ПОДГОТОВКА ЗА ДЪРЖАВЕН ИЗПИТ

Въпрос 8

Компютърни мрежи и протоколи – OSI модел. Канално ниво.Маршрутизация. IP, TCP, HTTP.

Оценяване

Въпрос 8 се състои от 7 подточки:

1)Седемслоен модел OSI на ISO – характеристики на нивата.

2)Канално ниво – кадри, предаване, грешки, номерация, прозорци.

3)Метод на достъп до съобщителната среда в ETEPHET и формат на кадрите.

4)Разпределена маршрутизация: алгоритъм с дистантен вектор и алгоритъм със следене състоянието на връзката.

5)IP протокол – формат на дейтаграмата, адресация, подмрежи и маски.

6)TCP протокол и установяване на съединения.

7)Хипертекстов протокол HTTP.

Оценяване (прод.)

Всяка подточка се оценява поотделно в 6-бална система.

За крайна оценка се взима средно-аритметично:

Кр. Оц. = Σ оц. {1) + 2) + 3) + 4) + 5) + 6) + 7)} : 7

Следват слайдове с най-основни неща по 7-те подточки,

1)Седемслоен модел OSI на ISO – характеристики на нивата.

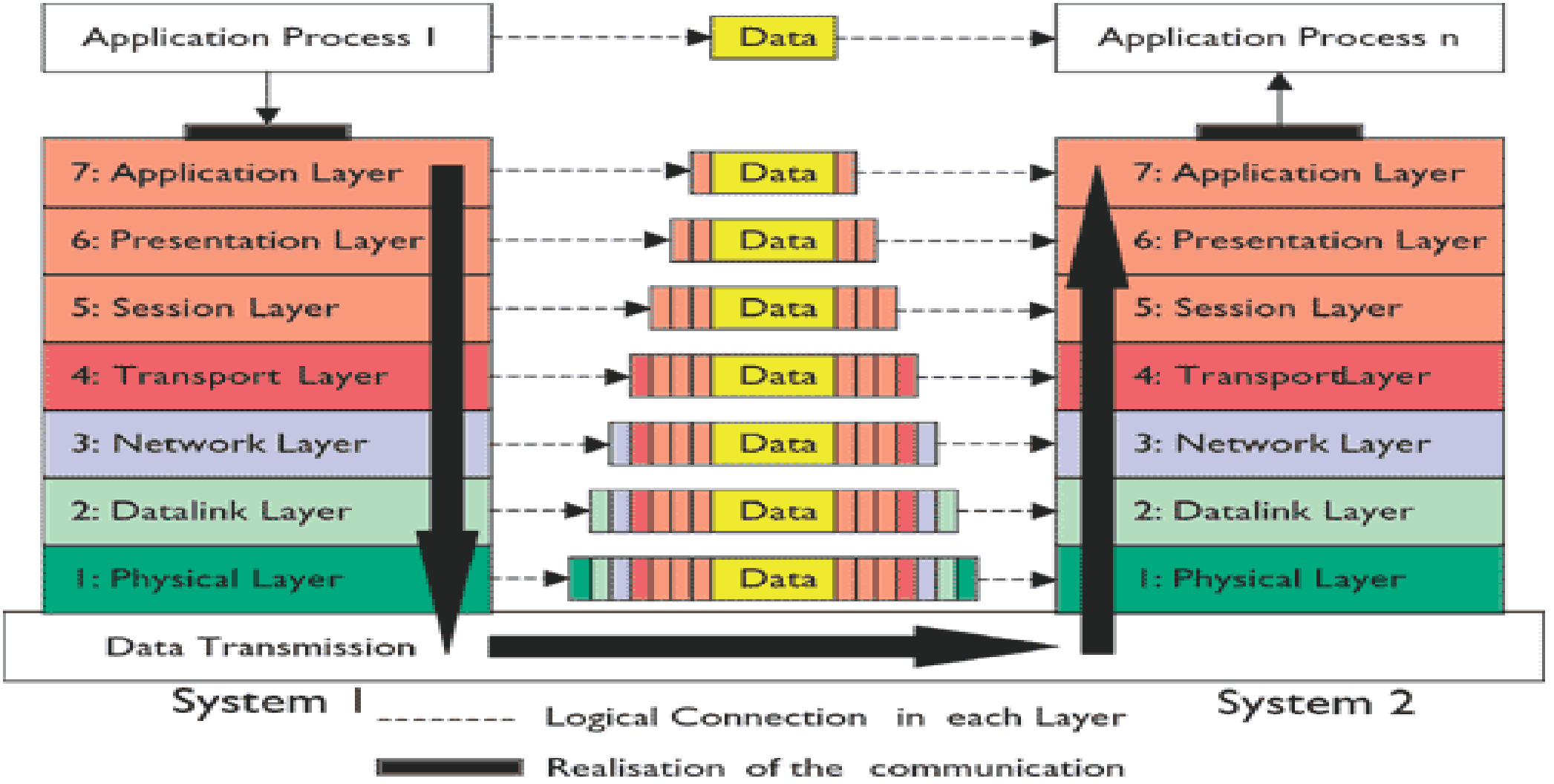
Основен принцип в съвременните мрежови архитектури е принципът за разслояване на функциите по управление на връзките, като всеки слой ползва услугите, предоставени от по-долните слоеве, без да знае как са реализирани тези услуги. Това е принципът на прозрачност.

Слоят *n* на една машина взаимодейства със слоят *n* (на същото ниво) на друга машина. Правилата, по които се осъществява това взаимодействие, се определят от протокола на *n*-то ниво.

Най-общо под **протокол** се разбира съгласувани правила между комуникиращите страни за това как да протича комуникацията.

На практика при комуникацията между съответните слоеве на двете машини не се предават данни. Всеки слой *n* предава данни и контролна информация (header+trailer) на непосредствено по-долния слой *n-1*, докато се достигне най-долния слой *1*, където се осъществява реалната комуникация между машините през физическата среда. В приемника получените данни се разпространяват в обратна посока - от слой *1* нагоре, като всеки слой премахва контролната информация, която се отнася до него. Опаковане и разопаковане (encapsulation – decapsulation).

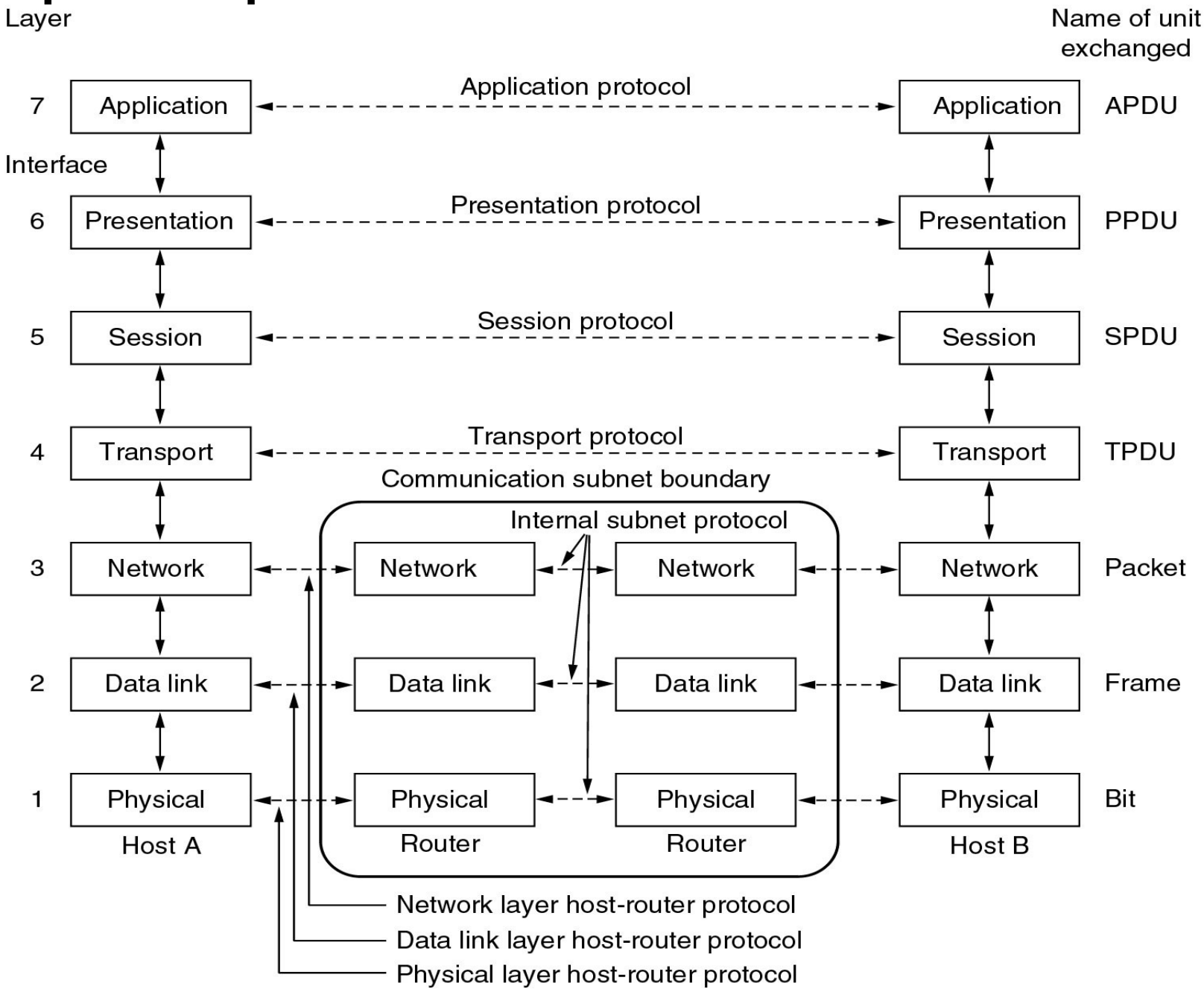
1)Седемслоен модел OSI на ISO – характеристики на нивата



Данните+Контр. Инф. на слой *n* се наричат протоколен блок от данни (PDU). За слой *n-1* PDU(*n*) са си обикновени данни. Чисто потребителските данни – payload.

1)Седемслоен модел OSI на ISO – характеристики на нивата.

The OSI reference model.

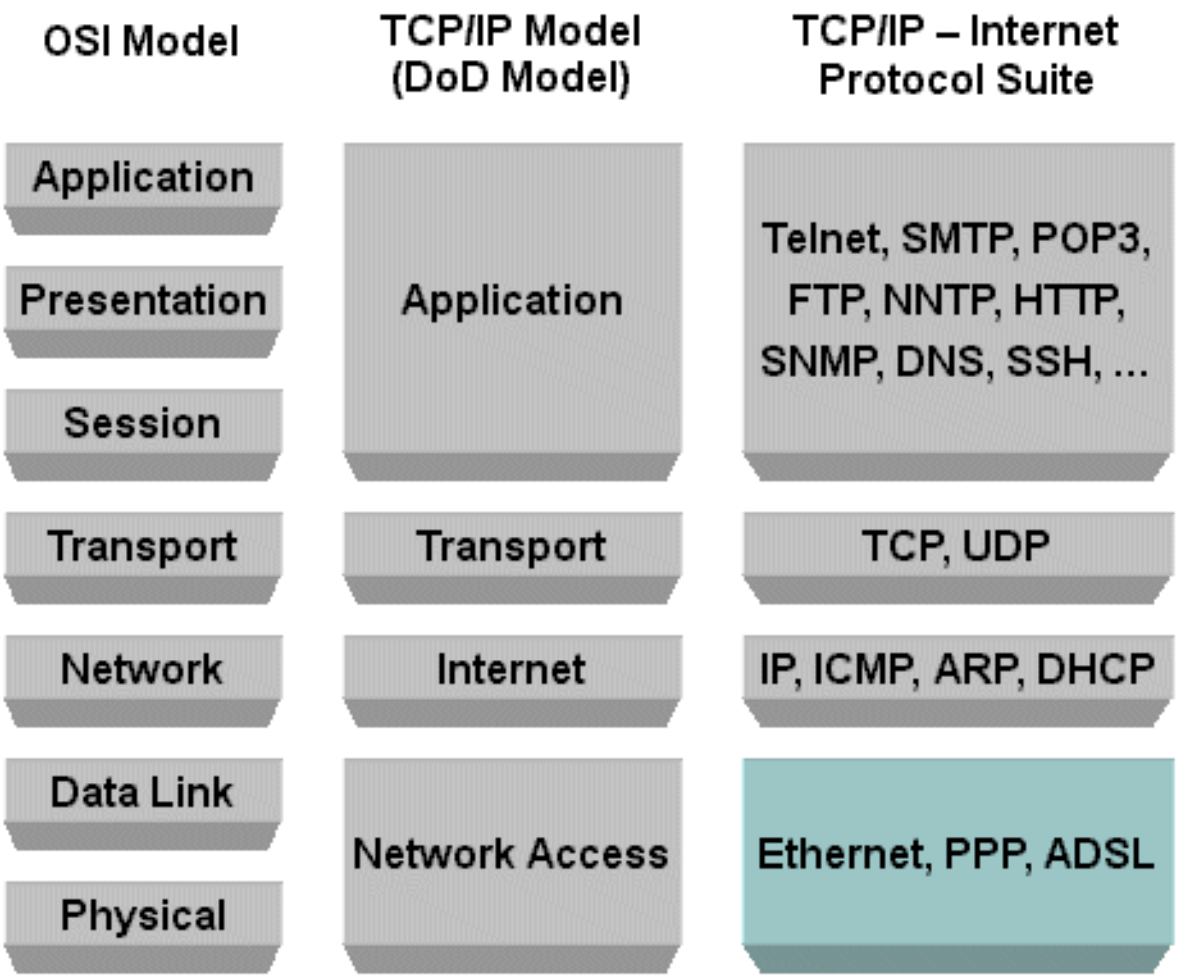


1)Седемслоен модел OSI на ISO – характеристики на нивата.

