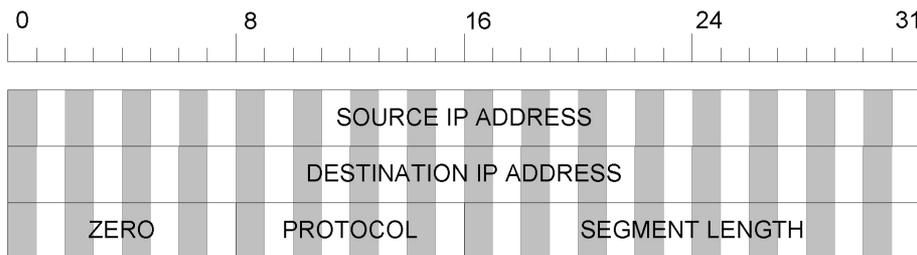
- ★ **UDP SOURCE PORT и UDP DESTINATION PORT –** порты процесса-отправителя и процесса-получателя
 - source port имеет ненулевое заполнение, если процесс-отправитель должен получить ответное сообщение**
- ★ **UDP MESSAGE LENGTH –** полная длина заголовка и сегмента данных
- ★ **UDP CHECKSUM –** контрольная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

Декомпозиция управления

Архитектура TCP/IP

Протокол IP

Транспортные протоколы

- сервис

- UDP

- TCP

Литература

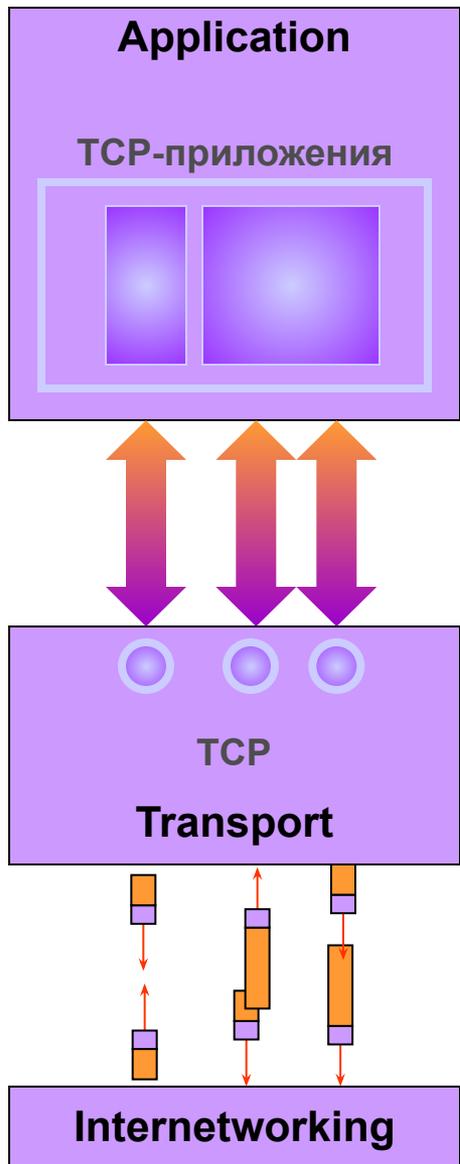
Транспортные протоколы: TCP

s•terra

C S P

Cisco Solution Technology Integrator

● Сервис протокола ТСР



★ 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

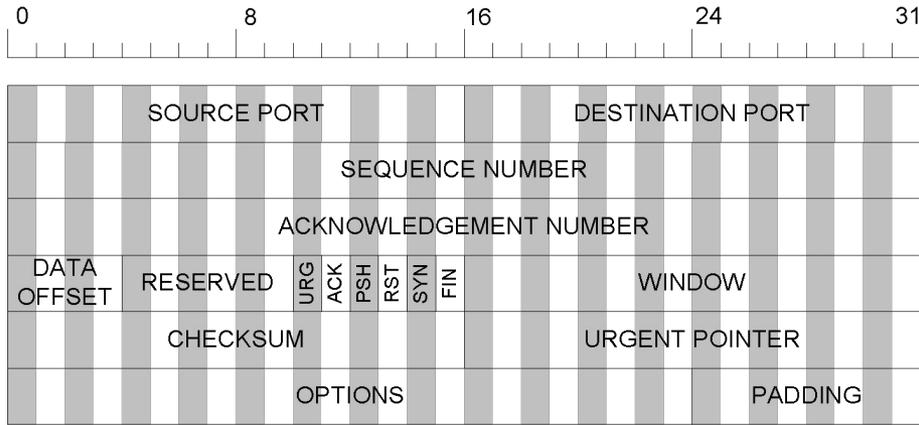
буферизует данные, передаваемые в сеть

организует т.н. виртуальные соединения посредством предварительной согласовательной процедуры

