

# Канално ниво в локалните мрежи

6. Методи на достъп до  
съобщителната среда в Ethernet.

7. Управление на канала в Ethernet.

Превключватели и мостове.

Виртуални локални мрежи и протокол  
Spanning Tree.

# Какво ще научим

- ✓ Втори слой в мрежовата архитектура по отношение на локалните мрежи. Методи на достъп до преносната среда.
- ✓ Логическа и физическа топология на локална мрежа
- ✓ История на възникване на метода на достъп CSMA/CD
- ✓ Ethernet – технологичната конвергенция в Internet
- ✓ Формат на кадъра в Ethernet. MAC адрес.
- ✓ MTU и производителност на мрежата
- ✓ Жична преносна среда в Ethernet. 10/100/1000 Mbps, 10/40/100 Gbps и по-високи скорости
- ✓ От хъбове към свичове. Технология на превключването.
- ✓ Протокол Spanning Tree. Виртуални локални мрежи (VLANs).
- ✓ Ethernet в глобалните мрежи

# LANs - мрежите с общодостъпно предаване

Мрежите с общодостъпно предаване се характеризират с общ комуникационен канал, който се споделя от всички машини, включени в мрежата.

Всеки изпратен кадър минава през общия канал и достига до всички машини в мрежата. Адресно поле в кадъра посочва за кой е предназначен този кадър.

Когато една машина получи кадър, тя проверява дали той е предназначен за нея. Ако това е така, кадърът се приема и обработва, в противен случай се отхвърля.

# Мрежи с общодостъпно предаване

При мрежите с общодостъпно предаване основен проблем е да се определи кой да започне да използва канала, дали да има състезание или поредност.

Протоколите, които разрешават този проблем се отнасят към подниво на каналния слой, наречено **подниво за достъп до средата** (**medium access control - MAC**). Наричат се още протоколи за множествен достъп (**Multiple Access**)

Регионалните мрежи използват връзки "точка-точка" (point-to-point), докато общодостъпни многоточкови (**multipoint**) канали се използват най-вече при локалните мрежи.

# Мрежи с общодостъпно предаване

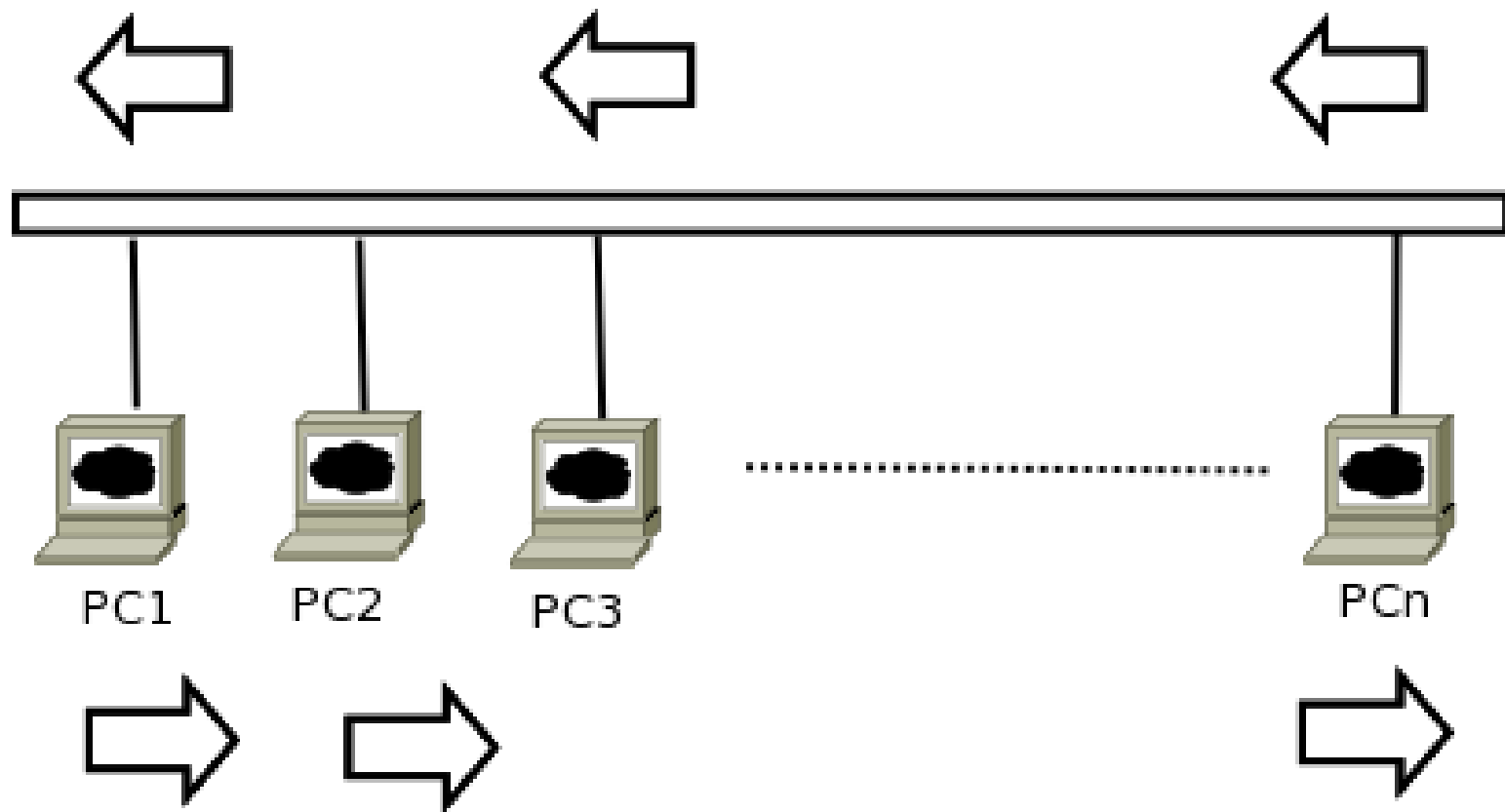
Протоколите (процедурите) за достъп до канала се делят на две основни групи:

- 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.

# Локална мрежа Token Ring



# “Чиста” ALOHA

Идва от мрежата в Университета в Хонолулу –  
Хавайските острови.

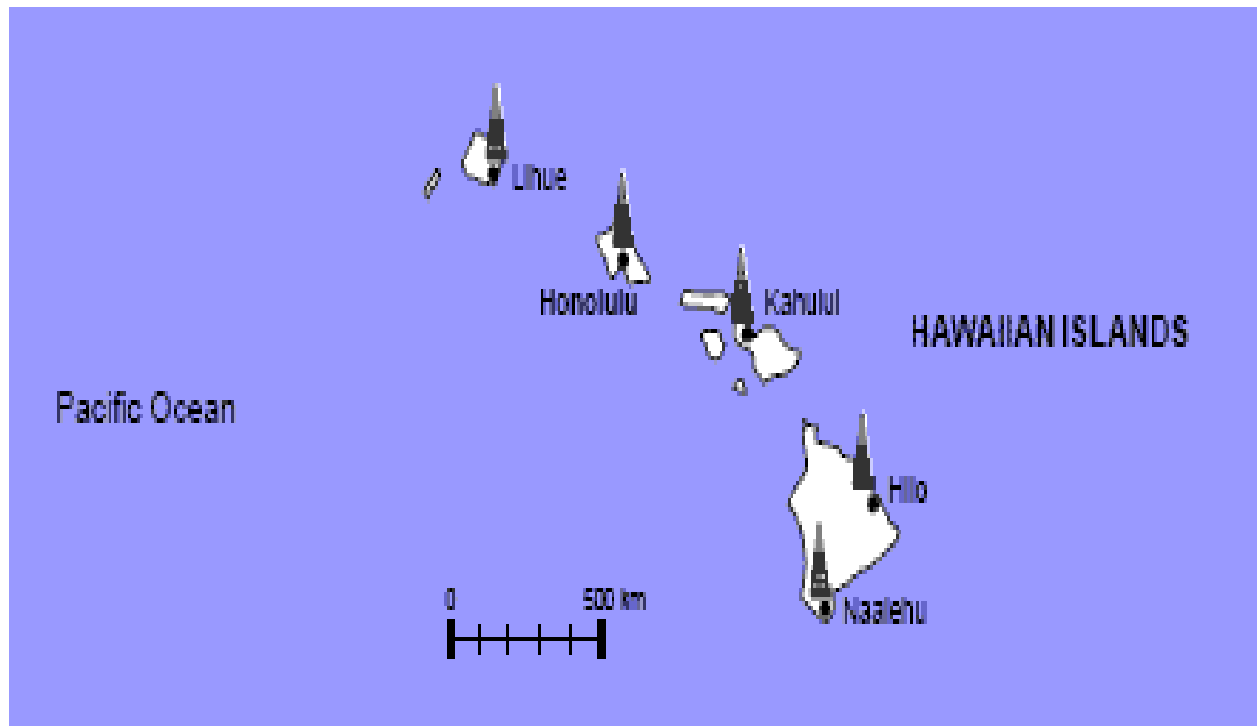
Множество радиостанции, разположени на  
различните острови.

Всяка предава, “когато си поиска”, без да се  
съобразява с другите.

Aloha си е Multiple Access (МА) и

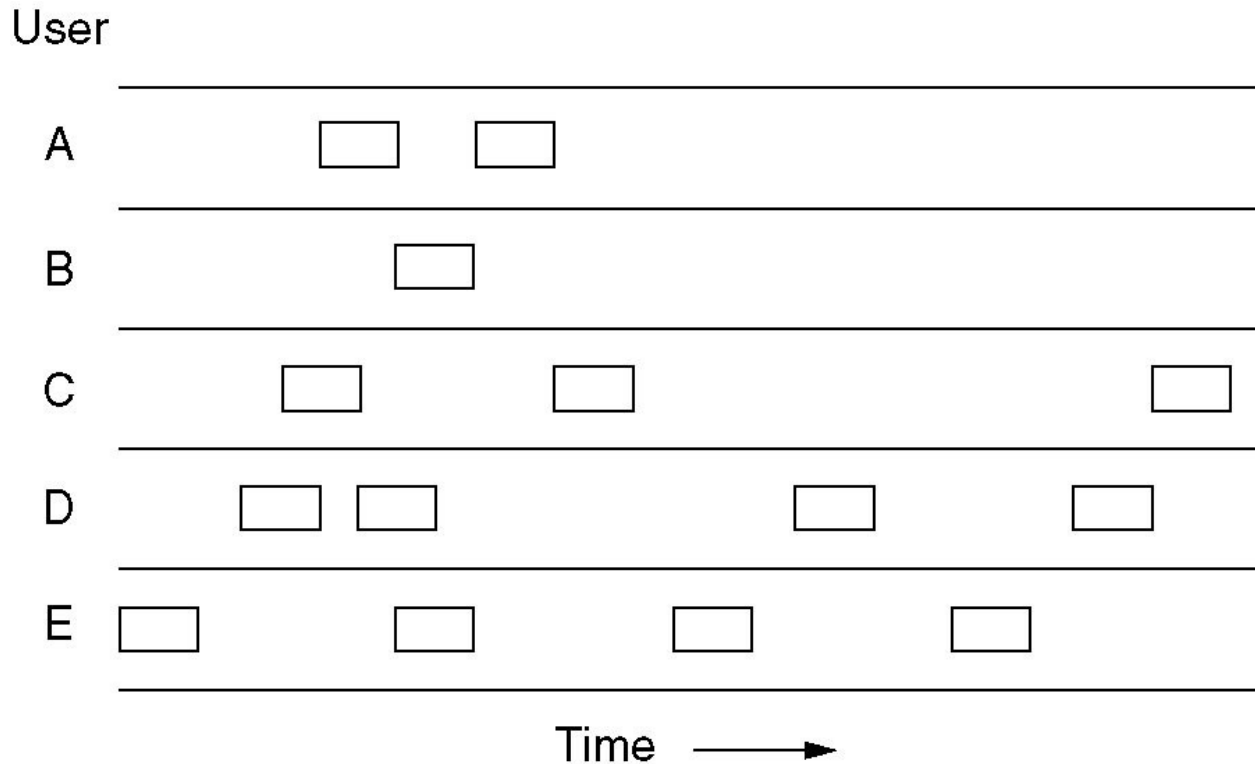
Съответства на “нерегулируемо кръстовище”