Накратко описвате основните характеристики на всеки слой.

Какви са разликите с модела TCP/IP, какви са предимствата на последния.

OSI vs. TCP/IP


OSI Model
TCP/IP Model (DoD Model)
TCP/IP – Internet Protocol Suite
Application
Presentation
Session
Application
Telnet, SMTP, POP3, FTP, NNTP, HTTP, SNMP, DNS, SSH, ...
Transport
Transport
TCP, UDP
Network
Internet
IP, ICMP, ARP, DHCP
Data Link
Physical
Network Access
Ethernet, PPP, ADSL

2)Канално ниво – кадри, предаване, грешки, номерация, прозорци.

Каналното ниво има три основни функции:

- да осигури подходящ интерфейс на по-горното мрежово ниво,
- да открива грешки по време на предаването и
- да управлява информационния обмен.

Данните за каналното ниво представляват последователност от **кадри** (frame).

Каналите са три вида - **симплексни**, **полудуплексни** и **дуплексни**.

Дуплексните канали позволяват едновременно предаване в двете посоки.

Полудуплексните канали позволяват предаване и в двете посоки, но в даден момент може да се предава само в една посока.

Симплексните канали позволяват предаване само в една посока.

2)Канално ниво – кадри, предаване, грешки, номерация, прозорци.

Най-общата услуга - прехвърляне на данни (надеждно или best effort, но изчистено от грешки) между мрежовото ниво на източника и мрежовото ниво на получателя (всъщност самото предаване се извършва от физическото ниво, но това остава невидимо за мрежовото ниво).

Основните варианти на тази услуга са:

- непотвърдено, без установяване на сесия (Unacknowledged connectionless service),
- потвърдено, без установяване на сесия (Acknowledged connectionless service) и
- потвърдено, с установяване на сесия (Acknowledged connection-oriented service).

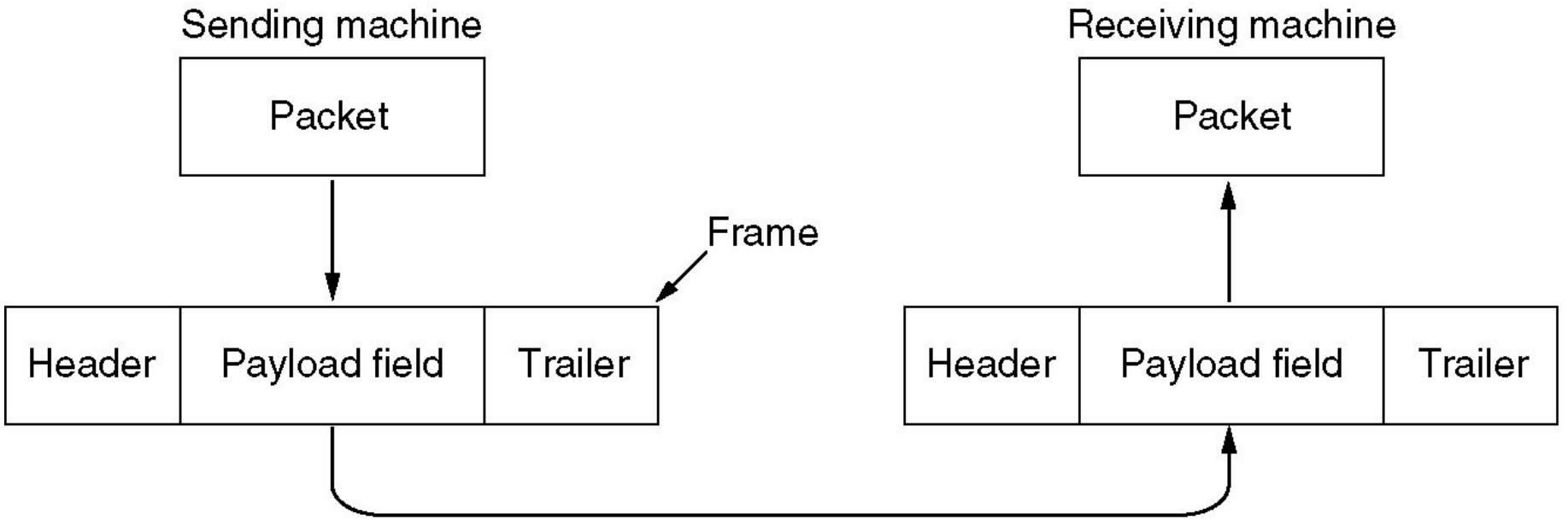
2)Канално ниво – кадри, предаване, грешки, номерация, прозорци.

Друг проблем, който е свързан с управлението на обмена на канално ниво е източникът да изпраща кадри по-бързо, отколкото те могат да бъдат приети от получателя.

За целта се въвеждат механизми за управление на потока от кадри, който осигурява обратна информация на източника за темпа на предаване.

Обикновено механизмите по управление на обмена се изпълняват в транспортния слой над цели масиви от данни, обхващащи последователност от кадри.

Формиране на кадри



Прехвърляне на данни между мрежовите нива на източник и получател (две съсседни машини - възли). Пакети и кадри.

2)Канално ниво – кадри, предаване, грешки, номерация, прозорци.

Вмъкване на байтове

Вмъкване на битове

“Опашка” (trailer) – контролна сума, CRC

Sliding Window Protocols

3)Метод на достъп до съобщителната среда в ЕТЕРНЕТ и формат на кадрите.

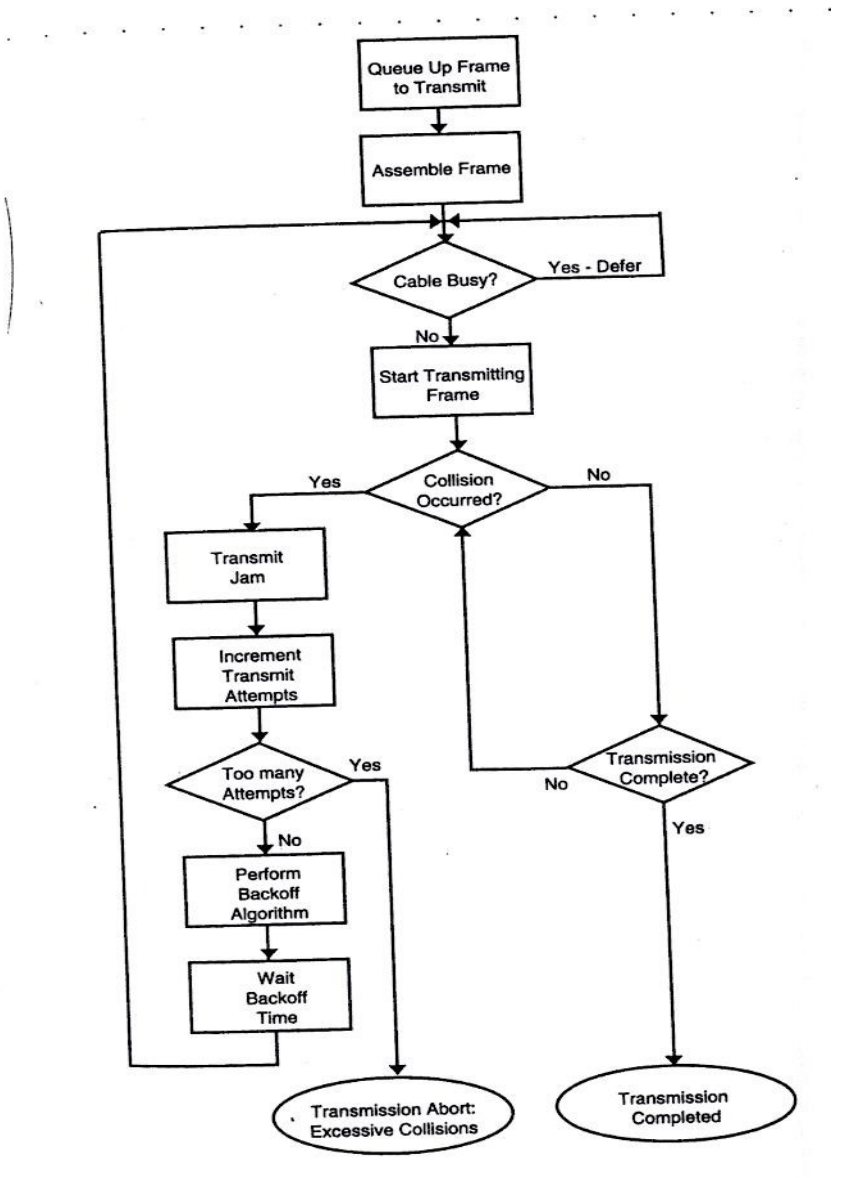
Протоколите (процедурите) за достъп до канала се делят на две основни групи:

- детерминирани и
- състезателни

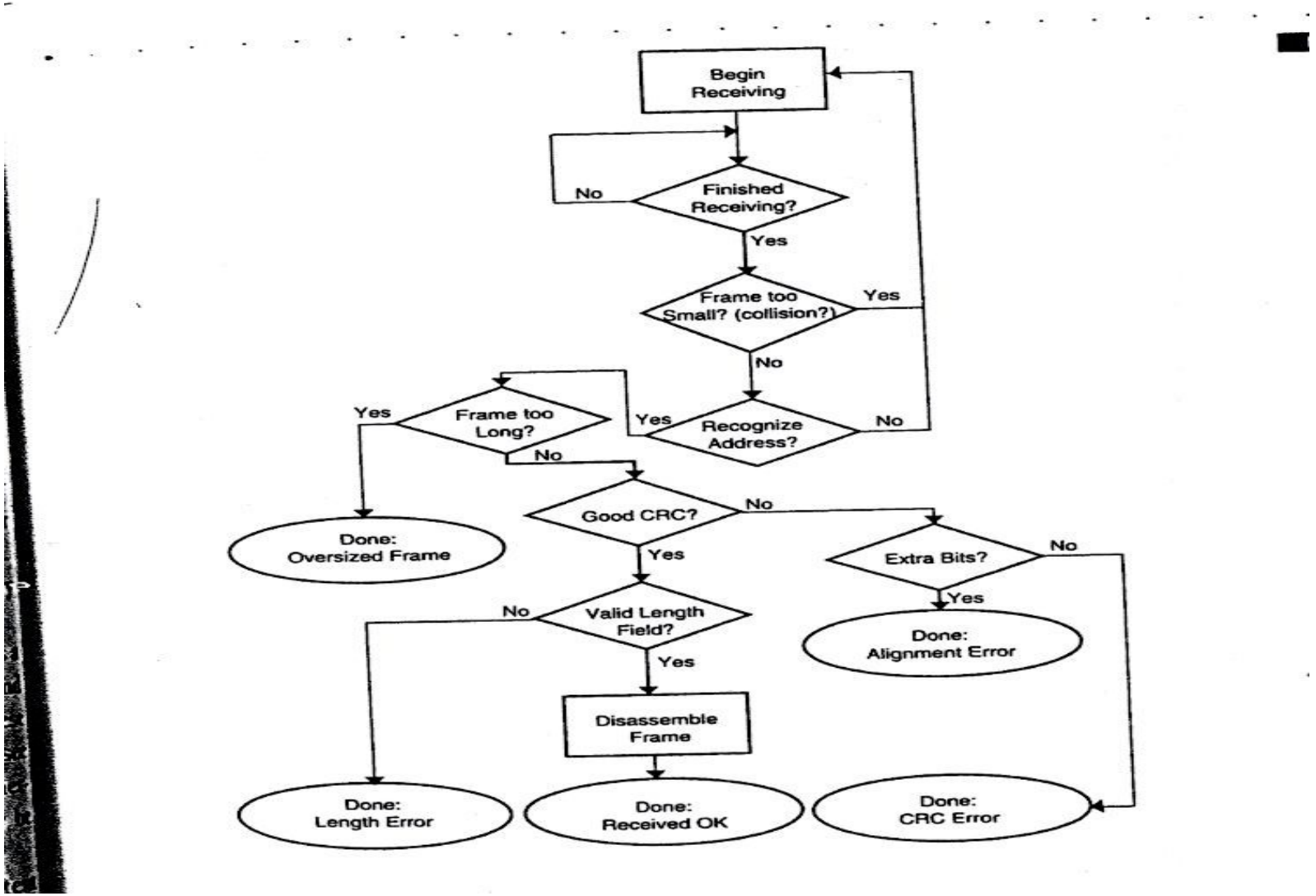
От първите най-известни са Token Ring (разработка на IBM) и FDDI. Те могат да се сравнят с кръгово кръстовище, регулирано със светофари.