обеспечивает полнодуплексное соединение при этом обеспечивается

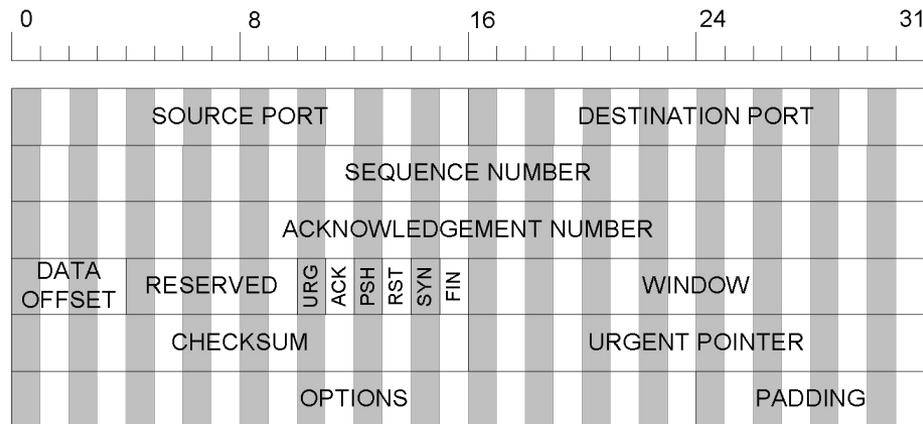
- управление потоком (в зависимости от пропускной способности и загрузки сети)

обеспечивает целостность потока и гарантирует доставку данных



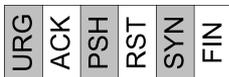
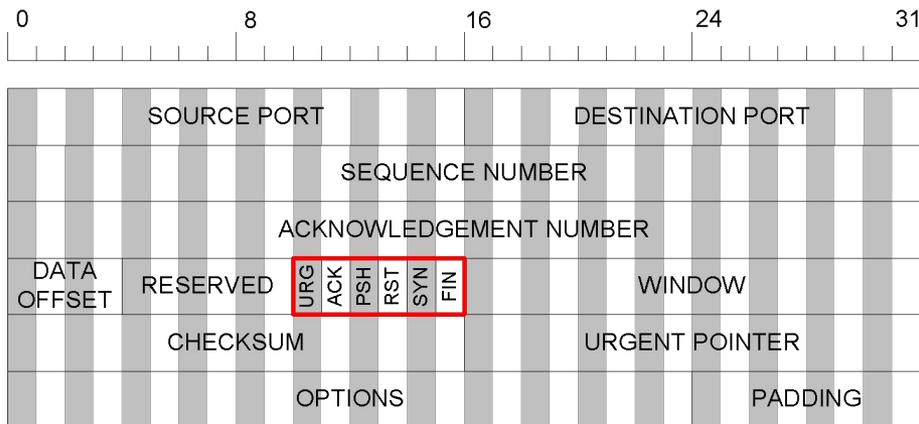
- ✦ ТСР передает данные порциями (сегментами), каждый из которых включается затем в IP пакет
- ✦ Заголовок сегмента (транспортный заголовок ТСР) обеспечивает возможность для передачи управляющей информации протокола вместе с трафиком (piggybacking)