# ALOHA



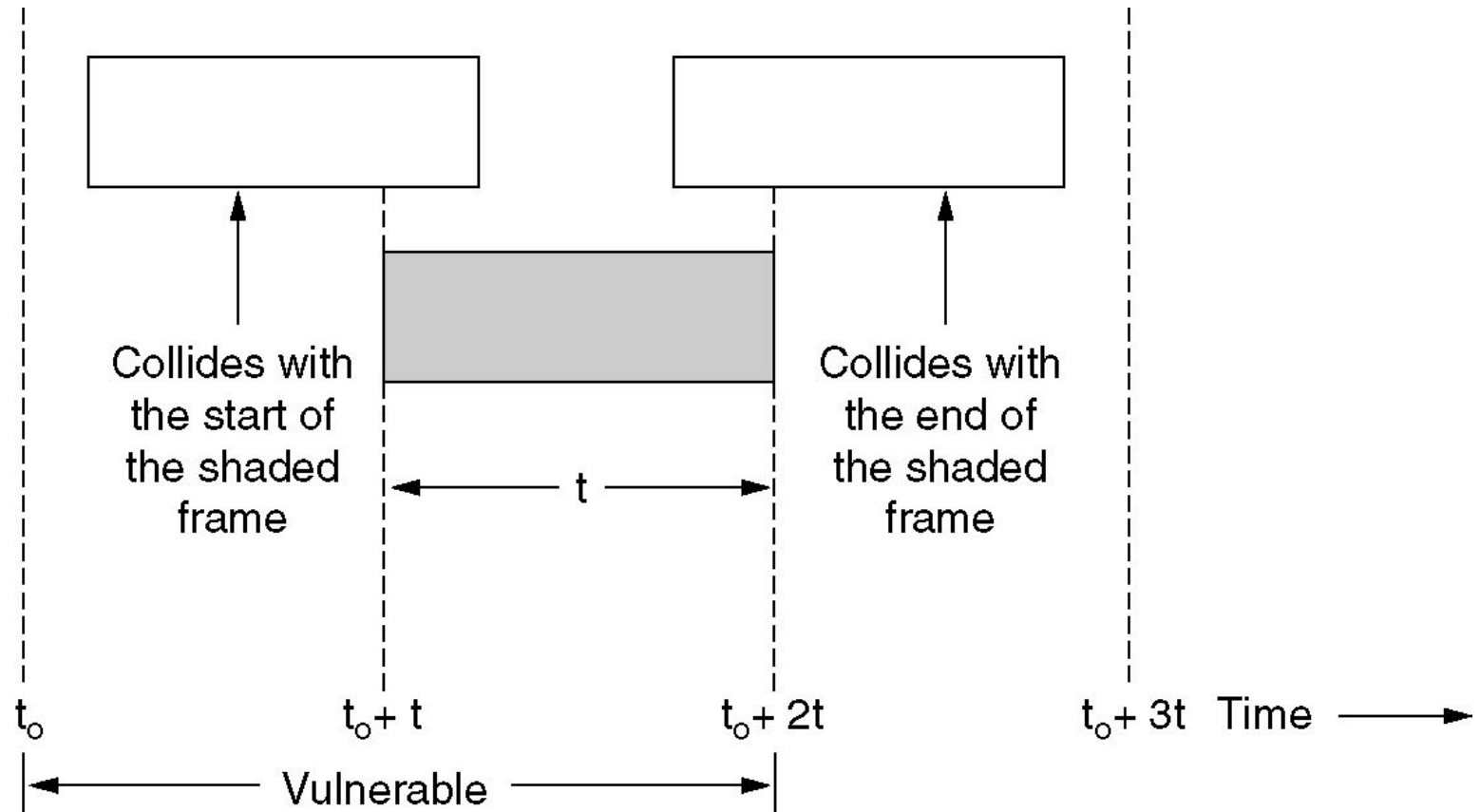


# Чиста АЛОНА



Кадрите се предават в произволно време.

# Чиста АЛОНА. Колизии.

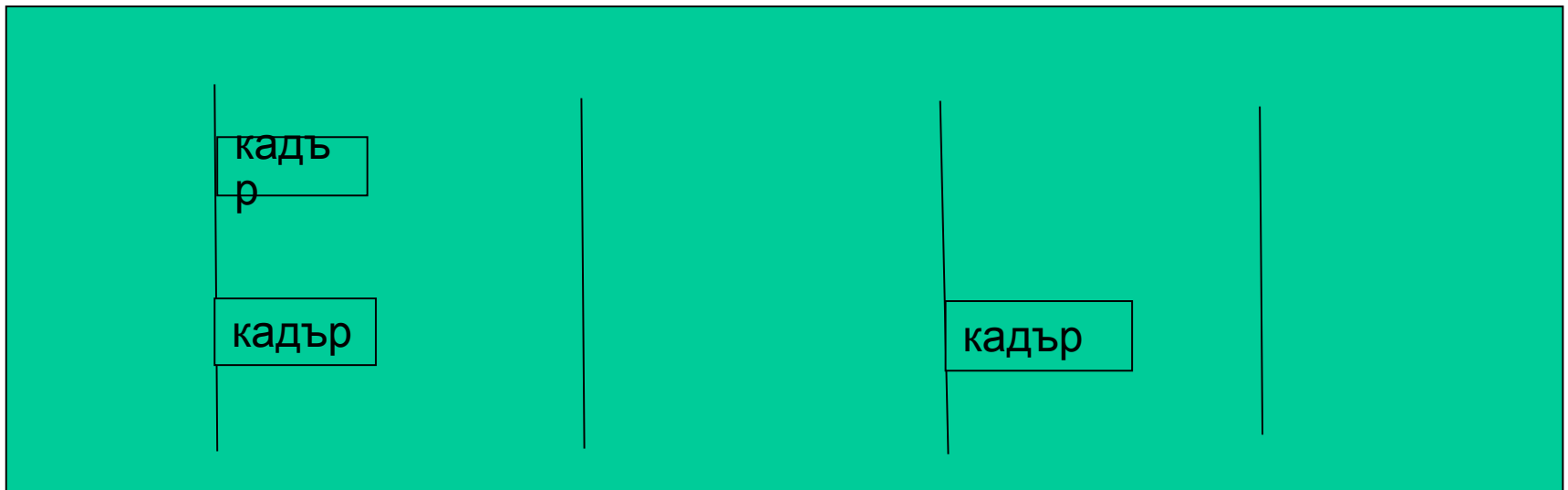


Колизии с началото и края на долния кадър.

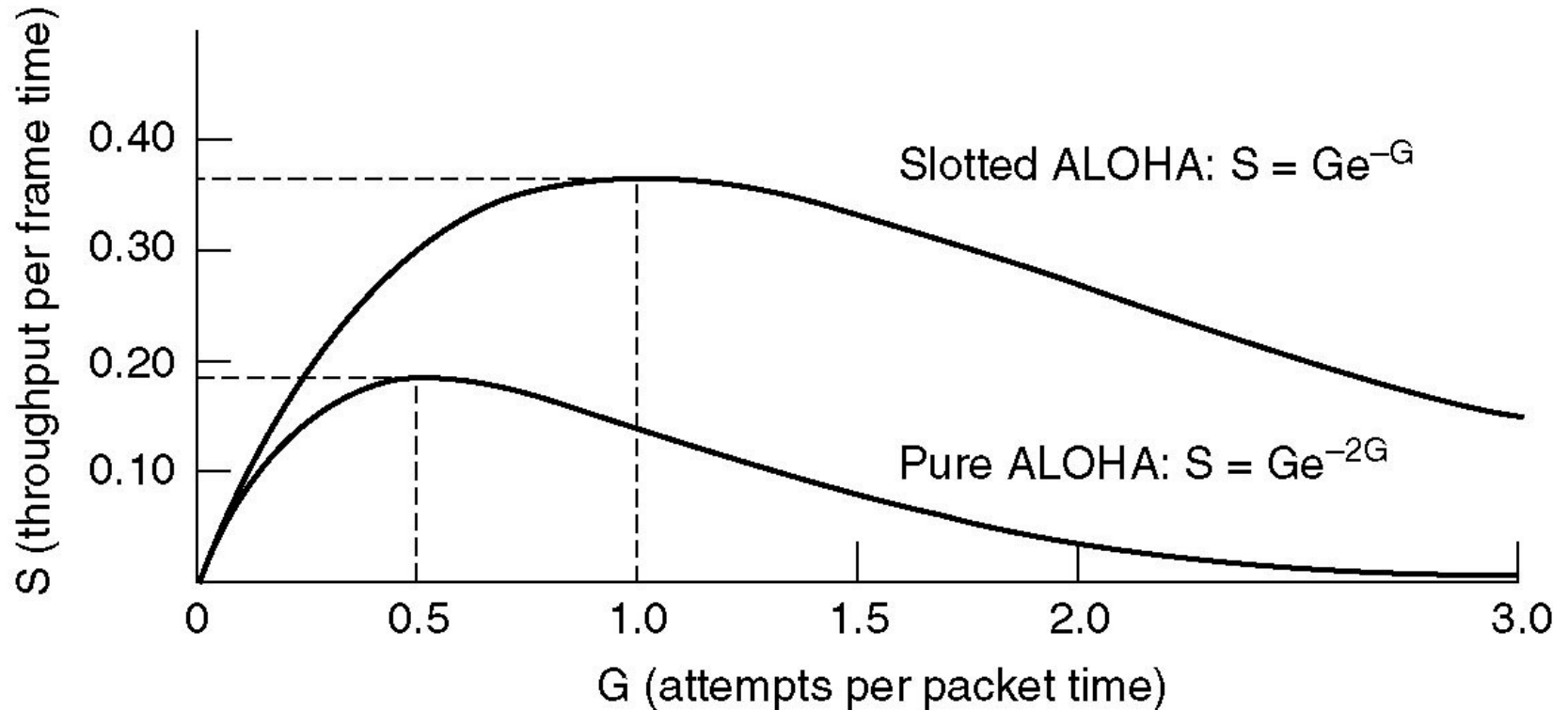
# Slotted ALOHA

Предава само в началото на синхронизирани отрязъци от време - “slot times”

Колизииите се ограничават само във времето на предаване на един кадър

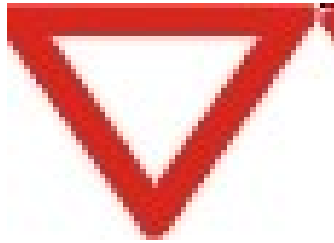


# Pure vs. Slotted ALOHA



Пропускателна способност спрямо ниво на трафика

# Carrier Sense Multiple Access (CSMA)



Можем да го сравним със знака „Пропусни движещите се по пътя с предимство!“

Протоколите, които прослушват носещата, се наричат **carrier sense multiple access** (множествен достъп с откриване на носещата – **МДОН**).