Поради сложността им бяха изместени изцяло от състезателните. По-нататък ще се занимаваме с тях.

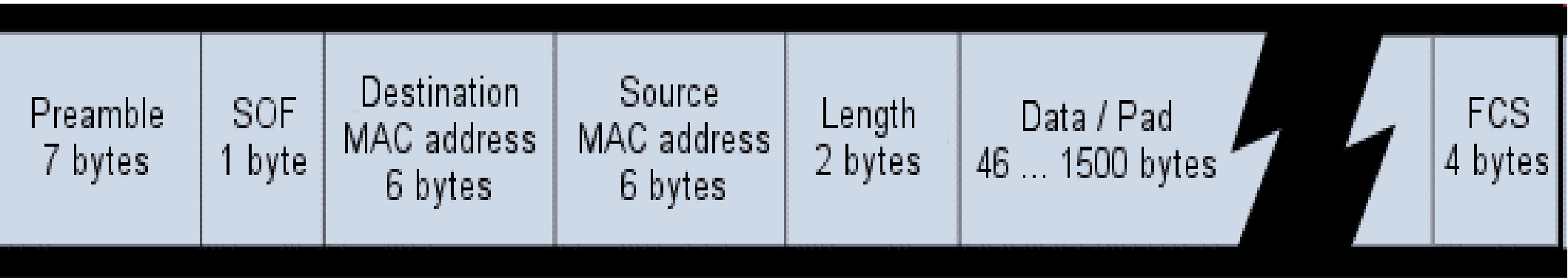
3)Метод на достъп до съобщителната среда в ЕТЕРНЕТ и формат на кадрите.



3)Метод на достъп до съобщителната среда в ЕТЕРНЕТ и формат на кадрите.



802.3 Кадър (Novell raw)



Preamble = 56 бита 0-и и 1-ци.

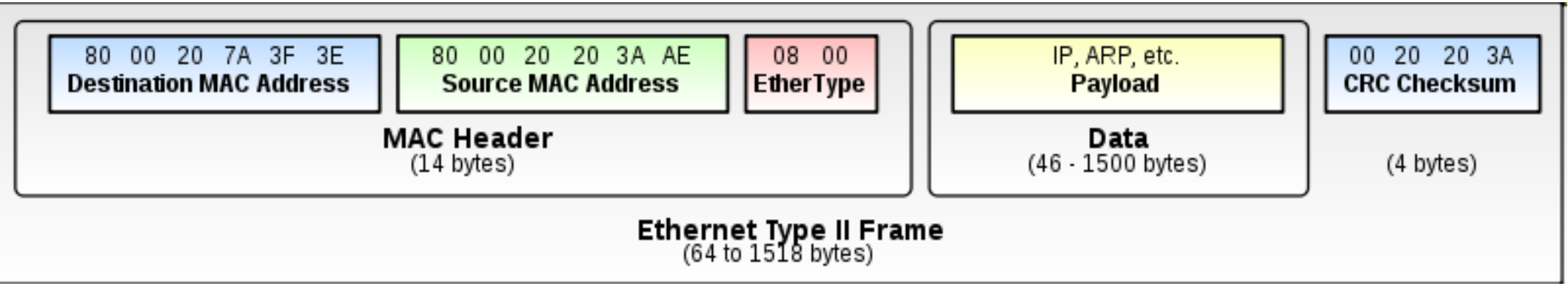
SOF = Start of frame: "10101011"

Data / Pad = ако няма достатъчно данни (payload), полето за данни се допълва, за да имаме минимален размер на кадъра

FCS = Frame check sequence – CRC

Днес се използва **Ethernet II frame**, **DIX** frame (DEC, Intel и Xerox); директно от Internet Protocol.

Ethernet II кадър (DIX)



Destination address съдържа адресът на получателя на кадъра

Source address - адресът на изпращача на кадъра.

Най-младшият бит на най-старшия байт на адреса на получателя е 0 за нормален адрес и 1 за групов адрес. При групов адрес, кадърът е предназначен за група станции (multicast). Адрес на получател, състоящ се само от 1 означава, че кадърът е предназначен за всички станции (broadcast).

Полето *EtherType*: 0x0800 кадърът носи IPv4 дейтаграма; 0x0806 - ARP, 0x8100 - IEEE 802.1Q и 0x86DD - IPv6.

Maximum Transmission Unit (MTU)

В компютърните мрежи **MTU** в протокол на даден слой е максималната дължина на полето за данни (в байтове), който може да понесе дадения слой. Т.е максималния payload.

По-голям MTU означава по-висока ефективност:

- един пакет носи повече потребителски данни;
- по-малко служебна информация (overhead).

Но, по-големите пакети окупират за по-голям период бавните линии. Например, 1500-байтов Ethernet кадър “захваща” за цяла секунда 14.4k модемна линия. Затова се налага фрагментиране.

Ефективност и нетна скорост

$$\text{Efficiency} = \frac{\text{Payload size}}{\text{Frame size}}$$

Максимална ефективност се постига с максимален payload:

$$\frac{1500}{1538} = 97.53\%$$

за untagged ethernet кадри и е $\frac{1500}{1542} = 97.28\%$

за 802.1Q VLAN tagging.

Net bit rate: $\text{Net bit rate} = \text{Efficiency} \times \text{Wire bit rate}$

Максималната нетна скорост за 100BASE-TX Ethernet без 802.1Q is **97.53 Mbit/s**.

MTU. Jumbo Frames.

jumbo frames са Ethernet кадри с дължина по-голяма от 1500 байта payload (MTU). Приема се, че jumbo frames носят до 9000 bytes.

Много, не и всички, Gigabit Ethernet суичове и карти поддържат jumbo frames, но всички Fast Ethernet поддържат само стандартните 1500 байта.

Дължина на Ethernet кадъра от 1518 байта е избрана въз основа на оценка нанадеждността и скоростта на канала.

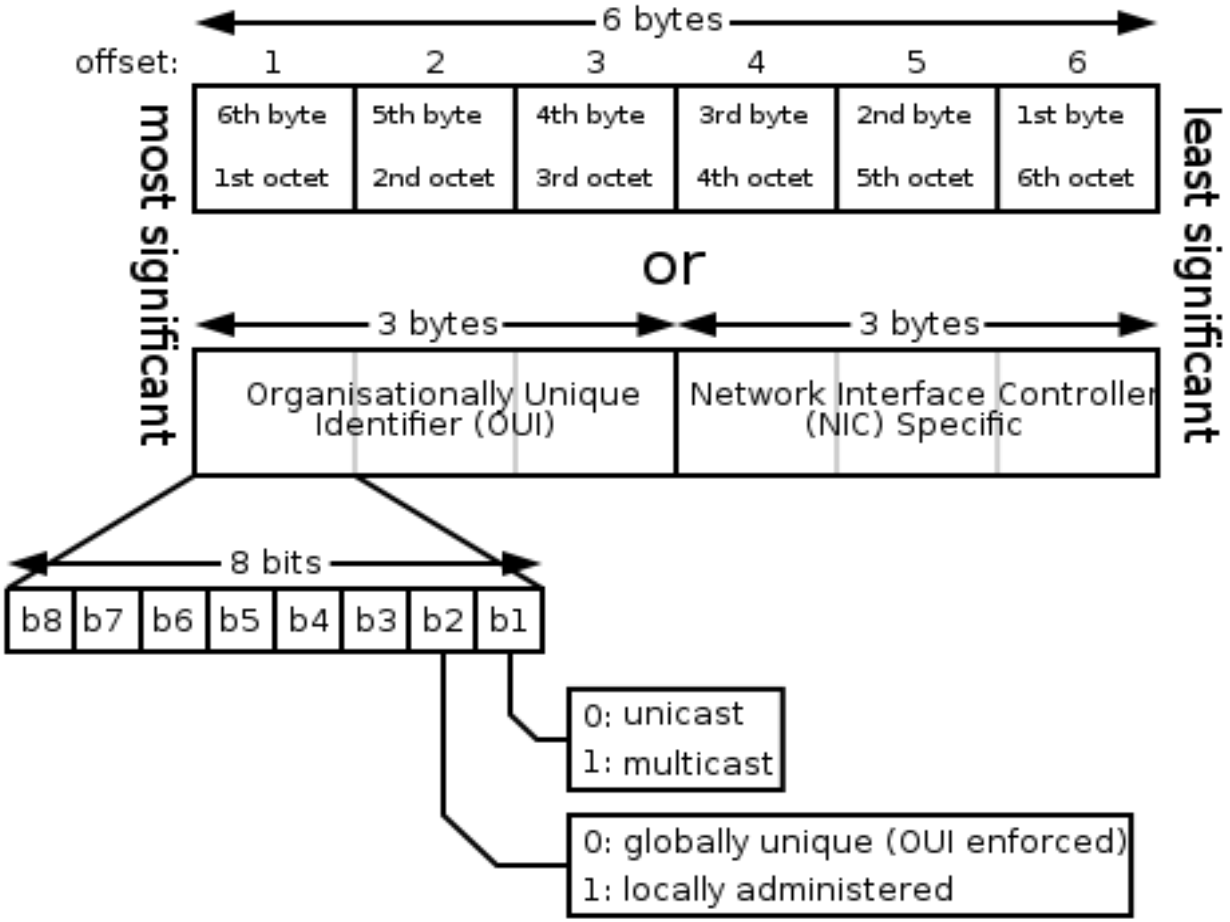
От друга страна, ако увеличим размера, по-големи обеми от данни ще се предадат с по-малко усилия:

- по-малко CPU цикли;

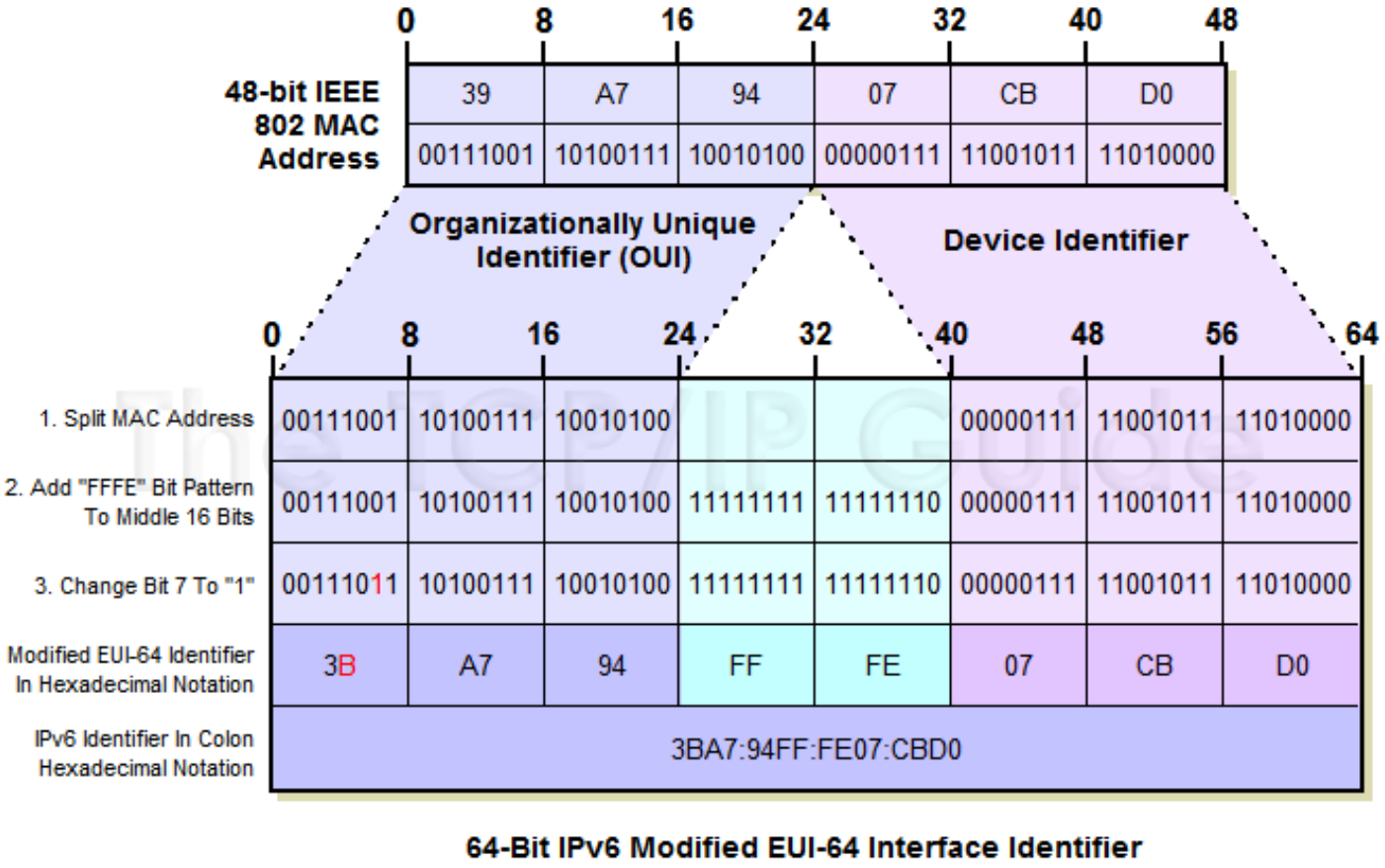
- по-малко прекъсвания;

- CPU се съсредоточава върху потребителските данни.