● Поля заголовка ТСР



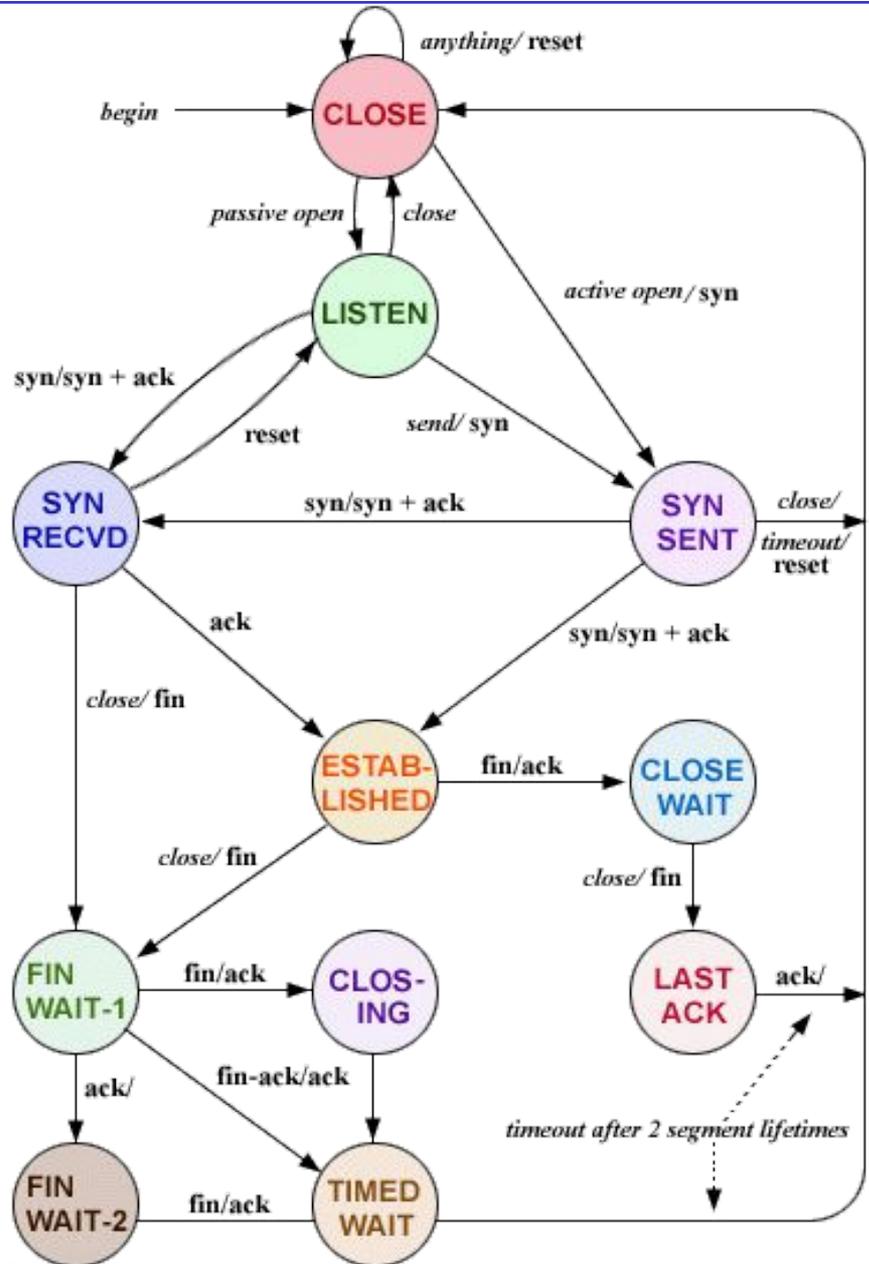
- ★ **SOURCE PORT, DESTINATION PORT** – номера портов отправителя и получателя сообщения
- ★ **SEQUENCE, ACKNOWLEDGEMENT NUMBER, WINDOW, URGENT POINTER** – поля для управления потоком
- ★ **DATA OFFSET** – указатель на конец заголовка (начало блока данных)
- ★ **CHECKSUM** – контрольная сумма по сегменту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