Предложени са от **Kleinrock и Tobagi** (1975), които са анализирали техни варианти.

Един от тях се нарича **1-persistent CSMA** (1 **настойчив**).

Протоколът се нарича **1-persistent**, защото станцията започва да предава с вероятност 1, ако има свободен канал.

# Nonpersistent CSMA

Този протокол не е толкова “лаком”. Станцията прослушва канала, ако никой не предава, започва тя.

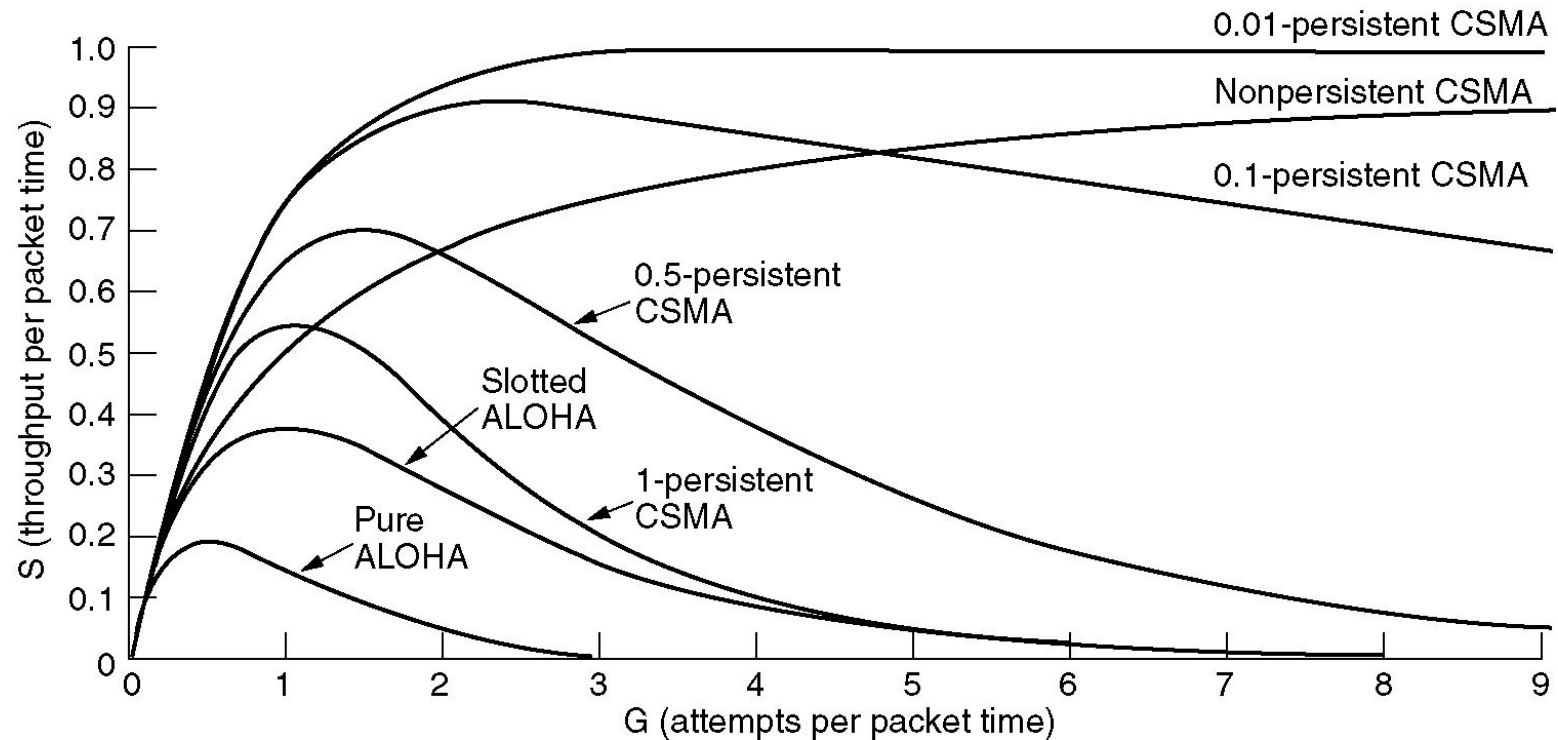
Ако каналът е зает, станцията не продължава да прослушва, а изчаква произволен период от време, след което повтаря алгоритъма.

Постига се по-добро оползотворяване на канала от 1-persistent CSMA.

**p-persistent CSMA** се отнася към канали с времоделене (time slot).

Ако каналът е свободен готовата станция започва да предава с вероятност  $p$ . С вероятност  $q = 1 - p$  отлага за следващия слот. Ако и той е свободен, или предава, или отлага с вероятност  $p$  или  $q$ .

# Persistent и Nonpersistent CSMA



Използване на канала спрямо натоварването

# CSMA плюс Collision Detection



Друго подобрение е станциите да прекратят предаването в момента, когато “забележат” колизия.

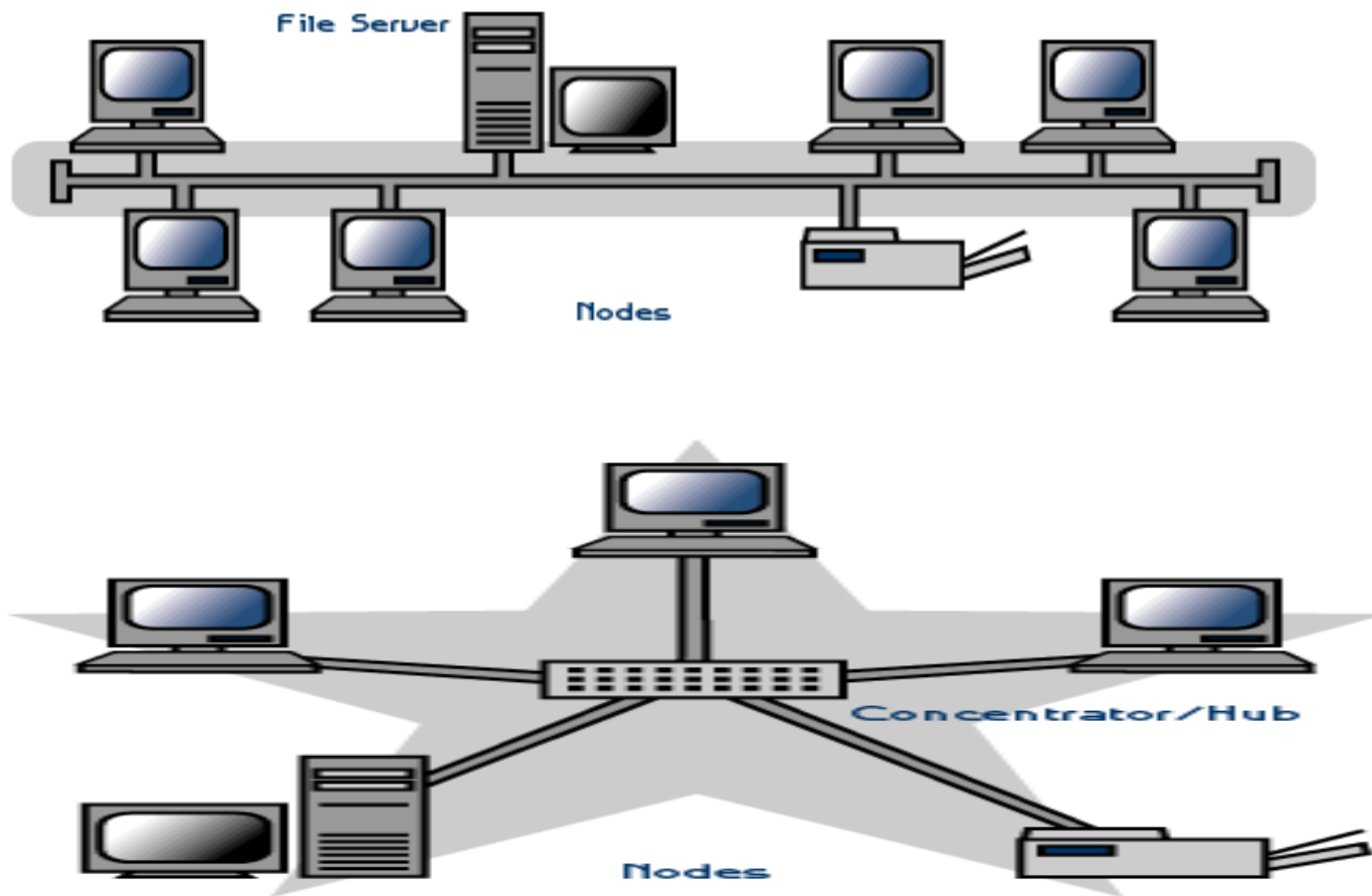
Т.е, ако две станции открият свободен канал и започнат да предават едновременно, едновременно ще разпознаят и колизията.

Те спират предаването веднага щом разпознаят колизията.