Формат на MAC адрес



EUI-64 формат



64-Bit IPv6 Modified EUI-64 Interface Identifier

4)Разпределена маршрутизация: алгоритъм с дистантен вектор и алгоритъм със следене състоянието на връзката.

Информацията в един distance vector (DV) е сравнима с пътния знак.

Докато link state (LS) протоколите са пътна карта.

LS рутерът има пълна картина на мрежата и и по-трудно ще вземе неправилно решение.

DV маршрутизират по слухове.

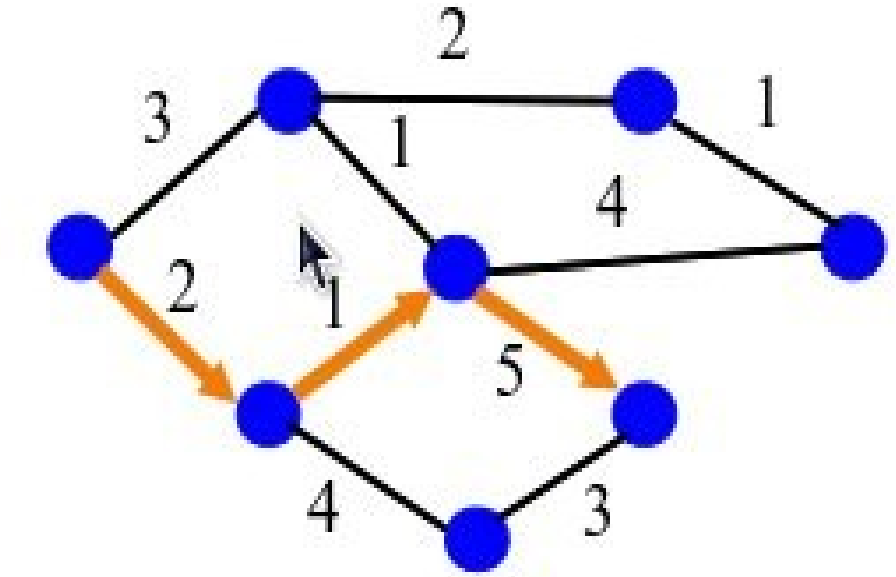
4)Разпределена маршрутизация: алгоритъм с дистантен вектор и алгоритъм със следене състоянието на връзката.

LS – подава на съседите си информация за директно свързаните връзки и състоянието им (затова е link state).

Тази информация се подава от рутер на рутер, всеки я копира, без да я променя. Всеки рутер има идентична информация за мрежата.

Всеки рутер сам за себе си изчислява най-добрите пътища до съответните префикси.

Алгоритъм на Dijkstra



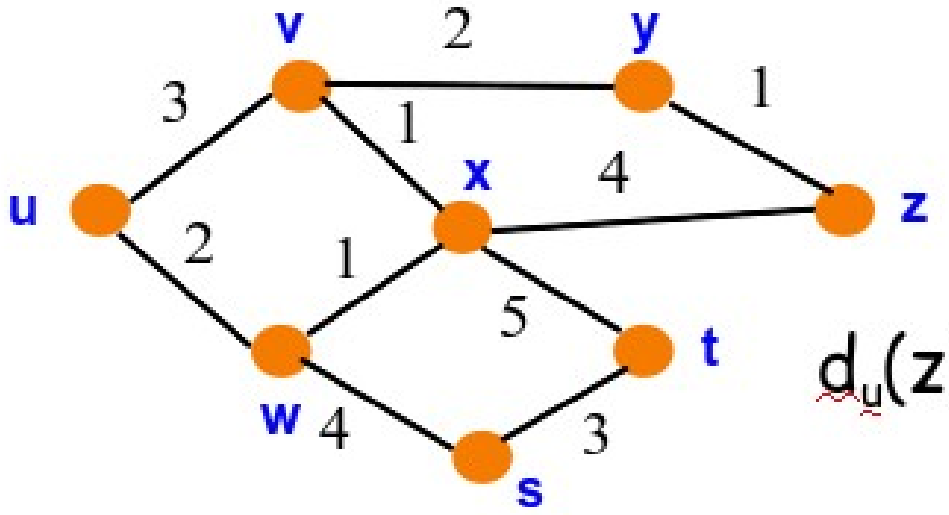
4)Разпределена маршрутизация: алгоритъм с дистантен вектор и алгоритъм със следене състоянието на връзката.

Distance Vector – рутерите се анонсират (рекламират) като вектори:

Посока - адреса на следвашия възел (next hop) и изходящия интерфейс и

Разстояние (метрика), напр. брой възли до дестинацията (hop count).

Алгоритъм Белман-Форд



$$d_u(z) = \min\{c(u,v) + d_v(z),\ c(u,w) + d_w(z)\}$$

$d_v(z)$ – дистантния вектор от v до z

(Всеки възел периодически изпраща до съседите си своя дистантен вектор)

Дистантен вектор

При маршрутизацията с дистантен вектор (distance vector routing) всеки маршрутизатор изгражда и поддържа маршрутна таблица, в която всеки ред съдържа адрес на местоназначение, адрес на следващата стъпка към това местоназначение по най-добрия известен до момента път и дължината на този път (метрика).

Периодически маршрутизаторите изпращат на съседите си цялата или част от маршрутната таблица.

Предимства и недостатъци

DV алгоритмите:

- не товарят процесора и паметта;
- лесни са за реализация и поддръжка;

Но

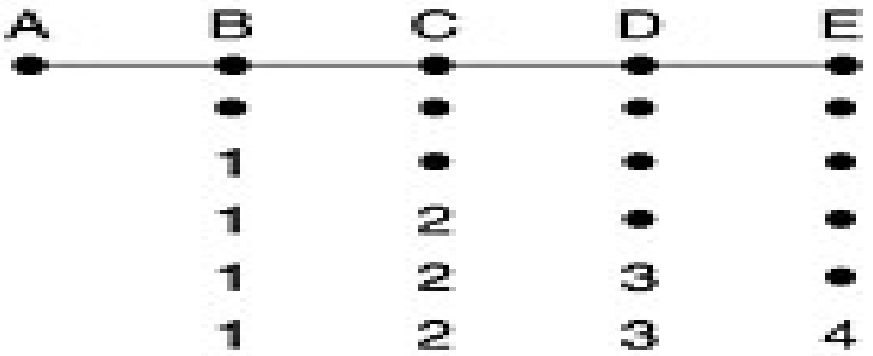
- Периодическите update-и отнемат пропускателна способност от потребителите.

Недостатък

Сериозен недостатък на маршрутизиращите алгоритми с дистантен вектор е **ниската им скорост на сходимост**.

Добрите новини се разпространяват бързо в мрежата, но лошите новини обикновено изискват твърде голям брой периодични съобщения за да достигнат до всички маршрутизатори.

разделяне на хоризонта (split horizon)



LS vs. DV

Размяна на съобщения

LS: с **n** възела, **E** връзки, **O(nE)** изпратени съобщения

DV: разменят се само **м/у съседи**

Скорост на сходимост

LS: висока

DV: ниска
Зацикляния

Проблем “Count-to-infinity”

Стабилност: рутер не е в ред?

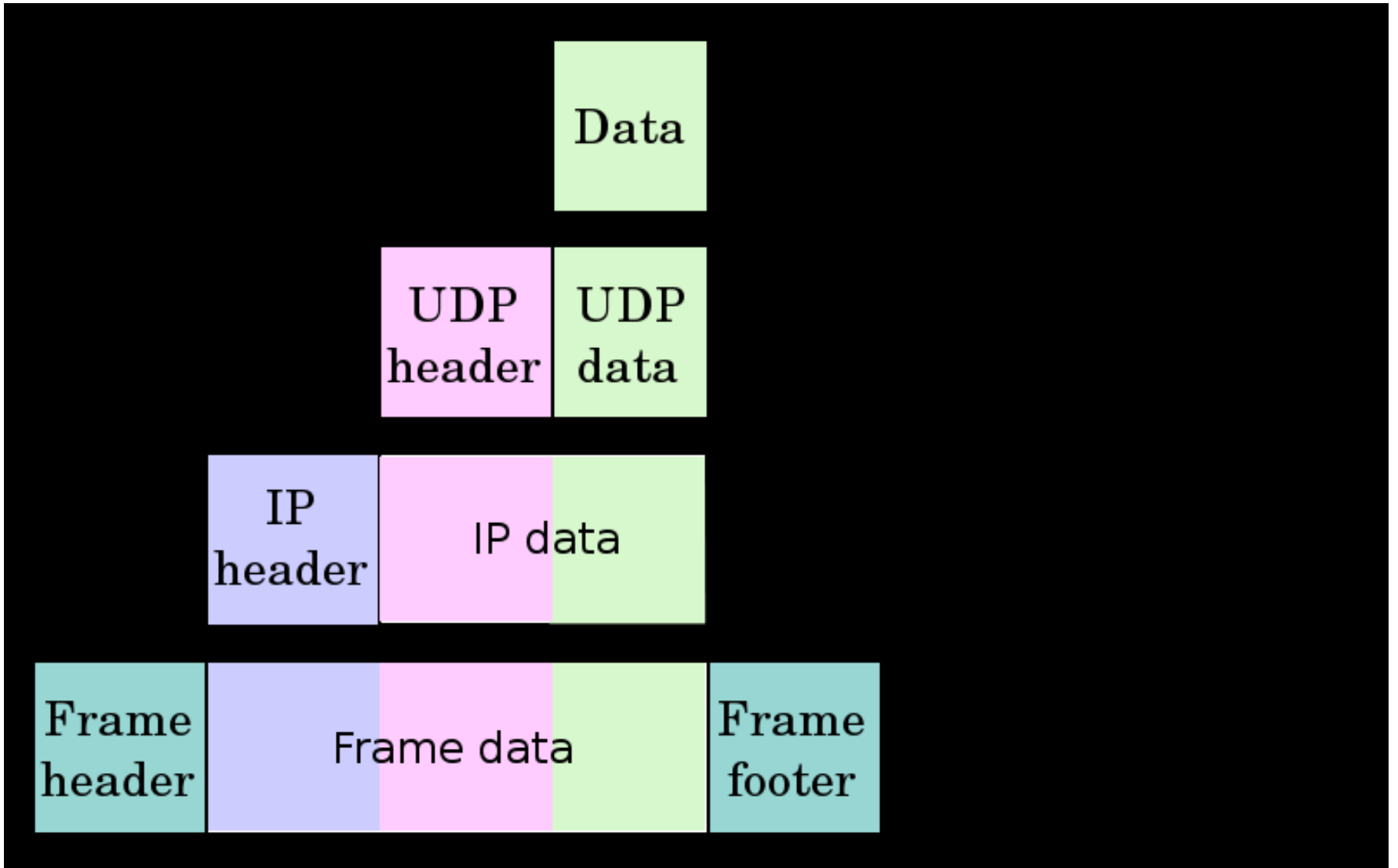
LS:
Възелът рекламира неточна “*link cost*”