● Биты управления



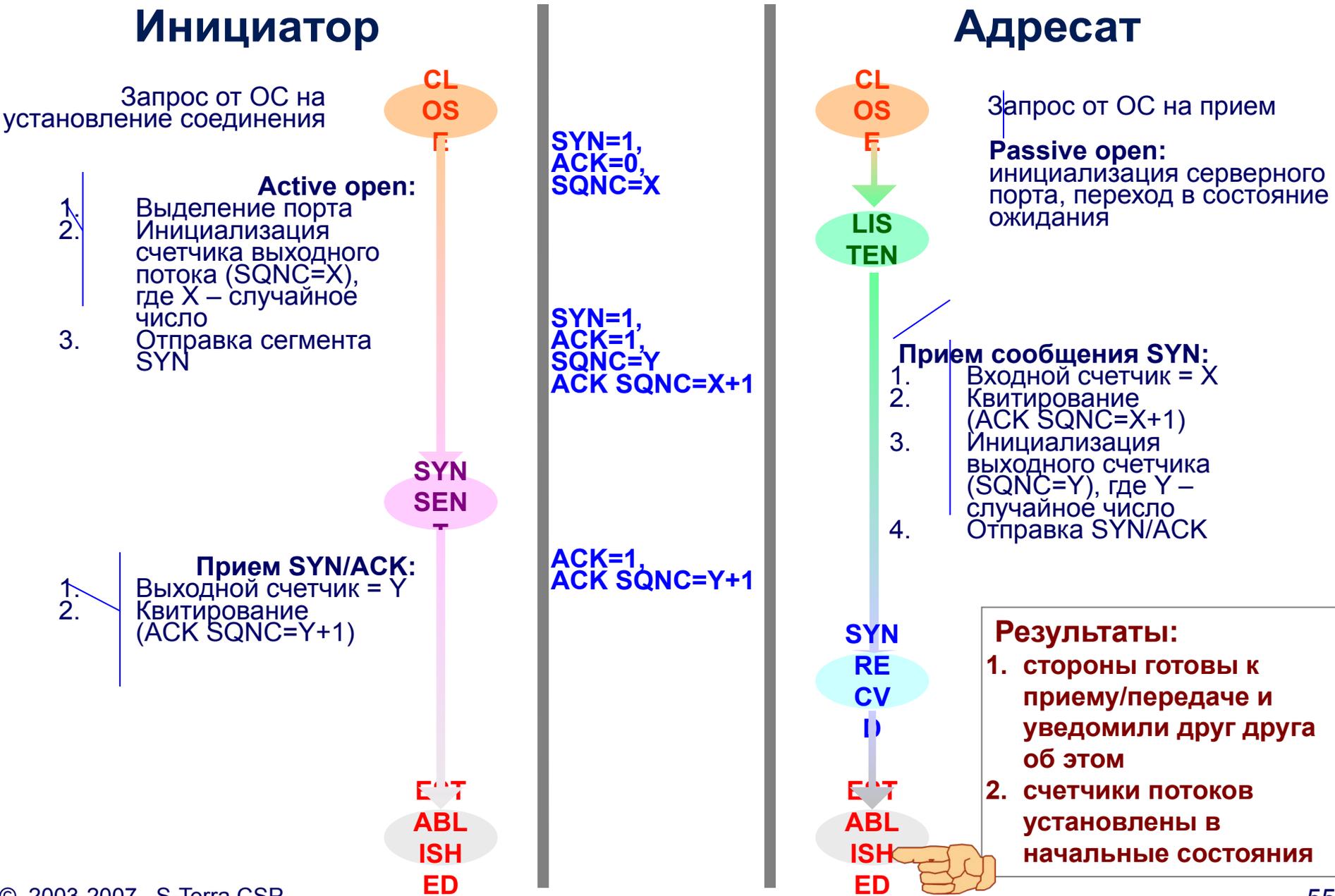
- ★ **URG, urgent** – срочная передача данных
- ★ **ACK, acknowledgement** – подтверждение приема
- ★ **PSH, push** – очистка буфера
- ★ **RST, reset** – переустановка соединения
- ★ **SYN, synchronize** – синхронизация потоков
- ★ **FIN, finish** – окончание потока данных