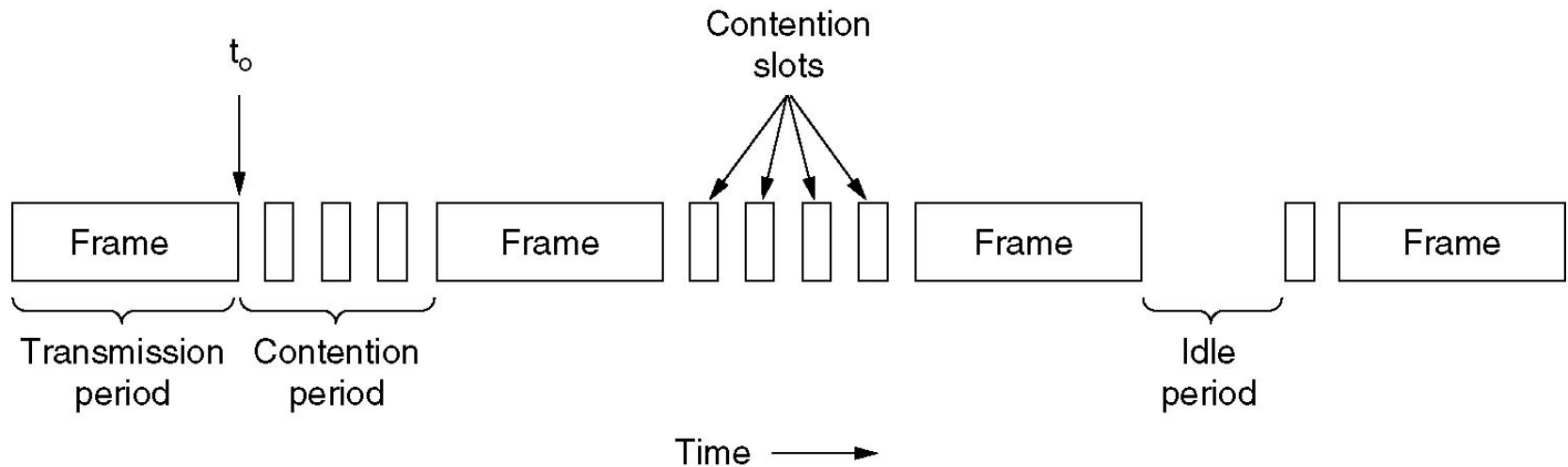
Този протокол е **CSMA/CD** (**CSMA with Collision Detection**) и се използва в LAN **Ethernet**.



# Ethernet. Логическа шина.



# CSMA/CD



CSMA/CD е в едно от трите състояния: състезание, предаване и свободно.

# CSMA/CD

Малко по-подробно за състезателната процедура.

Нека двете станции започнат да предават в момент  $t_0$ . Колко време им трябва да разберат за колизията?

Най-лошият случай. Нека времето за пътуване между двете най-отдалечени станции е  $\tau$ . В  $t_0$  една станция започва да предава. В  $\tau - \epsilon$  най-отдалечената също започва да предава. Тя веднага разпознава колизията и спира, но

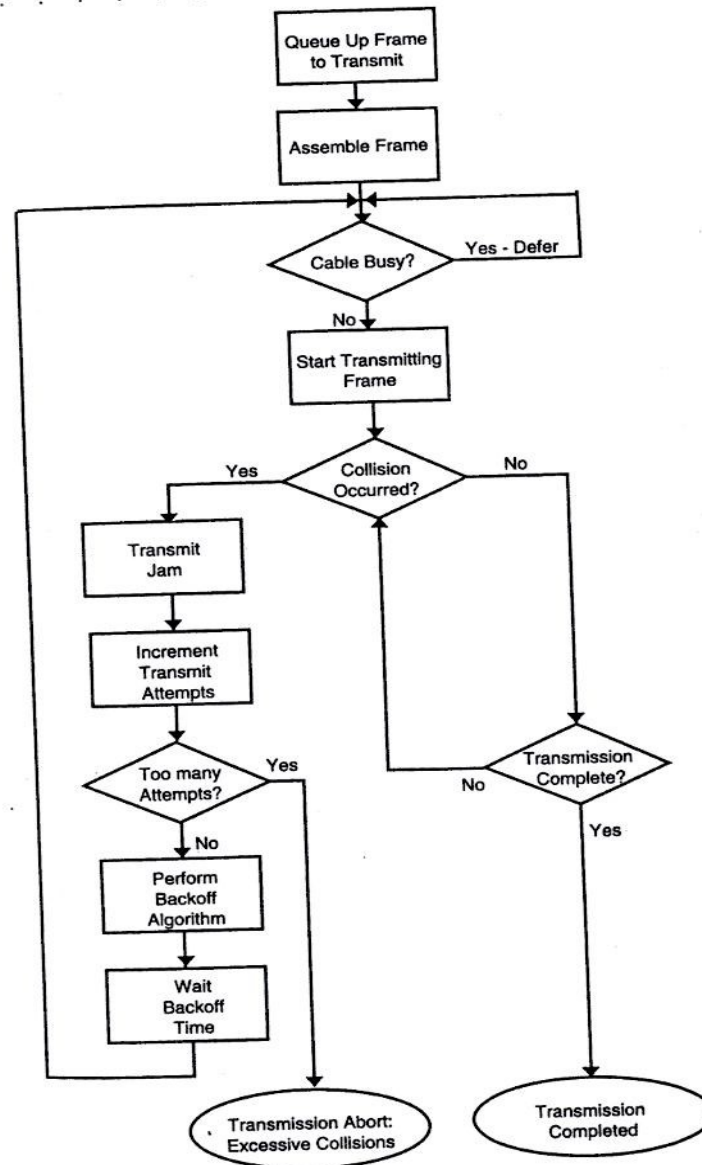
# CSMA/CD

Шумът от колизията достига до оригиналната за време  $2\tau - \epsilon$ .

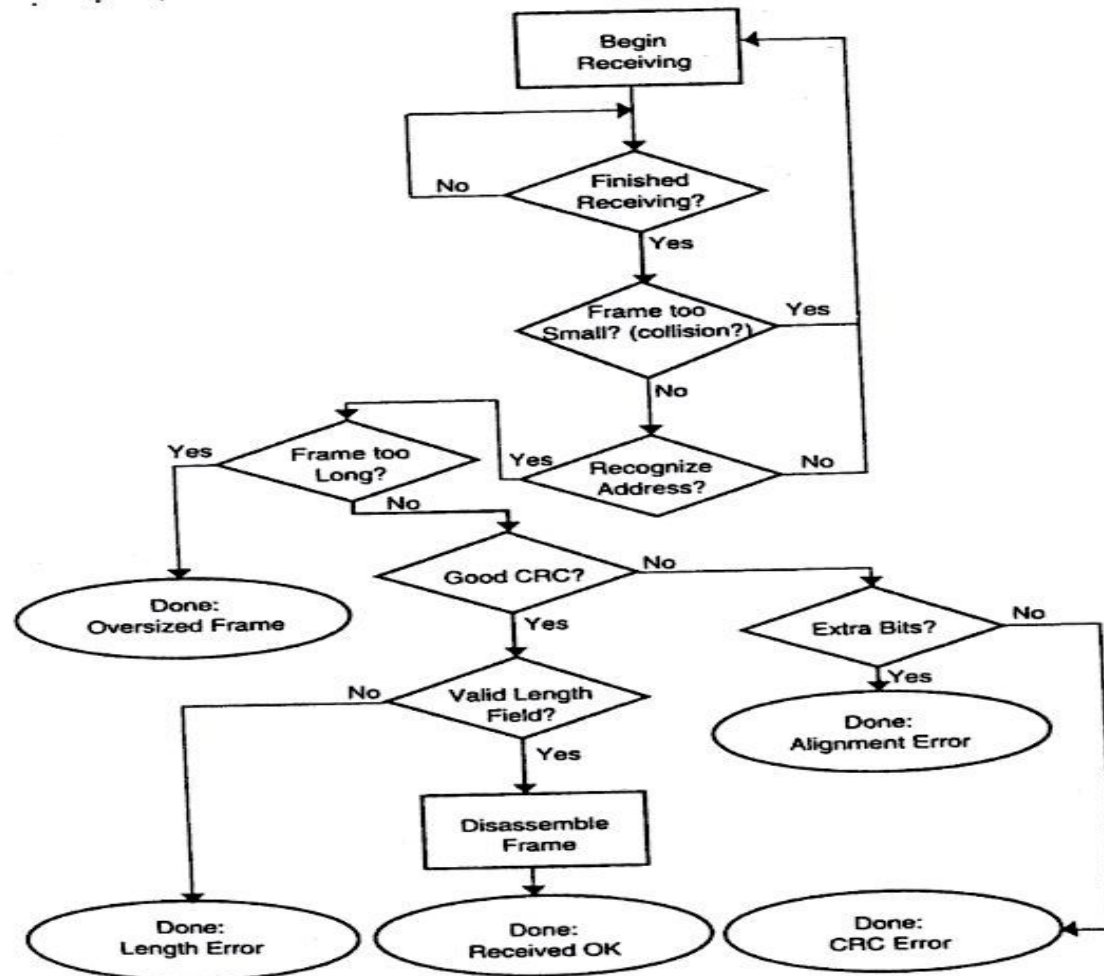
Т.е, в най-лошият случай една станция не може да е сигурна, че е “хванала” канала, докато не е предавала за  $2\tau$ , без да е чула колизия.

В 1-km коаксиален кабел  $\tau \approx 4.8 \mu\text{sec}$ .

# Предаване на Ethernet кадри



# Приемане на Ethernet кадри



# Robert M. "Bob" Metcalfe. Една жива легенда.

Откривателят на Ethernet



ART: DALE STEPHANOS

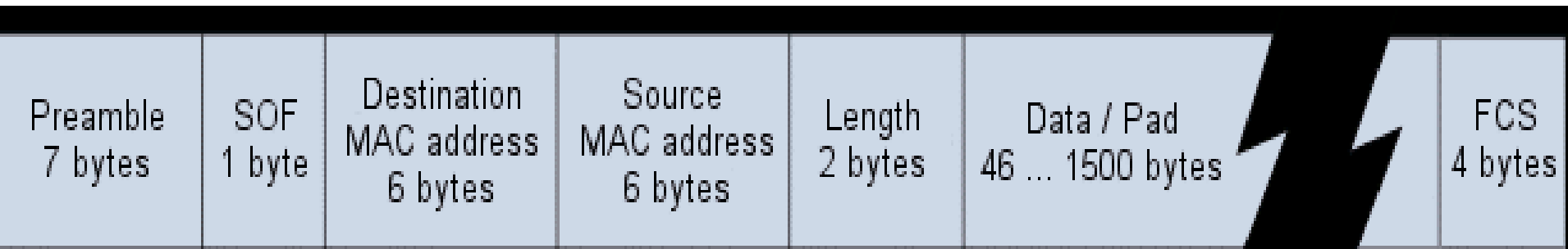
# Най-разпространената LAN Ethernet

Описана в стандарта IEEE (Institute of Electrical and Electronic Engineers) 802.3, издаден през 70-те години.

Един персонален компютър се свързва в Ethernet мрежа с помощта на NIC (Network Interface Card), която изпраща и приема кадри (frames).