Всеки възел изчислява **собствената си таблица**

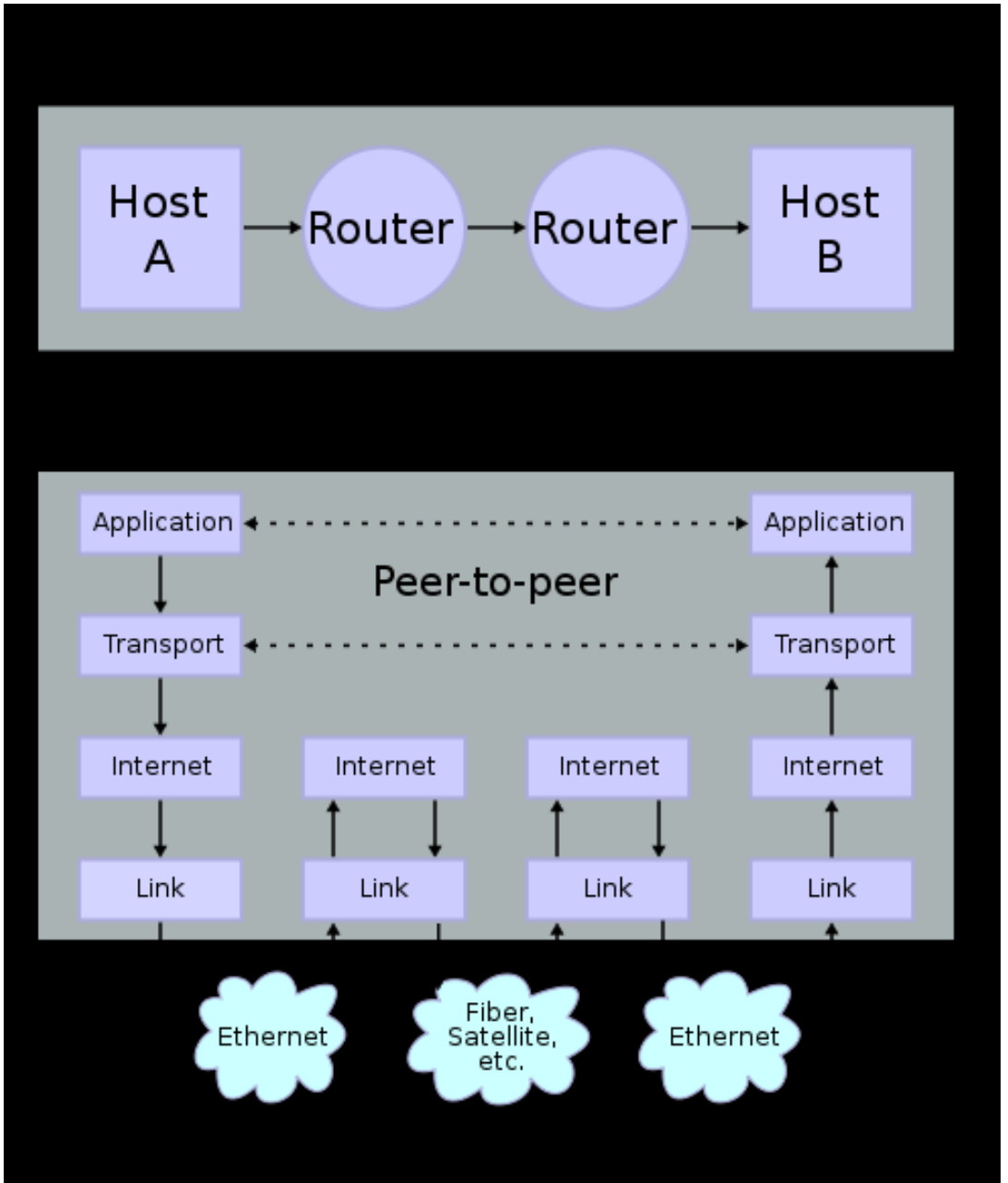
DV:
DV възел рекламира **неточен “*path cost*”**

Таблицата на даден възел се използва от други (**грешката се разпространява**)

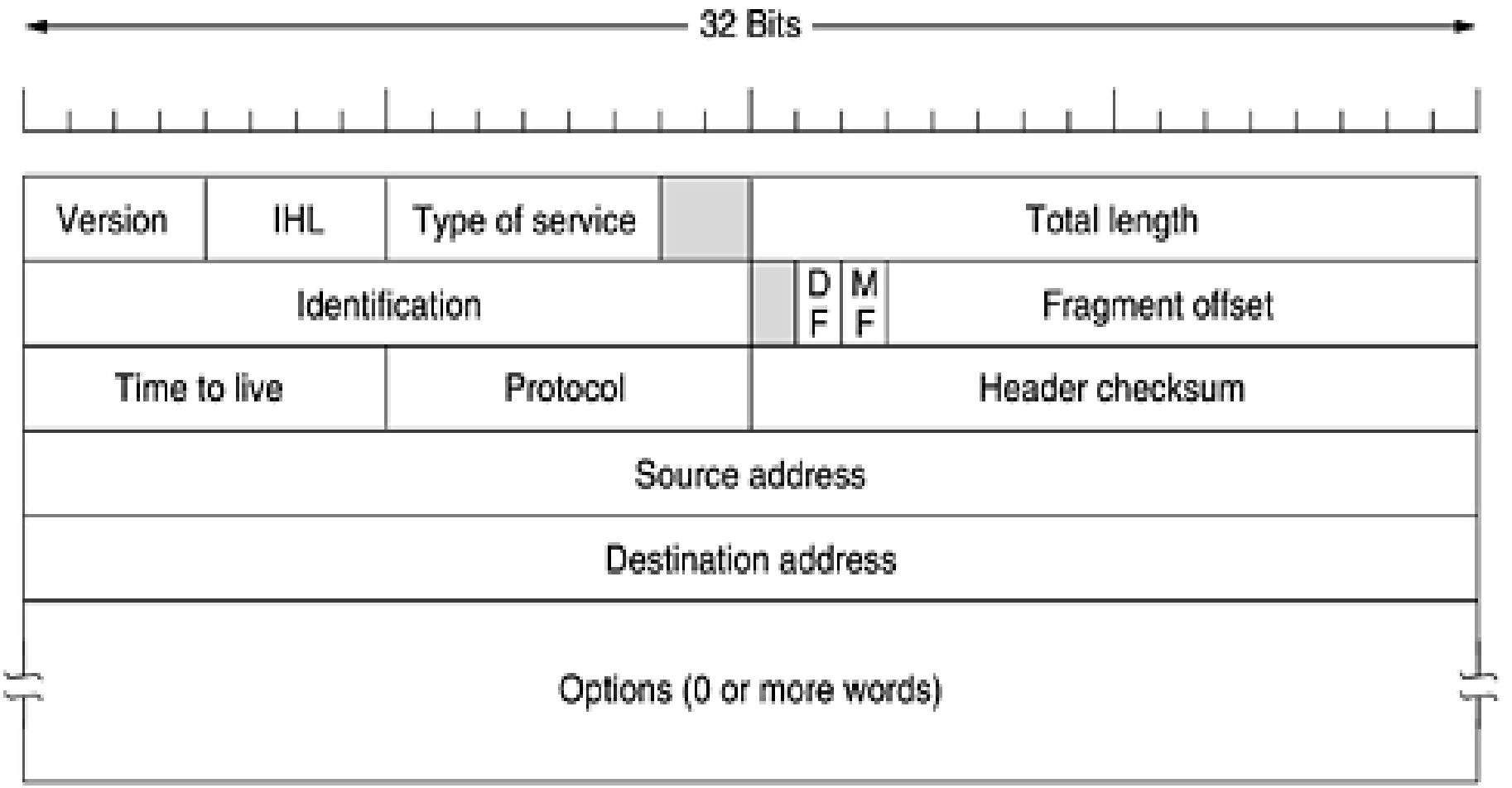
5)IP протокол – формат на дейтаграмата, адресация, подмрежи и маски.



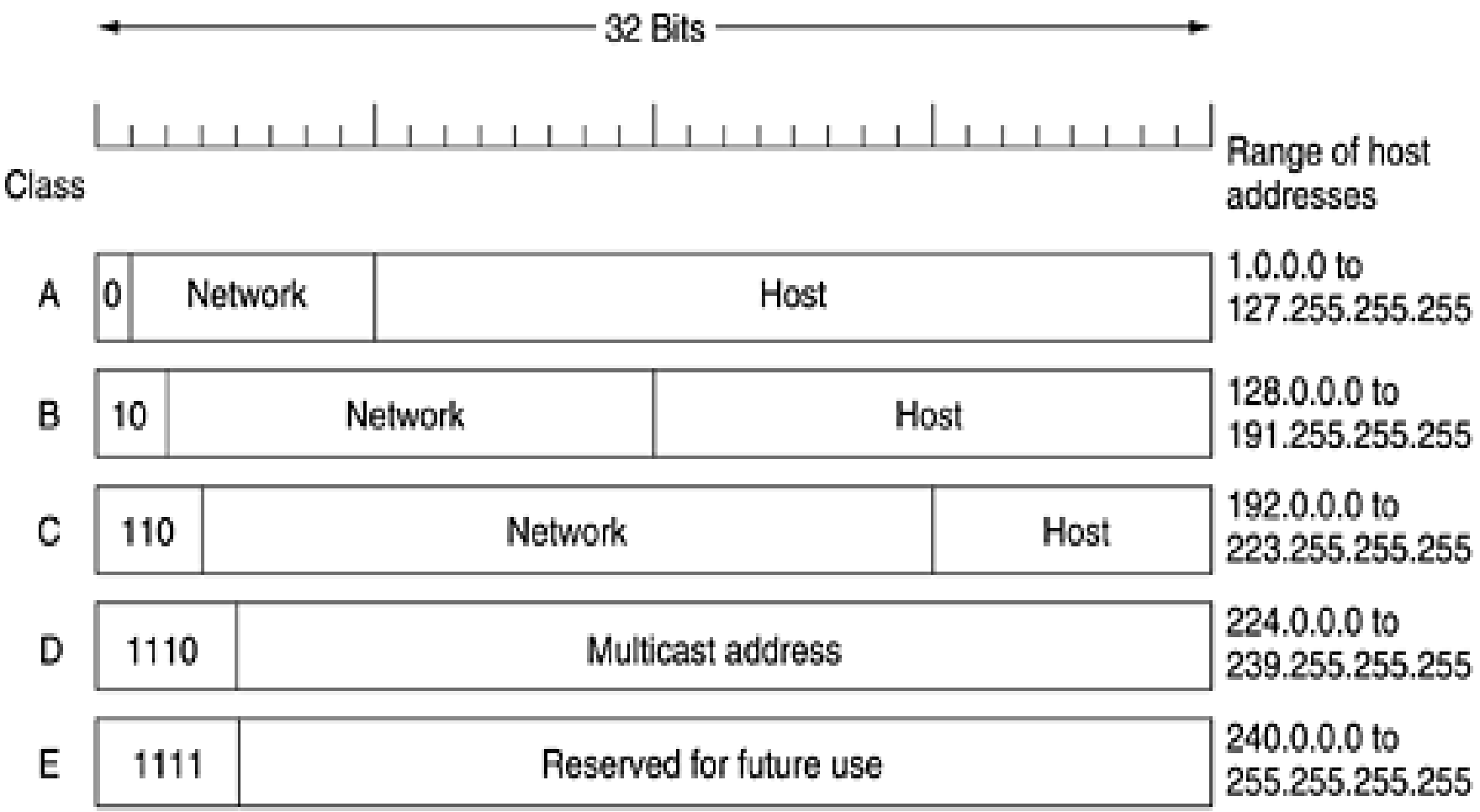
Задачата на IP протокола



Формат на IPv4 пакета



Класове от IP адреси



Записване на IP-адресите

За удобство IP-адресите се изписват в точкова десетична нотация, като всеки от четирите байта се изписва като десетично число от 0 до 255. Най-малкия IP-адрес е 0.0.0.0, а най-големия 255.255.255.255.

Адрес, който съдържа само единици се интепретира като broadcast-адрес, т.е. адресират се всички хостове в дадена мрежа.

Мрежи и подмрежи

Решението на проблема е да се разреши разделянето на една мрежа на **подмрежи**. За целта полето за мрежов номер се разширява надясно, като се отнемат битове от номера на хост.

Например за един адрес от клас B вместо 16 бита за номер на мрежата и 16 бита за номер на хост се използват 22 бита за номер на мрежа, като десните 6 от тях са за номер на подмрежа и 10 бита за номер на хост.

Изписване на маската. Префикси.

SM има същия формат като IPv4 адреса:

старшите битове, които не принадлежат на хост частта са = 1 и се наричат **префикс**,

а останалите (хост частта) са = 0.