● Конечный автомат протокола ТСР

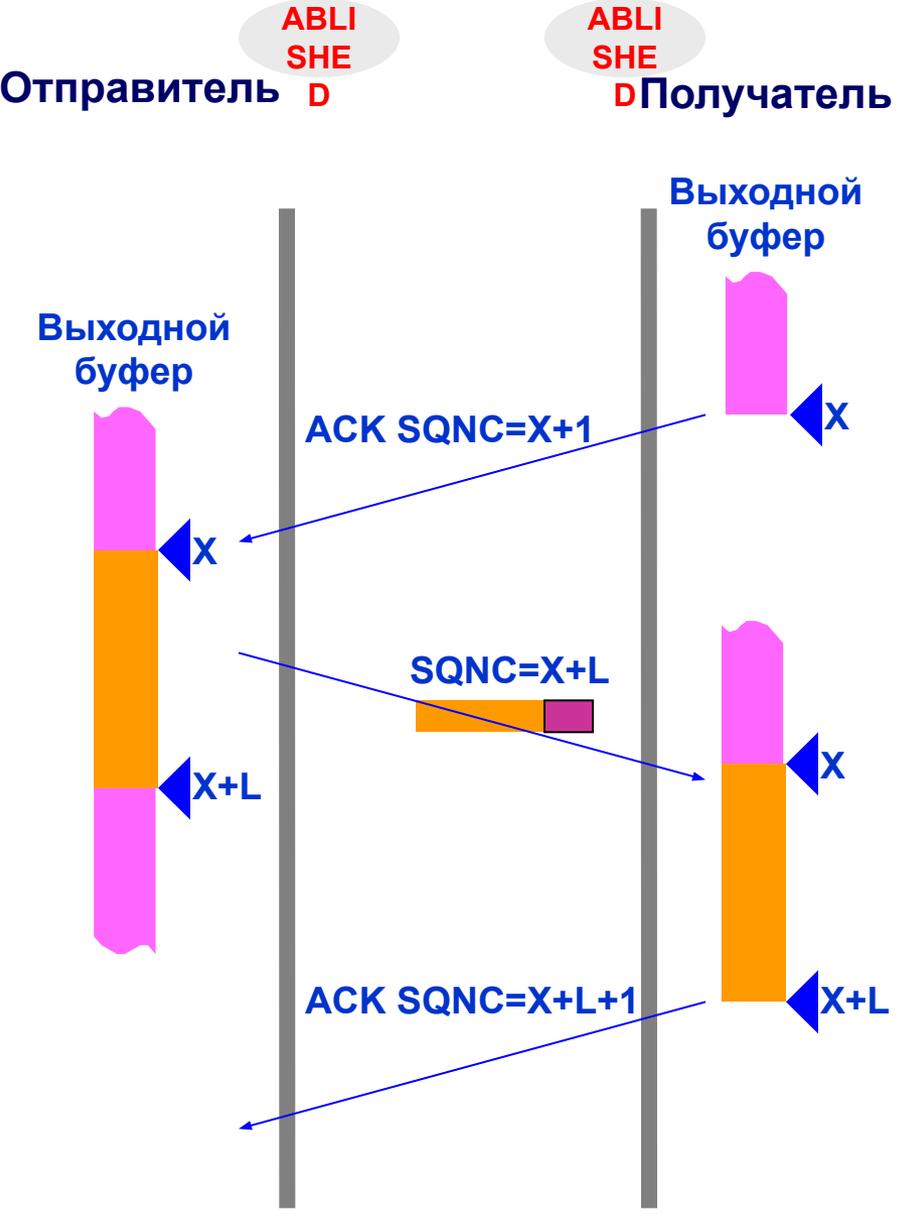


- ★ **Работу протокола ТСР удобно пояснить на основе конечного автомата; состояния:**
 - CLOSE** – холостое состояние, отсутствие соединения
 - LISTEN, SYN RECVD, SYN SENT** – промежуточные состояния фазы установления соединения
 - ESTABLISHED** – соединение установлено, передача данных
 - CLOSE WAIT,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

Установление TCP-соединения



● 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)

Отправитель

Выходной
буфер



ACK SQNC=X+1

Получатель

Выходной
буфер

X

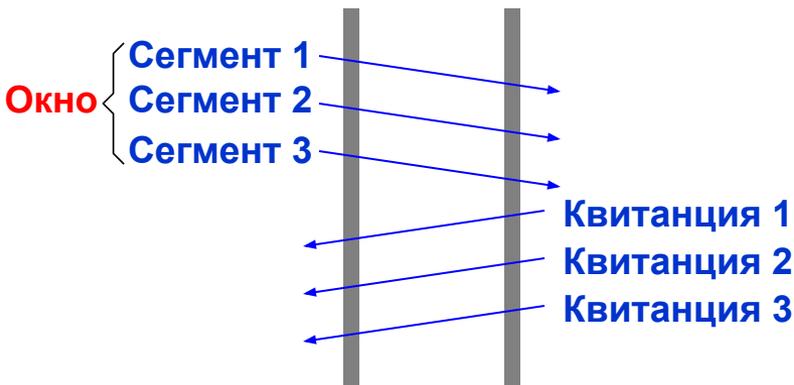
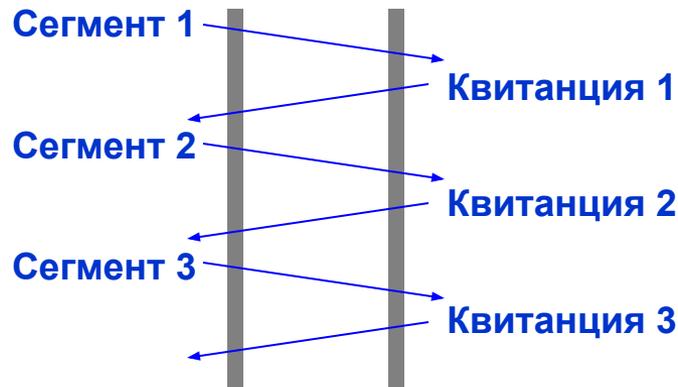
SQNC=X+L