# 802.3 Кадр (Novell raw)



**Preamble** = 56 бита 0-и и 1-ци.

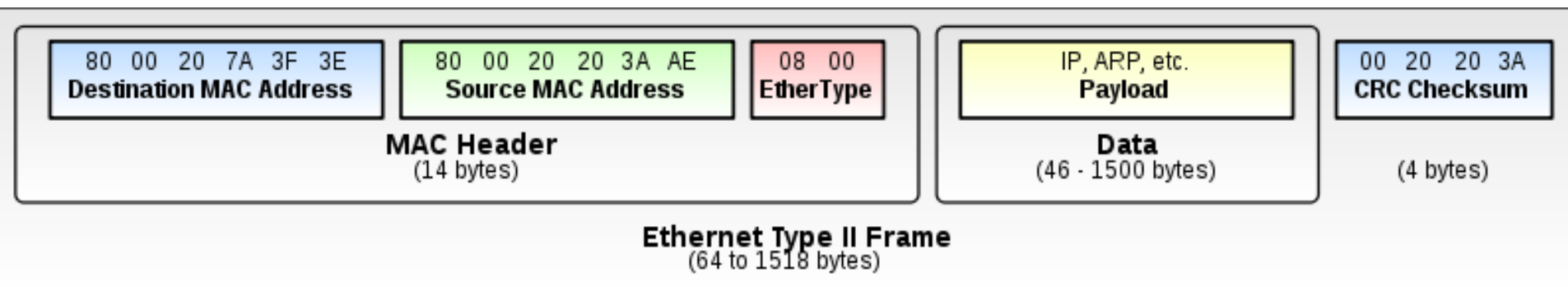
**SOF** = Start of frame: "10101011"

**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.

# Формат на кадрите в Ethernet

Данните се съдържат в полето *Data* и максималната им дължина е **1500 байта**. Това е т.нар. **payload**.

Освен максимална дължина на кадъра има и **минимална дължина** на кадъра.

В стандарта 802.3 минималната дължина на кадъра е **64 байта**.

Защото времето за разпознаване на колизия (конфликт) е времето за предаване на 64 байта.

Полето *Pad* за запълване на кадъра до 64 байта.

Полето *Checksum* е контролна сума, която се използва за откриване на грешки при предаването.

# 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:** Net bit rate = Efficiency × 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 на надеждността 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.

# Jumbo Frames.Super Jumbo Frames

9000 байта като предпочитан размер на jumbo frames е резултат от споразумение между Joint Engineering Team of Internet2 и правителствените мрежи в САЩ.

Super jumbo frames (**SJFs**) са кадри с дължина над 9000 байта.

С растежа на скоростта на линията пропорционално би трябвало да расте и **payload**. Това обаче зависи от възможностите на логическите схеми, обработващи пакетите.

Колкото и да са трудни преговорите в тази насока, възможно е да се достигне дължина от 64000 байта.

# Шестнадесетични числа (Hexadecimal)

Ед на шестнадесетична цифра:  
– 4 двоични разряда:

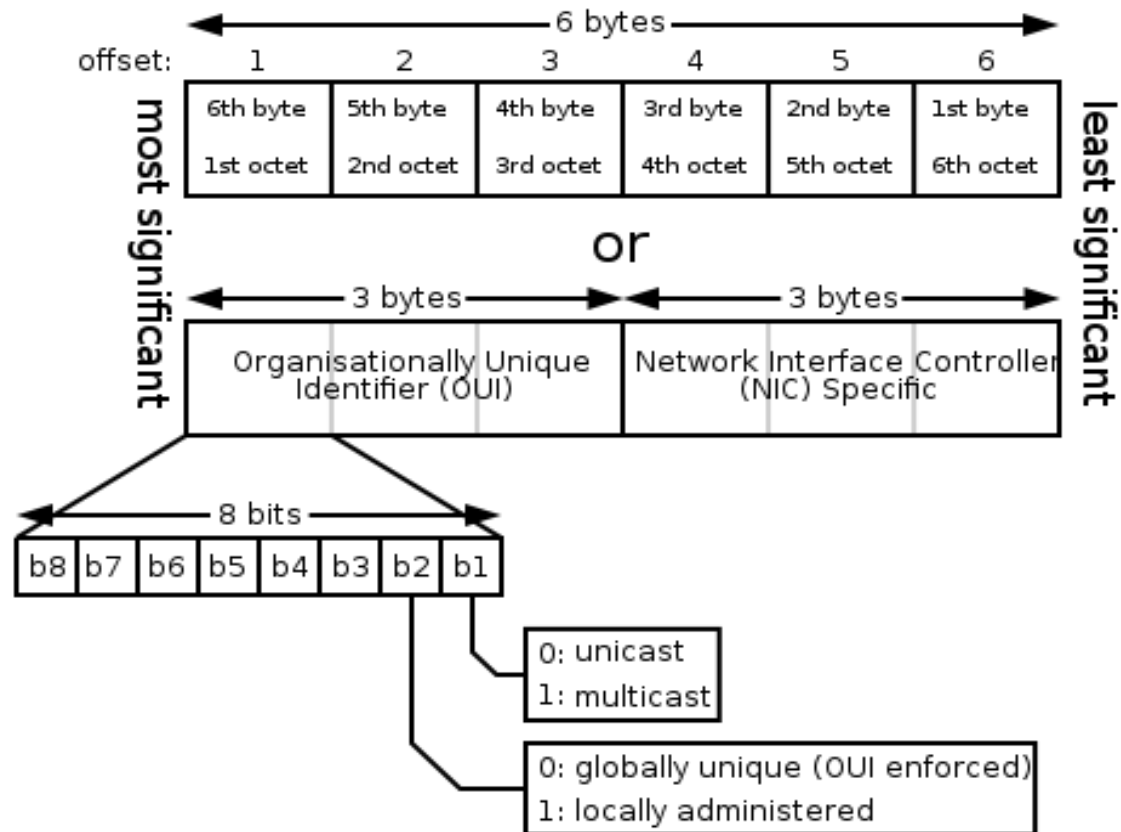
<b>0</b> <sub>hex</sub> = <b>0</b> <sub>dec</sub> = <b>0</b> <sub>oct</sub>	0	0	0	0
<b>1</b> <sub>hex</sub> = <b>1</b> <sub>dec</sub> = <b>1</b> <sub>oct</sub>	0	0	0	1
<b>2</b> <sub>hex</sub> = <b>2</b> <sub>dec</sub> = <b>2</b> <sub>oct</sub>	0	0	1	0
<b>3</b> <sub>hex</sub> = <b>3</b> <sub>dec</sub> = <b>3</b> <sub>oct</sub>	0	0	1	1
<b>4</b> <sub>hex</sub> = <b>4</b> <sub>dec</sub> = <b>4</b> <sub>oct</sub>	0	1	0	0
<b>5</b> <sub>hex</sub> = <b>5</b> <sub>dec</sub> = <b>5</b> <sub>oct</sub>	0	1	0	1
<b>6</b> <sub>hex</sub> = <b>6</b> <sub>dec</sub> = <b>6</b> <sub>oct</sub>	0	1	1	0
<b>7</b> <sub>hex</sub> = <b>7</b> <sub>dec</sub> = <b>7</b> <sub>oct</sub>	0	1	1	1
<b>8</b> <sub>hex</sub> = <b>8</b> <sub>dec</sub> = <b>10</b> <sub>oct</sub>	1	0	0	0
<b>9</b> <sub>hex</sub> = <b>9</b> <sub>dec</sub> = <b>11</b> <sub>oct</sub>	1	0	0	1
<b>A</b> <sub>hex</sub> = <b>10</b> <sub>dec</sub> = <b>12</b> <sub>oct</sub>	1	0	1	0
<b>B</b> <sub>hex</sub> = <b>11</b> <sub>dec</sub> = <b>13</b> <sub>oct</sub>	1	0	1	1
<b>C</b> <sub>hex</sub> = <b>12</b> <sub>dec</sub> = <b>14</b> <sub>oct</sub>	1	1	0	0
<b>D</b> <sub>hex</sub> = <b>13</b> <sub>dec</sub> = <b>15</b> <sub>oct</sub>	1	1	0	1
<b>E</b> <sub>hex</sub> = <b>14</b> <sub>dec</sub> = <b>16</b> <sub>oct</sub>	1	1	1	0
<b>F</b> <sub>hex</sub> = <b>15</b> <sub>dec</sub> = <b>17</b> <sub>oct</sub>	1	1	1	1



# Шестнадесетични числа към десетични

Decimal	Hex	Decimal	Hex	Decimal	Hex
1	1	11	B	30	1E
2	2	12	C	40	28
3	3	13	D	50	32
4	4	14	E	60	3C
5	5	15	F	70	46
6	6	16	10	80	50
7	7	17	11	90	5A
8	8	18	12	100	64
9	9	19	13	500	1F4
10	A	20	14	1000	3E8

# Формат на MAC адрес



# Формат на МАС адрес

Media Access Control адресът (МАС адрес), Ethernet Hardware Address (ЕНА) или хардуерен адрес, адрес на адаптера или **физически адрес** е квазиуникален идентификатор, присвоен на мрежов адаптер или NIC от производителя. В този случай МАС адресът съдържа закодиран идентификатора на производителя.

IEEE дефинира три схеми за формулиране на МАС адрес: **MAC-48, EUI-48 и EUI-64**. Търговски марки на IEEE са "EUI-48" и "EUI-64" (**EUI - Extended Unique Identifier**). Разликата между EUI-48 и MAC-48 е чисто семантична (но не и синтактическа): MAC-48 се използва за мрежов хардуер, а EUI-48 идентифицира други устройства и софтуер.

Записва се с **шестнадесетични** цифри.

# MAC spoofing

Макар че е смятан за перманентен и глобално уникален, днес е възможно да се смени MAC адреса (т.е не е “прогорен”) - **MAC spoofing**.

Оригиналният IEEE 802 MAC произлиза от Xerox Ethernet. Съдържа  $2^{48}$  или 281,474,976,710,656 възможни адреси.

Според IEEE MAC-48 пространството няма да се изчерпи до 2100 г.

Адресите могат да бъдат “**универсално администрирани**” или “**локално администрирани**”.

# Формат на MAC адрес

Универсално администриран е присвоен от производителя, още “прогорен” - "burned-in addresses" (BIA). Първите три октета показват организацията, издала идентификатора - Organizationally Unique Identifier (OUI).