Възможни са два начина на изписване на мрежов адрес. Напр., следния клас С адрес:

192.168.1.0 255.255.255.0

или

192.168.1.0/24

Второто означение се нарича 24-битов префикс или просто префикс. **По-нататък ще използваме него.**

Разделяне на класове и безкласово делене

- Около 1993 г. класовете A, B и C е заменено с Classless Inter-Domain Routing (CIDR).
- Преразпределение на Клас A, B и C мрежите така, че се получават по-малки (или по-големи) блокове
- Те се присвояват на ISPs (които са LIR – Local Internet Registries), а те от своя страна ги раздават на своите клиенти (PA – Provider Assigned)...

Classless Inter-Domain Routing

Classless Inter-Domain Routing (CIDR) включва:

- VLSM (variable-length subnet masking) – префикси с произволна дължина. Записва се с /брой на битове (1-ци) в префикса например, 192.168.0.0/16. По-ефективно използване на изчерпващите се IPv4 адреси.
- събиране (**aggregation)** на множество последователни префикси в “супермрежи” (**supernets**), наречено още обобщаване на маршрути - **route summarization**.

CIDR и VLSM

С помощта на VLSM се извършва обобщаване в супермрежи (supernetting) – съкращаване на броя на 1-те от дясно на ляво, което е обратно на деленето на подмрежи (subnetting) - увеличаване на броя на 1-те от ляво на дясно.

Където е възможно в Интернет се анонсират супермрежите, намалявайки броя на “редовете” в глоблната таблица с маршрутите.

Например, 16 последователни Клас С (/24) ще се анонсират като един единствен /20 префикс, респ. маршрут ($2^4 = 16$). Два последователни префикса /20 - като /19 ($2^1 = 2$).

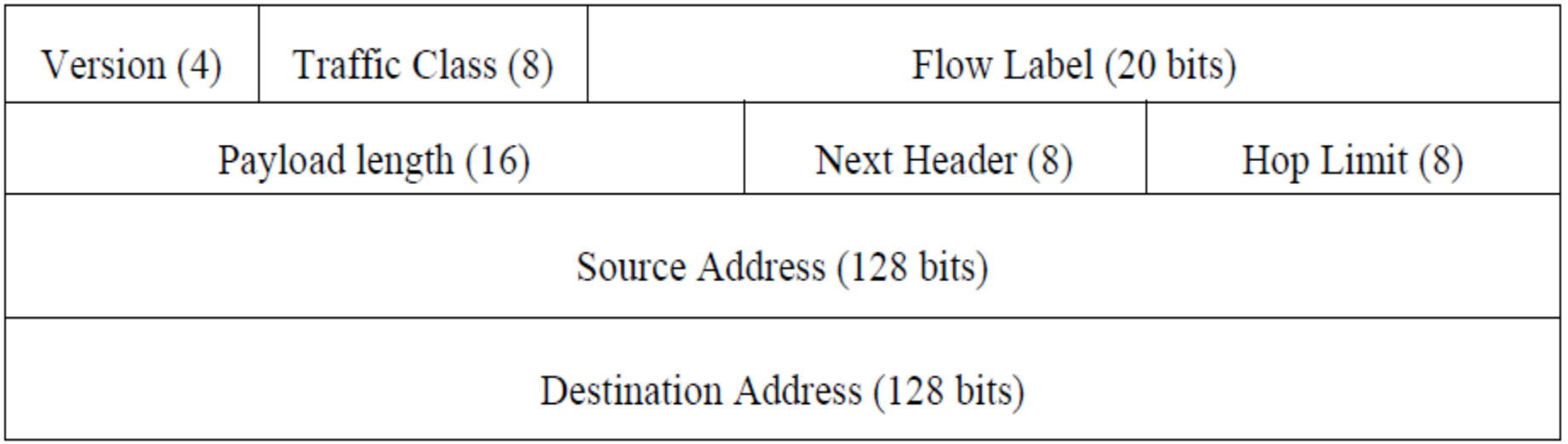
CIDR и VLSM

IP/CIDR	Маска	хост, бр.	Мрежи от Клас
a.b.c.d/32	255.255.255.255	1	1/256C
a.b.c.d/31	255.255.255.254	2	1/128
a.b.c.d**/30**	255.255.255.252	2	1/64 C
a.b.c.d/29	255.255.255.248	6	1/32 C
a.b.c.d/28	255.255.255.240	14	1/16 C
a.b.c.d/27	255.255.255.224	30	1/8 C
a.b.c.d/26	255.255.255.192	62	1/4 C
a.b.c.d/25	255.255.255.128	126	1/2 C
a.b.c.0/24	255.255.255.0	254	1 C

CIDR и VLSM

a.b.c.0/23	255.255.254.0	510	2 C
a.b.c.0/22	255.255.252.0	1022	4 C
a.b.c.0/21	255.255.248.0	2046	8 C
a.b.c.0/20	255.255.240.0	4094	16 C
a.b.c.0/19	255.255.224.0	8190	32 C
a.b.0.0/16	255.255.0.0	65534	256 C = 1 B
a.b.0.0/15	255.254.0.0	131070	2 B
a.0.0.0/8	255.0.0.0	16777214	256 B=1 A
0.0.0.0/0	0.0.0.0	4294967296	256 A

IPv6



traffic class (заменя IPv4 ToS);

flow label (ново QoS management);

payload length (до 64KB);

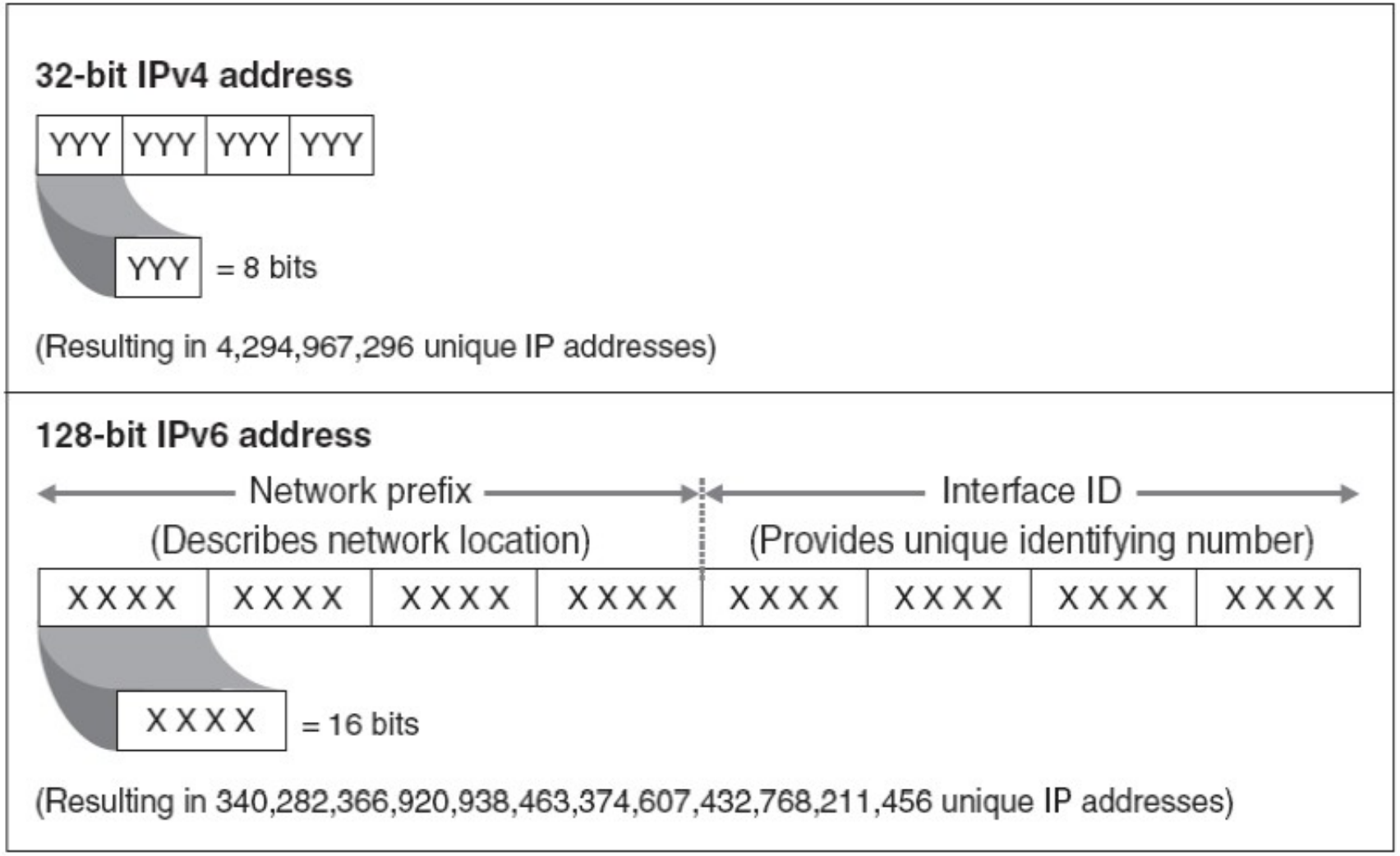
next header (заменя IPv4 protocol);

hop limit (заменя IPv4 TTL).

IPv4 vs. IPv6

Property	IPv4	IPv6
Address size and network size	32 bits, network size 8-30 bits	128 bits, network size 64 bits
Packet header size	20-60 bytes	40 bytes
Header-level extension	limited number of small IP options	unlimited number of IPv6 extension headers
Fragmentation	sender or any intermediate router allowed to fragment	only sender may fragment
Control protocols	mixture of non-IP (ARP), ICMP, and other protocols	all control protocols based on ICMPv6
Minimum allowed MTU	576 bytes	1280 bytes
Path MTU discovery	optional, not widely used	strongly recommended
Address assignment	usually one address per host	usually multiple addresses per interface
Address types	use of unicast, multicast, and broadcast address types	broadcast addressing no longer used, use of unicast, multicast and anycast address types
Address configuration	devices configured manually or with host configuration protocols like DHCP	devices configure themselves independently using stateless address autoconfiguration (SLAAC) or use DHCP

IPv4 vs. IPv6


32-bit IPv4 address
YYY
YYY
YYY
YYY
YYY
= 8 bits
(Resulting in 4,294,967,296 unique IP addresses)
128-bit IPv6 address
Network prefix
(Describes network location)
Interface ID
(Provides unique identifying number)
XXXX
XXXX
XXXX
XXXX
XXXX
XXXX
XXXX
XXXX
XXXX
= 16 bits
(Resulting in 340,282,366,920,938,463,374,607,432,768,211,456 unique IP addresses)

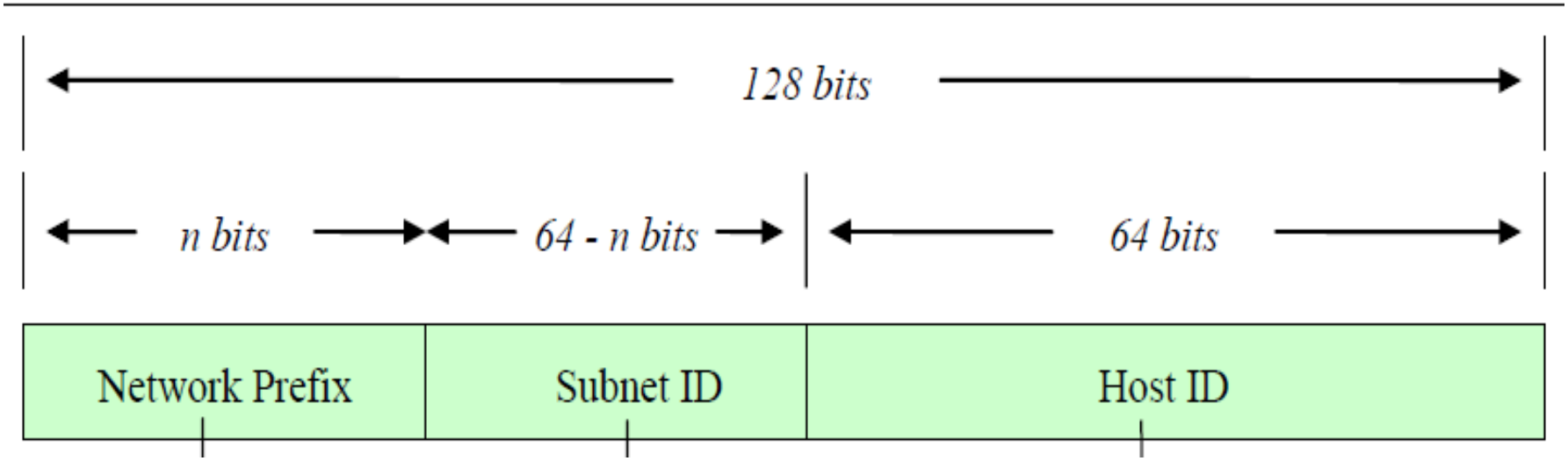
IPv6 адресиране

IPv6 адрес (пример):