X

- ✱ 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

Отправитель Получатель



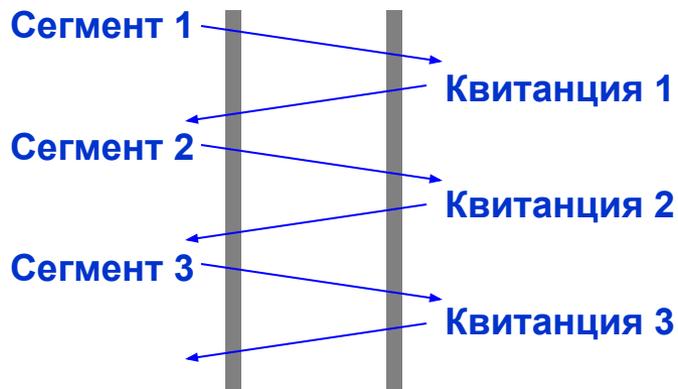
- ✦ Если отправлять сегменты только после поступления квитанций (верхний рисунок), пропускная способность линии сильно падает из-за больших времен ожидания квитанций
- ✦ Эффективность можно существенно поднять, если позволить отправителю высылать N сегментов до поручения квитанции на 1й сегмент из серии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

Оконное управление потоком

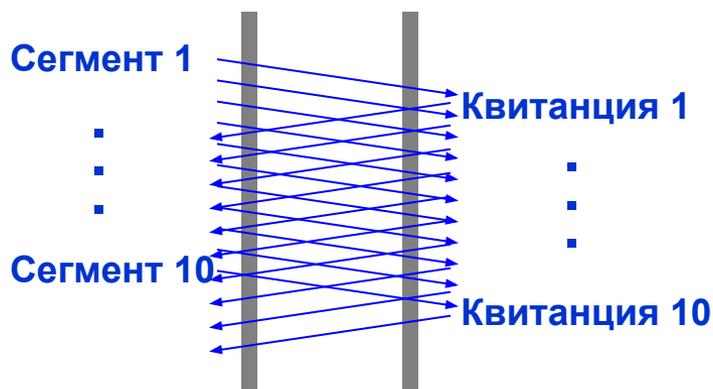
Отправитель

Получатель

Окно = 1



Окно = 10



- 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 конечных 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:

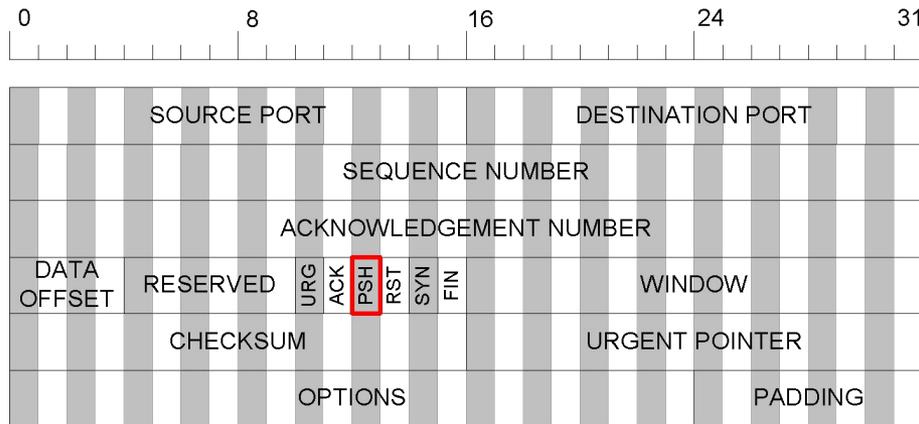
- усечение (сброс) хвоста очереди или, более поздний и оптимальный механизм – произвольный ранний сброс хвоста очереди