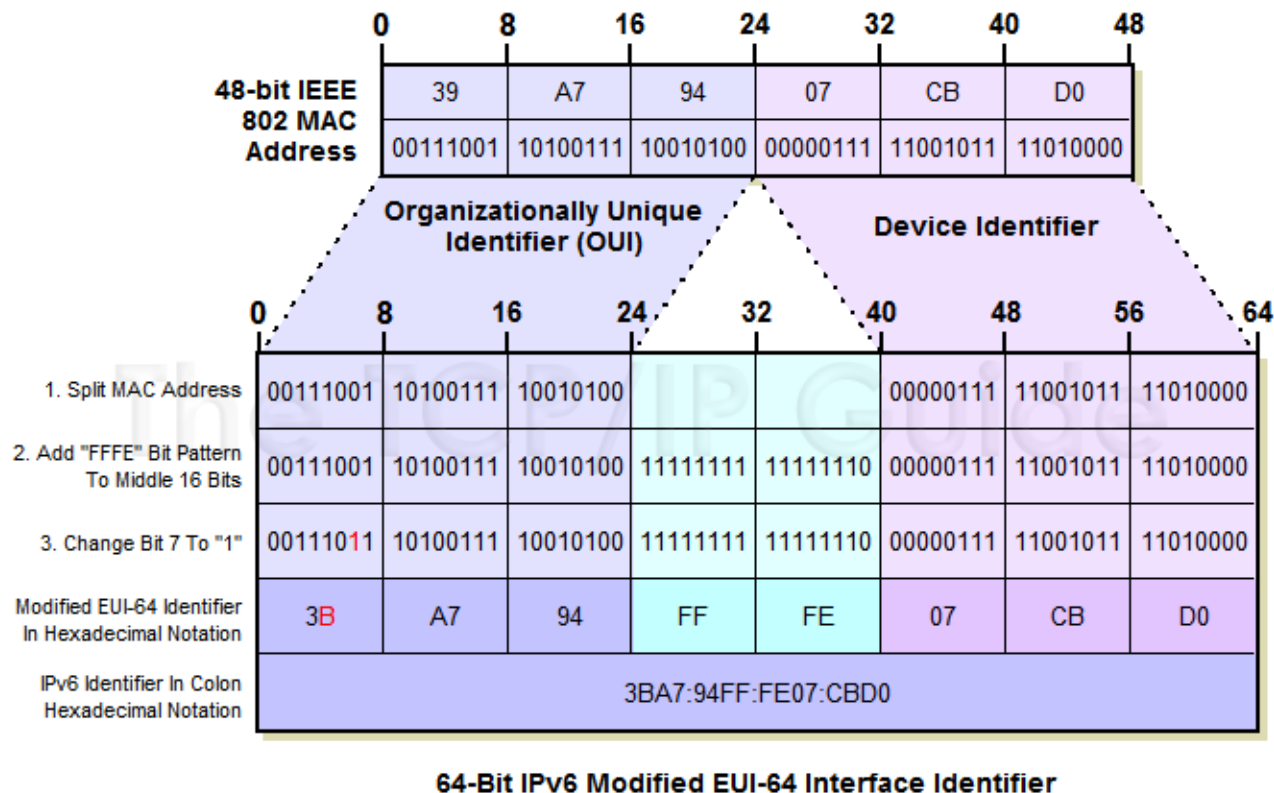
Следващите три октета (MAC-48 и EUI-48) или пет (EUI-64) се дават от самата организация.

Локално администриран се присвоява от мрежовия администратор, отменяйки “прогорения”. Те нямат OUI.

Разпознават се по bit 2 ( $2^1$ ) в най-старшия октет на MAC-а. Ако е 0, адресът е универсален. Ако е 1, адресът е локален. Т.е. е 0 на всички OUI-та.

Ако най-младшият бит – bit 1 ( $2^0$ ) е 0, кадърът е предназначен за конкретна NIC - unicast. Ако е 1, кадърът трябва да достигне няколко (група) NIC-ве. Нарича се групов - multicast.

# EUI-64 формат



# EUI-64 формат

EUI-64 се използват:

- \* FireWire
- \* IPv6 (младшите 64 бита в unicast мрежов адрес или [link-local](#) адрес)

Преобразуване на 48-бит MAC адрес в IPv6 модифициран EUI-64 идентификатор:

1. Вземаме 24-бит OUI частта и я поставяме в най-левите 24 бита на interface ID. А 24-бит локална част слагаме в най-десните 24 бита на interface ID.
2. В оставащите в средата 16 бита на interface ID поставяме стойността “11111111 11111110” (“FFFE” hex).
3. Така адресът ни е в EUI-64 формат. Променяме “[universal/local](#)” бита (бит 7 отляво) от 0 на 1.

И получаваме модифицирания [EUI-64 interface ID](#).

# Ethernet кабели и топологии

Name	Cable	Max. seg.	Nodes/seg.	Advantages
10Base5	Thick coax	500 m	100	Original cable; now obsolete
10Base2	Thin coax	185 m	30	No hub needed
10Base-T	Twisted pair	100 m	1024	Cheapest system
10Base-F	Fiber optics	2000 m	1024	Best between buildings

**100BASE-TX**: Използва 2 чифта по Category 5 ([IEEE 802.3u](#)).

**100BASE-FX**: 100 Mbit/s Ethernet по FO.

**1000BASE-T**: 1 Gbit/s over Category 5e copper cabling ([802.3ab](#)).

**1000BASE-SX**: 1 Gbit/s по MM FO.

**1000BASE-LX**: 1 Gbit/s по SM FO (големи разстояния).

**10GBASE-LX4**: WDM - 240 m и 300 m по MM FO. 10 km по SM FO ([802.3ae](#)).

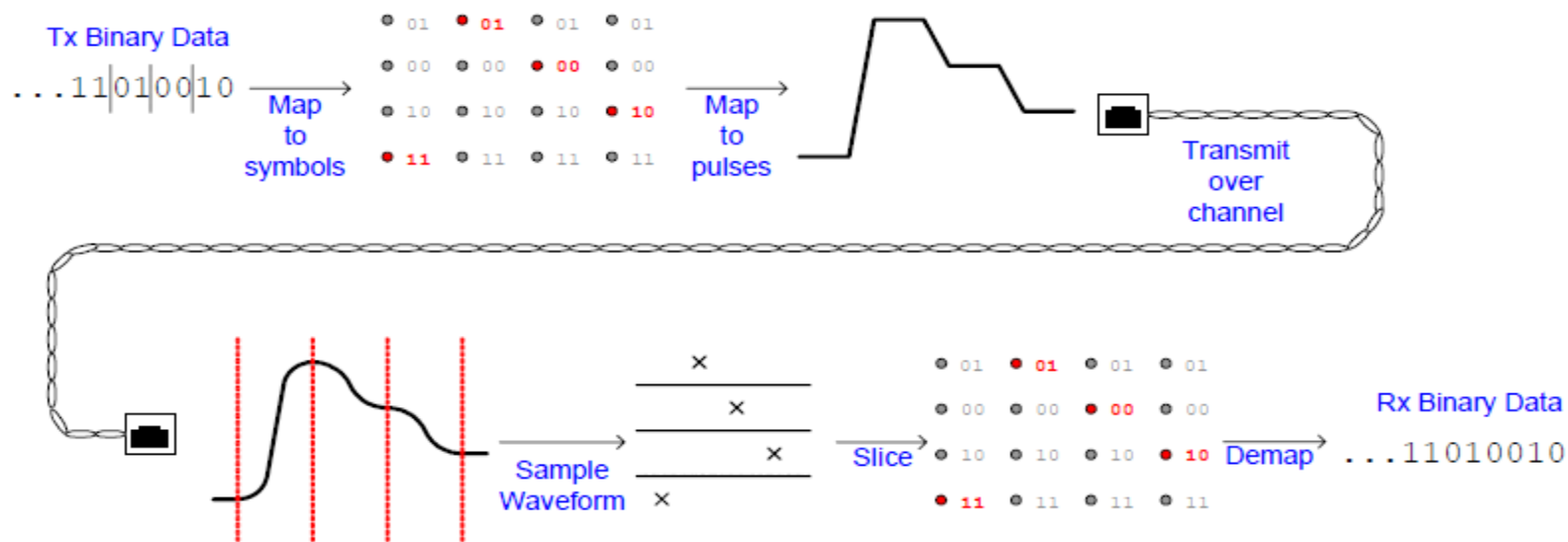
**10GBASE-LR** и **10GBASE-ER**: 10 km и 40 km по SM FO.

**10GBASE-SW**, **10GBASE-LW** и **10GBASE-EW**. Върху WAN PHY

**10GBASE-T**: меден кабел Категория 6a ([802.3an](#))



# Сигнали с много нива



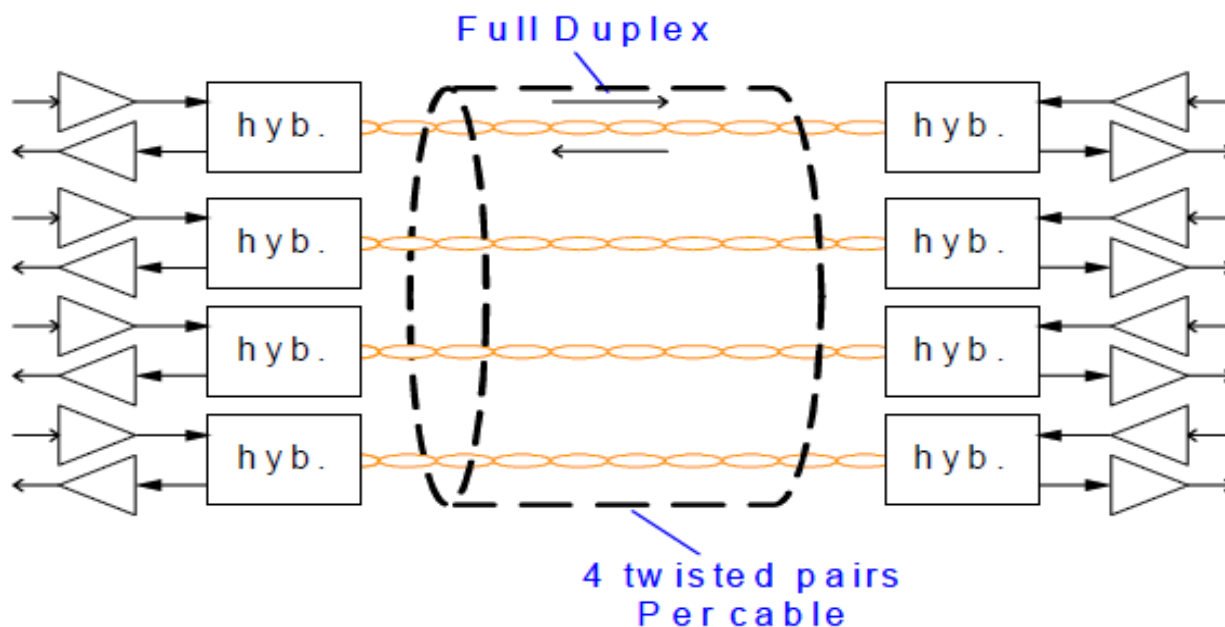
Предавайки сигнали с много нива на напрежение, **кодираме групи от битове**, а не с две нива да се кодира един единствен бит.

$$bw \sim 1/\text{бр.нива}$$

На фигурата имаме **4 нива**, кодираме **два бита** на ниво.

Така по по един и същ кабел предаваме 10/100/1000 Mbps.