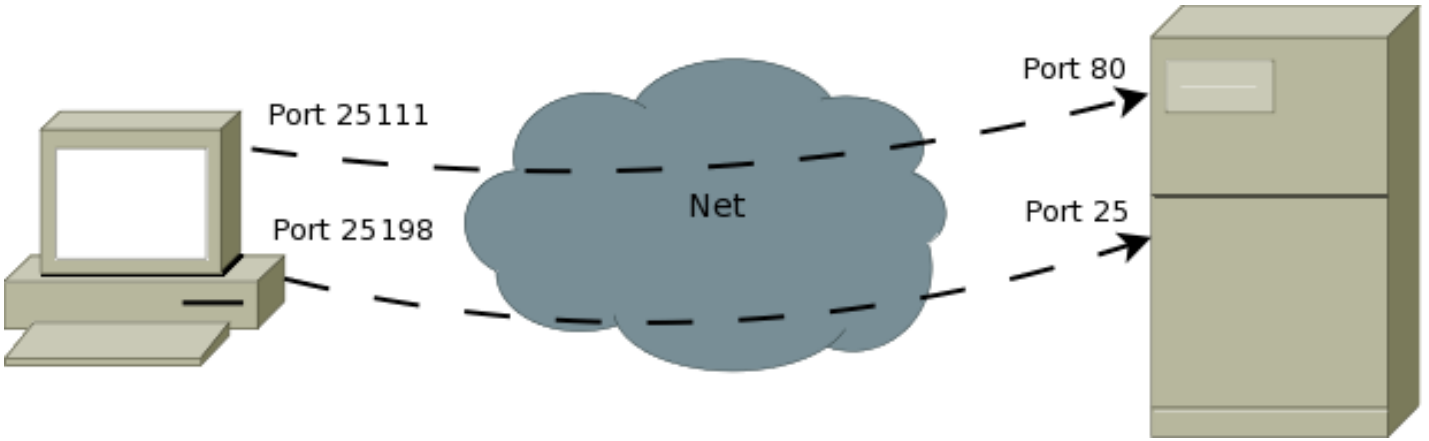
2001:0db8:9095:02e5:0216:cbff:feb2:7474

8 групи с по **4** шестнадесетични числа

8 * 4 * 4 = 128



6)TCP протокол и установяване на съединения.



Протоколи със съединение

При протоколите със съединение имаме три фази – установяване на съединение, трансфер на данни и закриване на съединението.

Транспортното ниво трябва да предоставя интерфейс на приложното ниво.

Примерен интерфейс за надеждно обслужване се състои от следните примитиви LISTEN, CONNECT, SEND, RECEIVE, DISCONNECT.

Да предположим, че имаме приложение с един сървър и много отдалечени клиенти.

Портове и сокети

Когато един процес от приложното ниво иска да се свърже с друг процес, той трябва да може да го идентифицира.

Методът е да се присвоят адреси на процесите, които слушат за създаване на връзка.

В Internet тези адреси се наричат **портове** и са 16-битови номера.

Комбинацията от IP адрес и порт се нарича socket и той уникално идентифицира процес, който се изпълнява на някакъв хост.

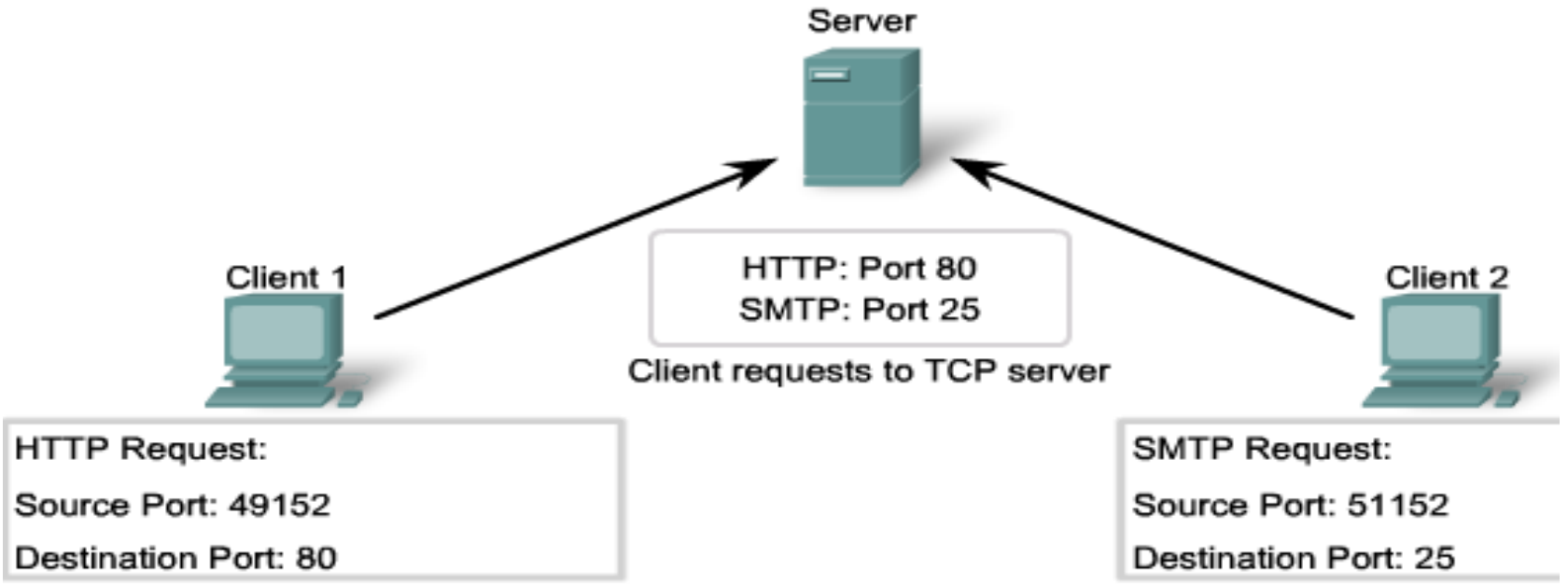
Transmission Control Protocol

Transmission Control Protocol (**TCP**) осигурява надеждно, подредено доставяне на потока от байтове от програма на един компютър до програма на друг компютър.

TCP следи за размера на съобщенията, скоростта на обмен и натоварването на мрежовия трафик.

TCP процеси на сървъра

На една и същ машина, сървър, не е възможно да има две услуги, присвоени на един и същ номер на порт.



Структура на TCP сегмента

Bit offset	Bits 0–3	4–7	8–15								16–31
0	Source port										Destination port
32	Sequence number										
64	Acknowledgment number										
96	Data offset	Reserved	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size
128	Checksum										Urgent pointer
160	Options (optional)										
160/192+	Data										

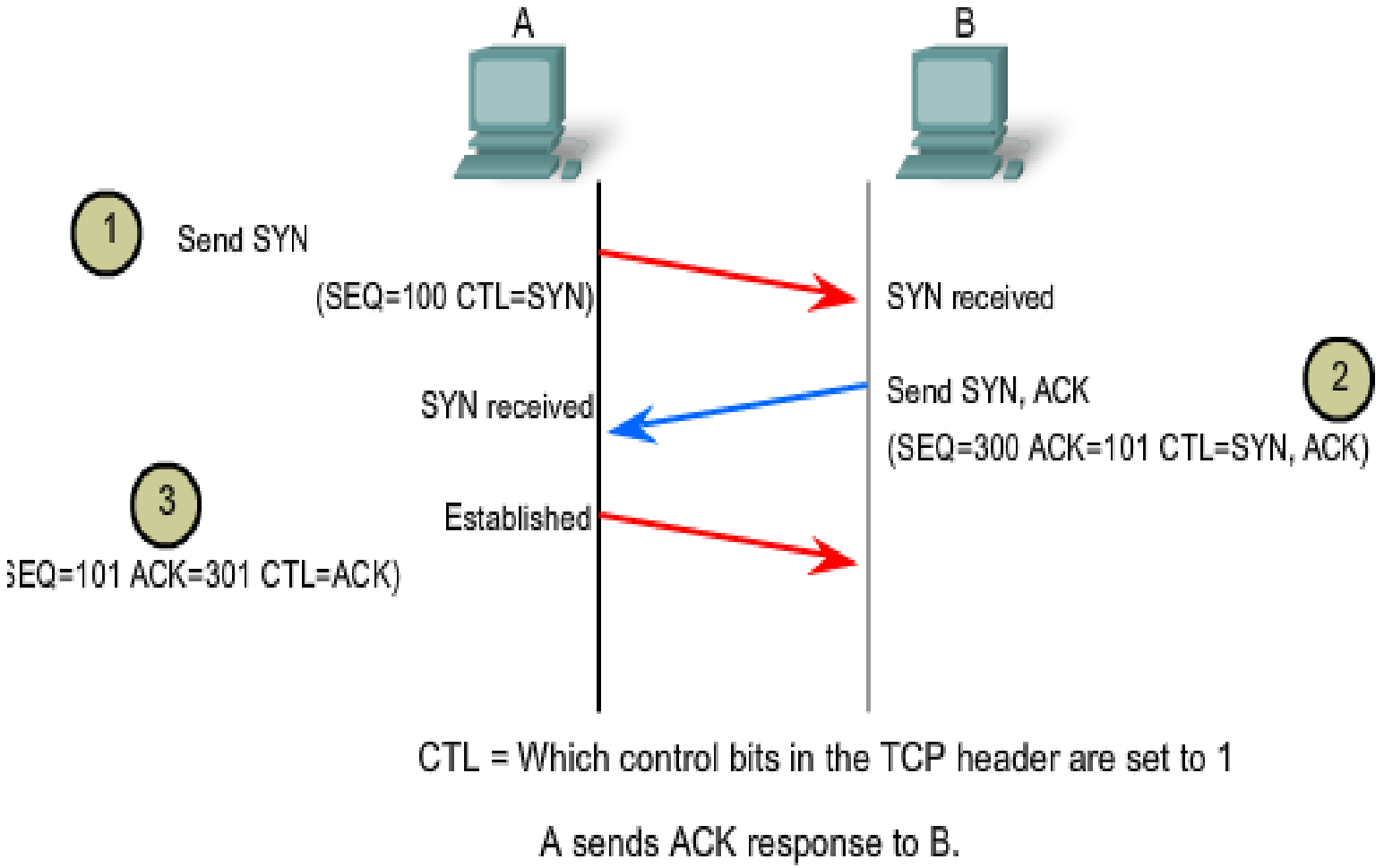
3-way handshake

1. Клиентът-инициатор изпраща сегмент SYN, съдържащ начална стойност на последователността SEQ, и представляващ заявка за начало на сесия.

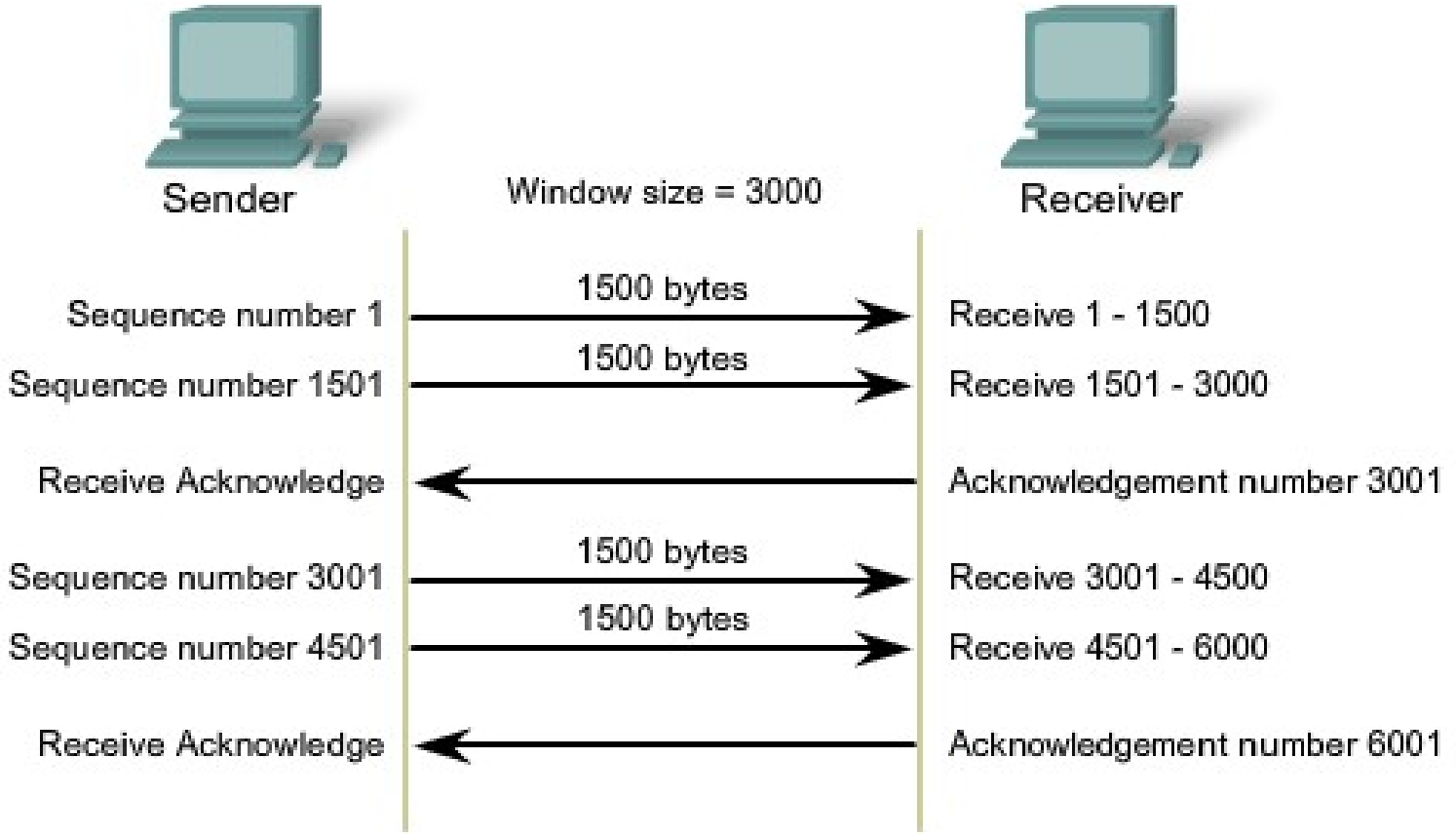
2. В отговор сървърът изпраща сегмент, съдържащ стойност за потвърждение ACK (SEQ + 1). Освен това - собствената си синхронизираща стойност на последователност SYN SEQ, която е по-голяма от получения и потвърден номер ACK – следващия очакван байт.