● Принудительная передача данных

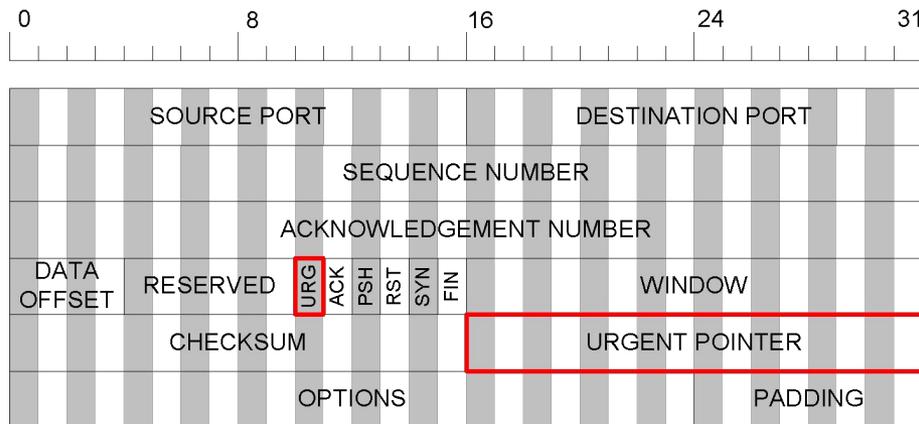
- ✦ Отправитель накапливает данные во входном буфере

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 ТСР-соединения

Инициатор

Адресат

Запрос от ОС на завершение соединения

1. Завершение передачи данных из выходного буфера (SQNC=X) и отправка сегмента с FIN=1
2. Закрытие полудуплекса на передачу

EST
ABL
ISH
ED
Close:

FIN
WAI
T 1

FIN
WAI
T 2

Прием FIN:

Отправка подтверждения завершения соединения (ACK SQNC=Y+1)

CL
OS
E

FIN=1,
SQNC=X

ACK=1
ACK SQNC=X+1

FIN=1
SQNC=Y

ACK=1,
ACK SQNC=Y+1

EST
ABL
ISH
ED
CL
CS

E
WAI
..

Подтверждение завершения

1. Отправка FIN=1
2. Переход к ожиданию подтверждения

Уведомление ОС/приложения о завершении соединения (возможно, завершение передачи данных)

Разрешение на завершение соединения (close)

L
S
T
AC
K

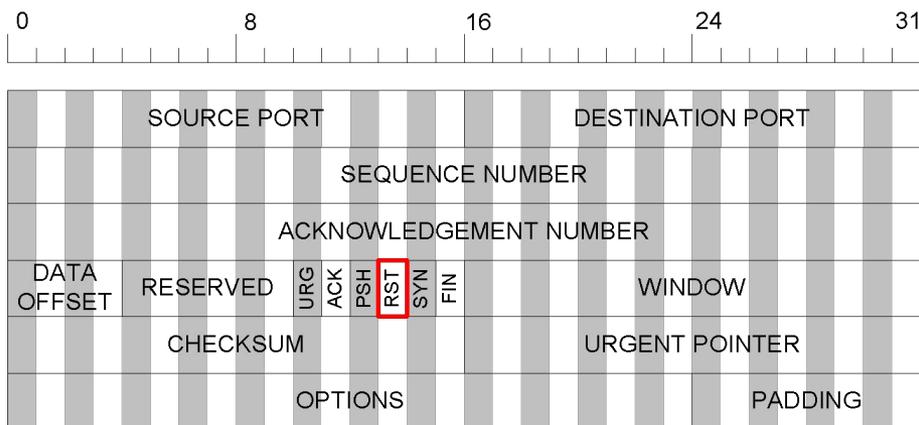
Завершение соединения

CL
OS
E

● 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



Декомпозиция управления
Архитектура TSP/IP
Протокол IP
Транспортные протоколы
Литература

Литература

s•terra

C S P

Cisco Solution Technology Integrator

● Источники разработки

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

КОНТАКТЫ

e-mail: information@s-terra.com

web: <http://www.s-terra.com/>

Тел.: +7 (495) 531 9789

+7 (495) 726 9891

Факс: +7 (495) 531 9789

Вопросы?

Обращайтесь к нам!

s•terra

C S P

Cisco Solution Technology Integrator