# 1000Base-T

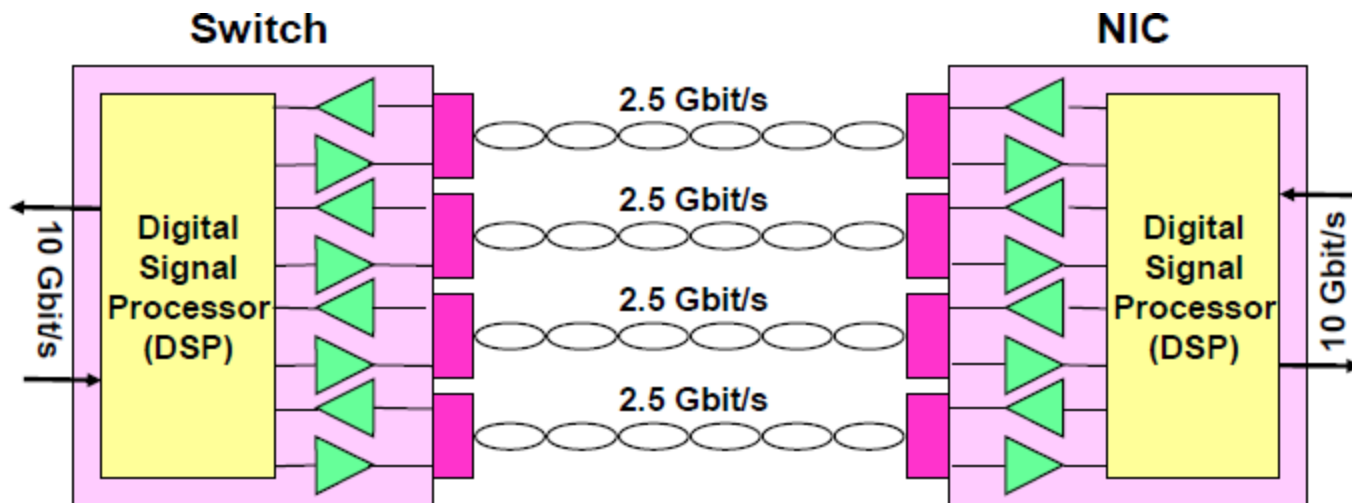


**10BASE-T** и **100BASE-T** предава по два от чифтовете.

**1000BASE-T** Twisted-pair cabling (Cat-5, Cat-5e, Cat-6, or Cat-7) 100 meters използва и 4-те чифта.

**1000BASE-TX** Twisted-pair cabling (Cat-6, Cat-7) 100 meters. **Не се използва.**

# 10GBase-T



10GBASE-T предава и по 4-те чифта - 100 m SFTP кабел (Cat. 6a).

2.5 Gbit/s на чифт.

# 40/100 Gigabit Ethernet

PHY	40 Gigabit Ethernet	100 Gigabit Ethernet
at least 1 m over a backplane	40GBASE-KR4	
approximately 7 m over copper cable	40GBASE-CR4	100GBASE-CR10
at least 100 m over OM3 MMF	40GBASE-SR4	100GBASE-SR10
at least 125 m over OM4 MMF <sup>[7]</sup>	40GBASE-SR4	100GBASE-SR10
at least 10 km over SMF	40GBASE-LR4	100GBASE-LR4
at least 40 km over SMF		100GBASE-ER4

40 Gigabit Ethernet (40GbE) и 100 Gigabit Ethernet (100GbE) са разработени от IEEE P802.3ba.

Ethernet кадрите се предават по множество 10 Gb/s или 25 Gb/s ленти. (50/125  $\mu\text{m}$  (OM2) и 62.5/125  $\mu\text{m}$  (OM1) multi-mode fiber. 50/125  $\mu\text{m}$  "лазерно оптимизиран" OM3 fiber.)

# 400-Gigabit Ethernet

Все повече нарастват нуждите от трафик: Web 2.0, cloud услуги, Facebook още в 2010 г. Казаха, че им трябва Terabit Ethernet в центровете им за обработване на данни.

Анализът необходимата пропускателна способност доведе до извода, че най-разумно е следващата скорост на Етернет да бъде 400 Gbps.

Така се създаде IEEE 802.3 400Gbps Study Group да пише новата глава в историята на Ethernet.

400-Gigabit Ethernet би се получила най-добре с  $4 * 100\text{-Gigabit}$  интерфейси.

Други възможности са  $8 * 50\text{ Gbps}$  или  $16 * 25\text{ Gbps}$ .

# Към Terabit Ethernet

Terabit Ethernet изисква подобрение на PCI Express стандарта.

Голямо предизвикателство за технологията.

Интегриране на оптика в силициеви чипове - 40 лазера, всеки предаващ по 40Gbps.

# Ethernet кабели и топологии



# Ethernet кабели и топологии

В началото в Ethernet се използва **коаксиален кабел** и скоростта на предаването е достигала 10 Mb/s.

По-нататък се въвежда използването на **хъбове (hub)**. При окабеляване 100Base-T4 каналните станции се свързват към хъба чрез четири усукани двойки **UTP Category 3**

100Base-TX чрез две усукани двойки (**UTP Category 5**). По една от усуканите двойки се предава към хъба, а по другата се приема от него (при 100Base-T4 останалите две усукани двойки се превключват по посока на предаването). Скоростта на предаване достига 100 Mb/s

Хъбът не премахва проблема с колизиите. Подобно на коаксиалния сегмент е един **колизионен домейн** и един **бродкаст домейн**.

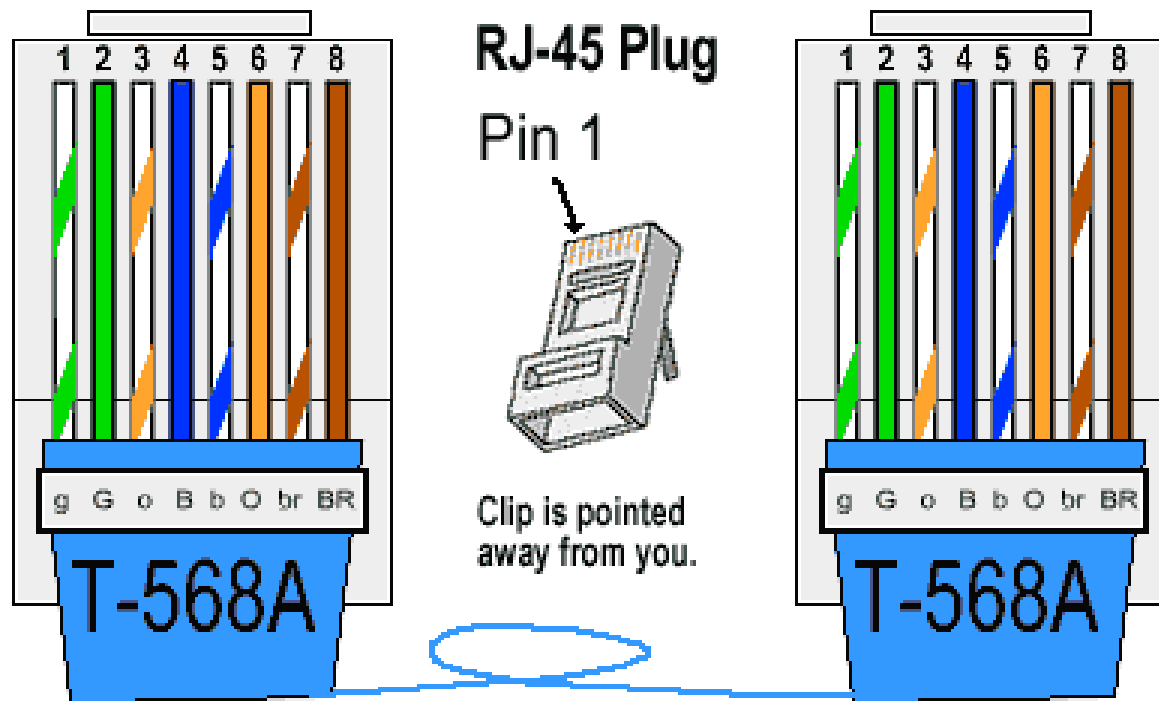


# Ethernet кабели и топологии

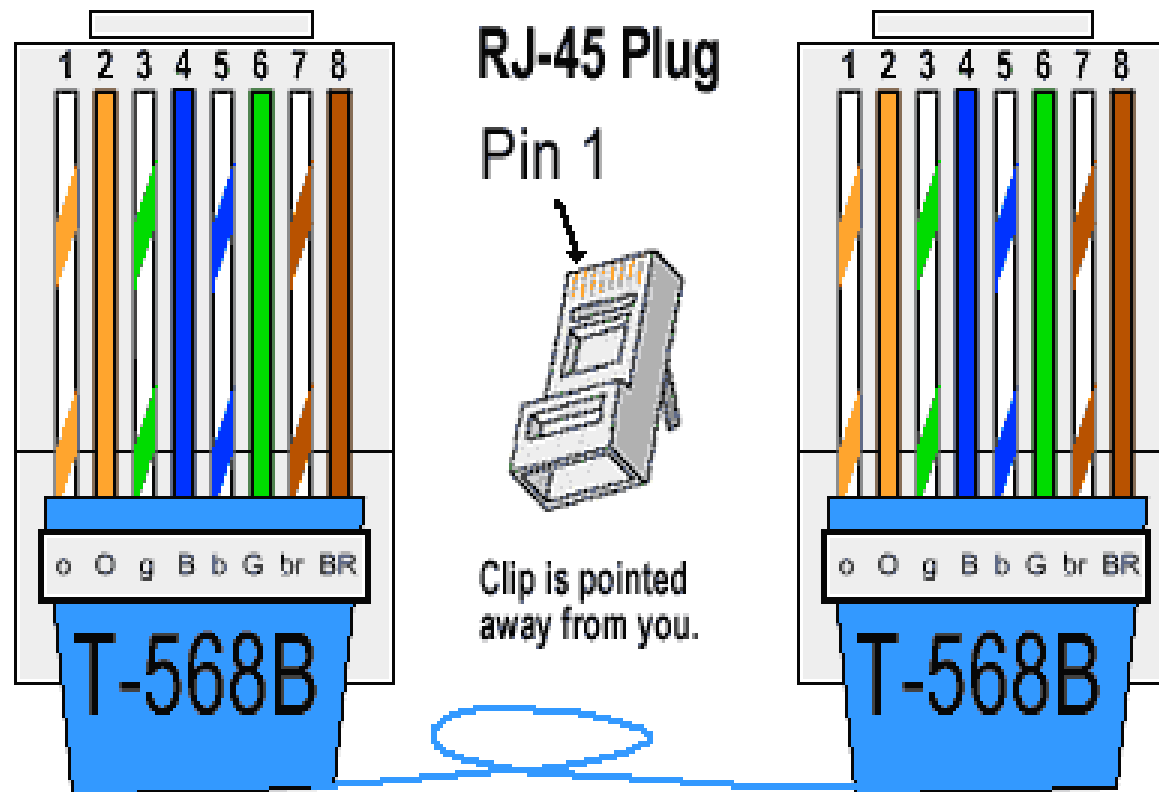
Станциите се свързват към хъба в **прав кабел**, т.е. предаващата двойка на всяка станция съответства на предаващата двойка на хъба и съответно приемащата двойка на всяка станция съответства на приемащата двойка на хъба.