3. Клиентът-инициатор отговаря със стойност ACK, равна на (получена SEQ + 1). С това съединението е установено.

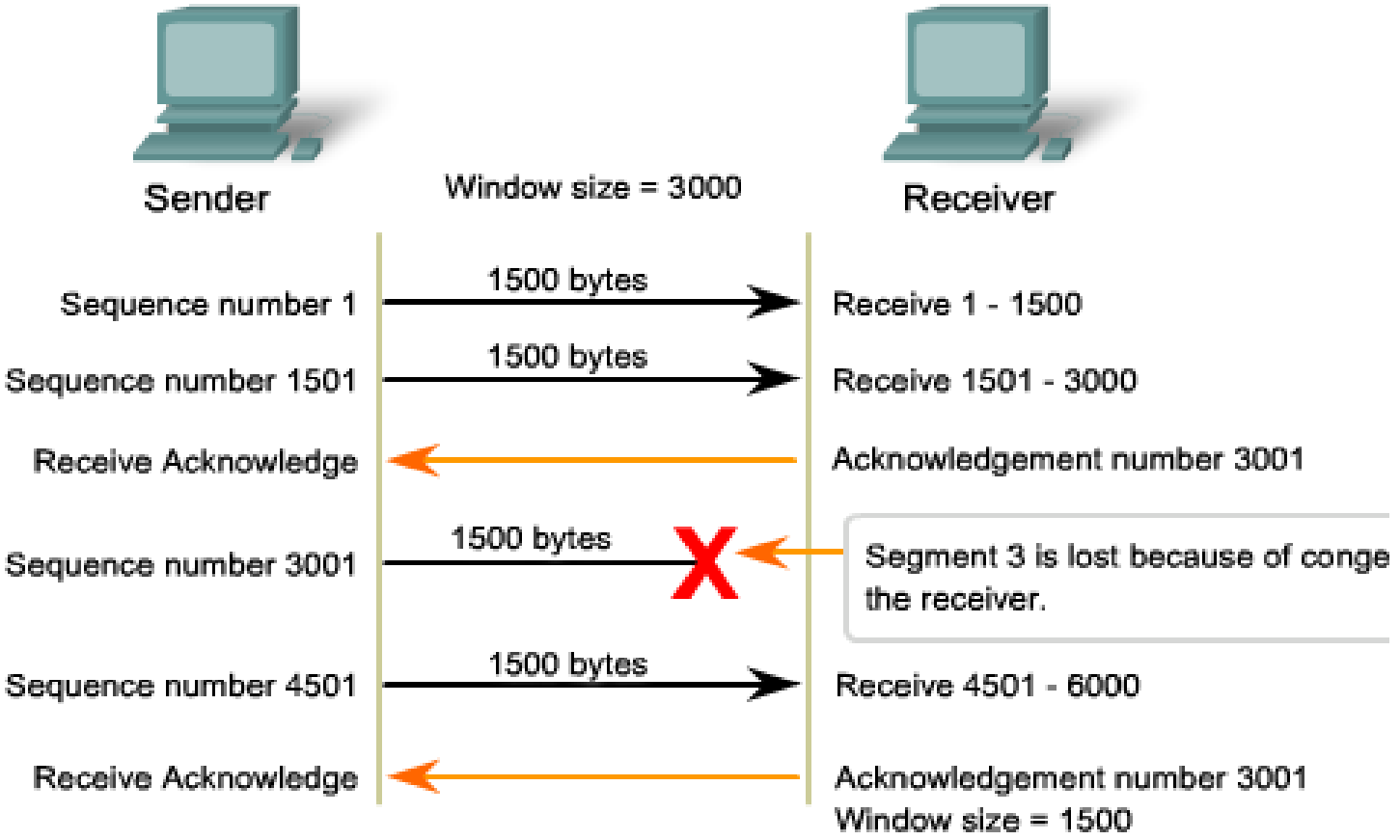
3 стъпки на TCP съединение



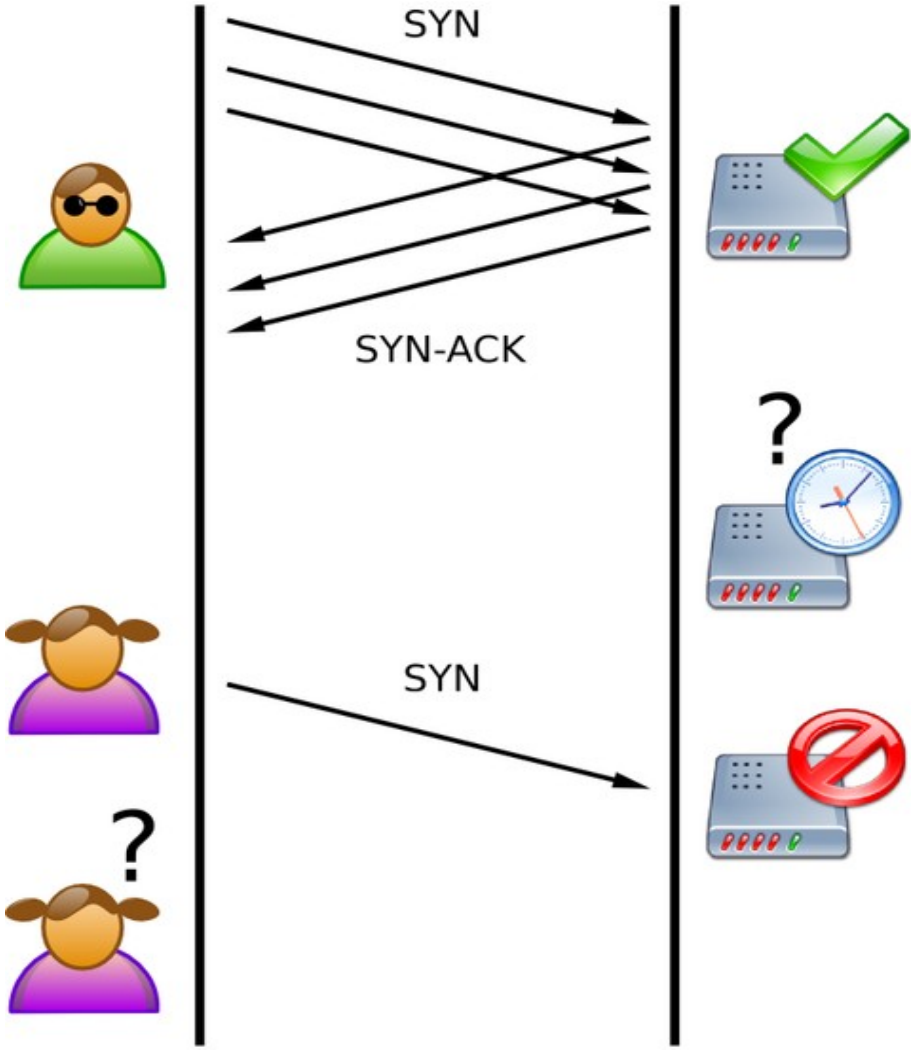
TCP. Управление на потока и задръстванията.



TCP. Редуциране на загубите.



SYN flood атаки



Атакуващият изпраща няколко SYN, но не връща ACK

В резултат: полуотворени конекции. Няма място за легитимния потребител,

7)Хипертекстов протокол HTTP

Хипертекстът се оформя като съвкупност от страници.

Всяка страница си има уникално URL – уникален адрес, който еднозначно указва местоположението на страницата в Internet.

Хиперлинк – под част от текста стои URL на друга страница.

Т.е хипертекстовите страници съдържат препратки към други хипертекстови страници.

Не се изисква тези страници да се намират на същия сървър.

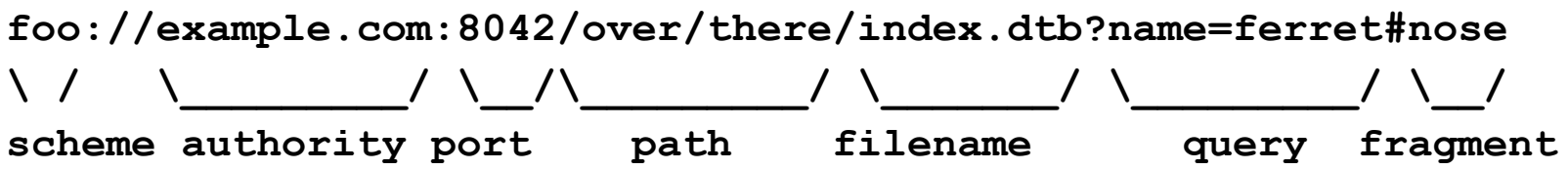
URL

URL съдържа протокол, име на домейн и пътя на страницата върху диска на сървъра.

При осъществяване на връзка между browser-а и сървъра по URL-то първо се търси по името на сървъра, а после по пътя върху диска на сървъра.

Страницата физически се изтегля от сървъра, предава се на browser-а и той я изобразява.

Една страница се прехвърля в рамките на една HTTP-сесия.



HTTP - request-response protocol

HTTP – request-response (заявка-отговор) протокол.

Клиентът отправя съобщение с HTTP заявка към сървъра.

Сървърът (съдържание, ресурси - HTML файлове и др.) връща съобщение с отговор:

- информация за състоянието - хедъра и заявеното съдържание – в тялото.

Версия 1.0 на HTTP

При първите версии на HTTP протокола за изпълнението на всеки метод се прави отделна HTTP сесия.

Тя отваря TCP съединение, праща нещо, след това затваря последователно съединението и сесията.

Имаме една сесия, едно съединение.

Не е ефективно

Версия 1.1 на HTTP

При версия 1.1 на HTTP тези недостатъци са преодолени:

- на едно TCP съединение отговарят няколко сесии (няколко команди).
- различни хипертекстови страници, но с едно TCP съединение.

Създава се „канал“ в пълен дуплекс: в едната посока - заявки, а в обратната – отговори.

HTTP сесия

HTTP сесията е последователност от request-response транзакции.

HTTP клиентът инициира заявка. Установява съединение на порт 80/TCP.

HTTP сървърът "слуша" на порт 80 за съобщение със заявка от клиент.

При получаване на заявка сървърът връща "HTTP/1.1 200 OK" и съобщение, което съдържа:

- търсения ресурс или
- съобщение за грешка.

HTTP методи

HTTP дефинира девет метода (наричани и "verbs" - глаголи), които показват какво действие да се изпълни върху желания ресурс.

Ресурсът е файл или данни, получени от изхода на програма на сървъра.

HTTP Secure

Hypertext Transfer Protocol Secure (**HTTPS**) е:

HTTP с SSL/TLS

за криптирани комуникации и сигурна идентификация на web сървър.

HTTPS се използва за е-разплащания и други поверителни операции

(напр. СУСИ: https://susi4.uni-sofia.bg/)