При свързване на два хъба чрез усукана двойка, обаче, се използва **кръстосан (cross) кабел**, т.е. предаващата двойка на единия хъб се свързва с приемащата двойка на другия хъб и обратно.

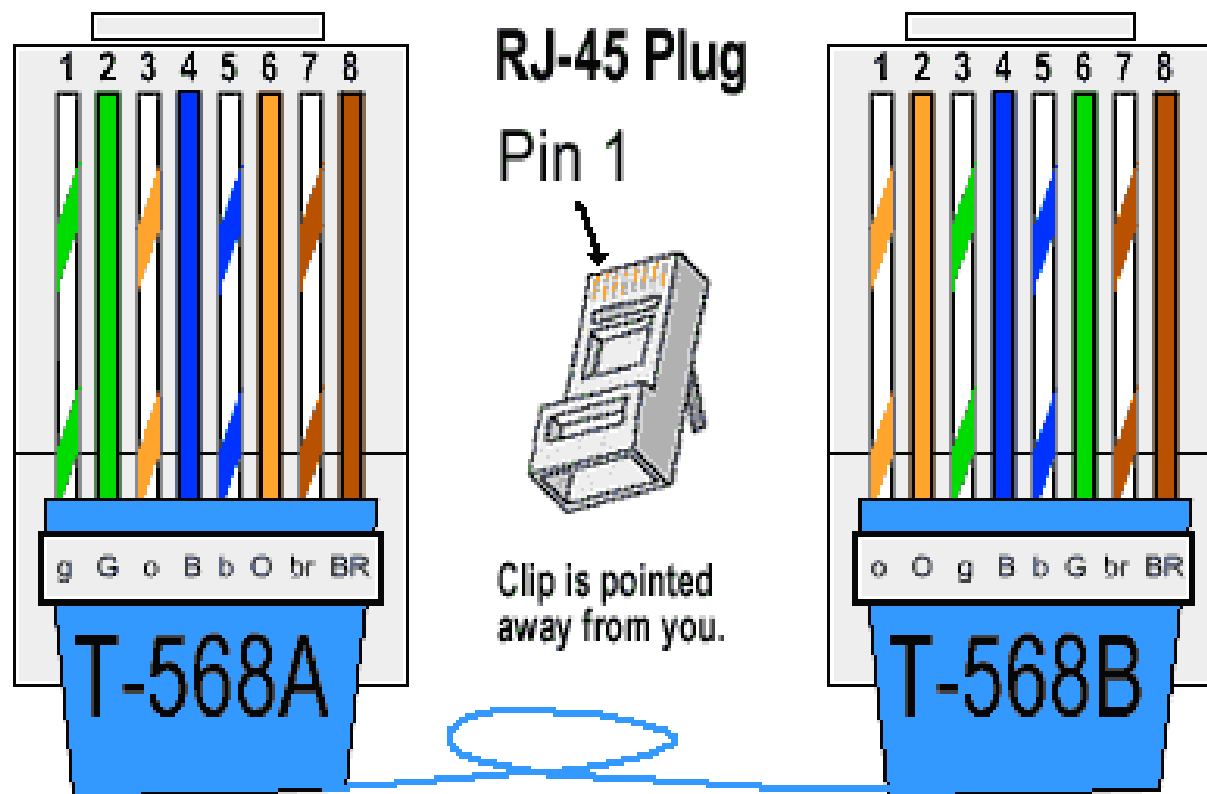
# Прав (Straight-Through) кабель



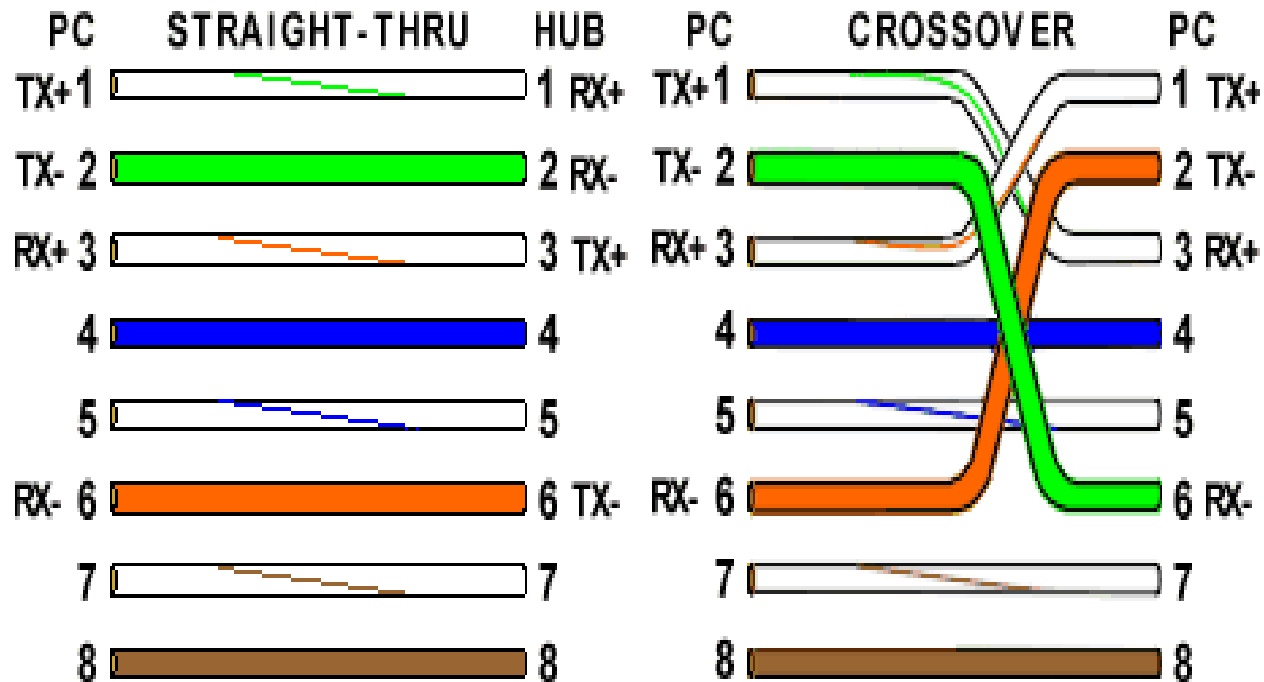
# Прав (Straight-Through) кабель



# Кръстосан (Crossover) кабел



# Straight vs. Cross (теория)



# Хъб и повторител

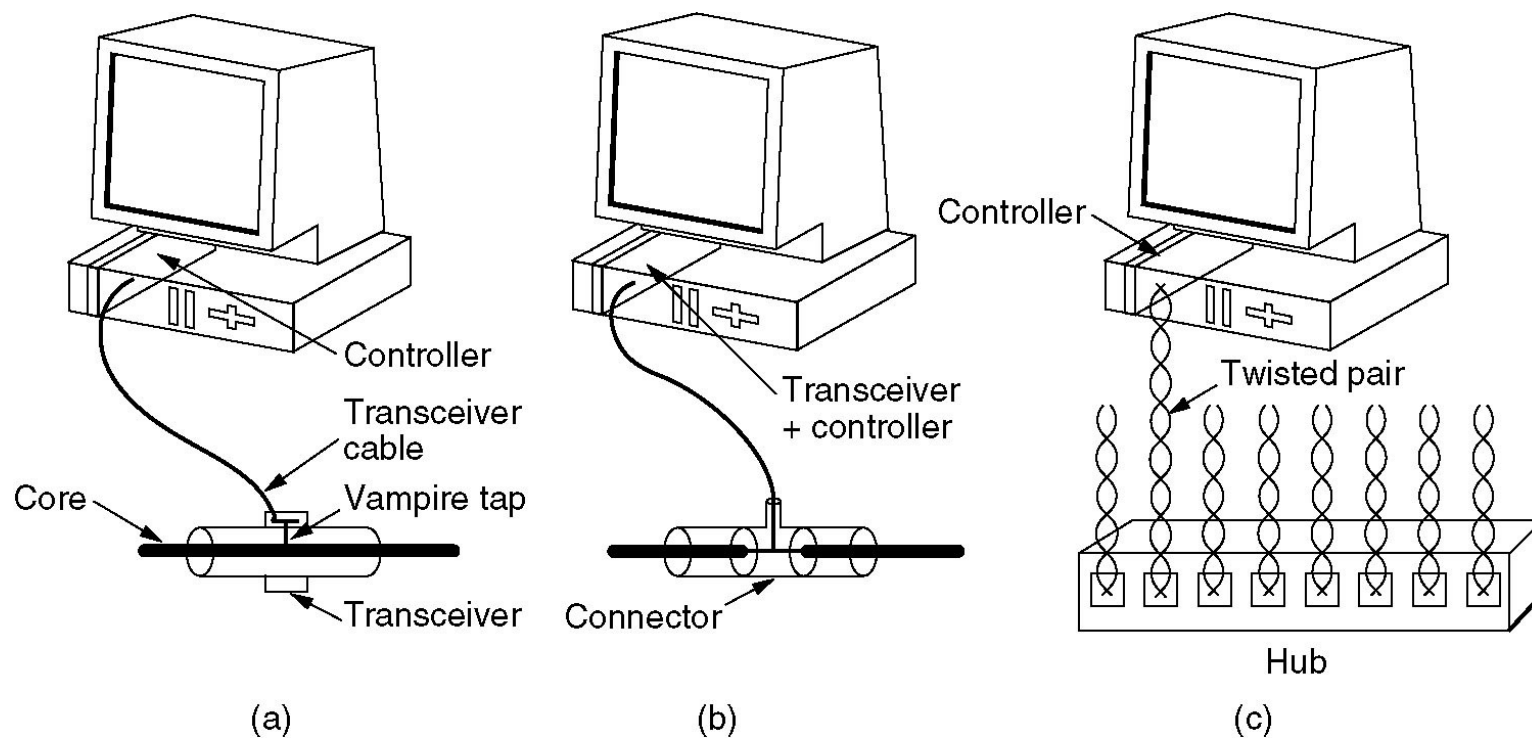
Ако хъбът получи кадър по някоя линия, той изпраща този кадър по всички останали линии. Хъбът не знае адресите на каналните станции.

Хъбът е пример за устройство, чрез което се препредават кадри от един кабел към друг. Той **работи на физическо ниво**.

Друго подобно устройство на физическо ниво е **повторителят (repeater)**.

Той приема сигнал на единия си порт, усилва го и предава сигнала на другия си порт. По този начин може да се увеличи максималната дължина на кабела в една локална мрежа

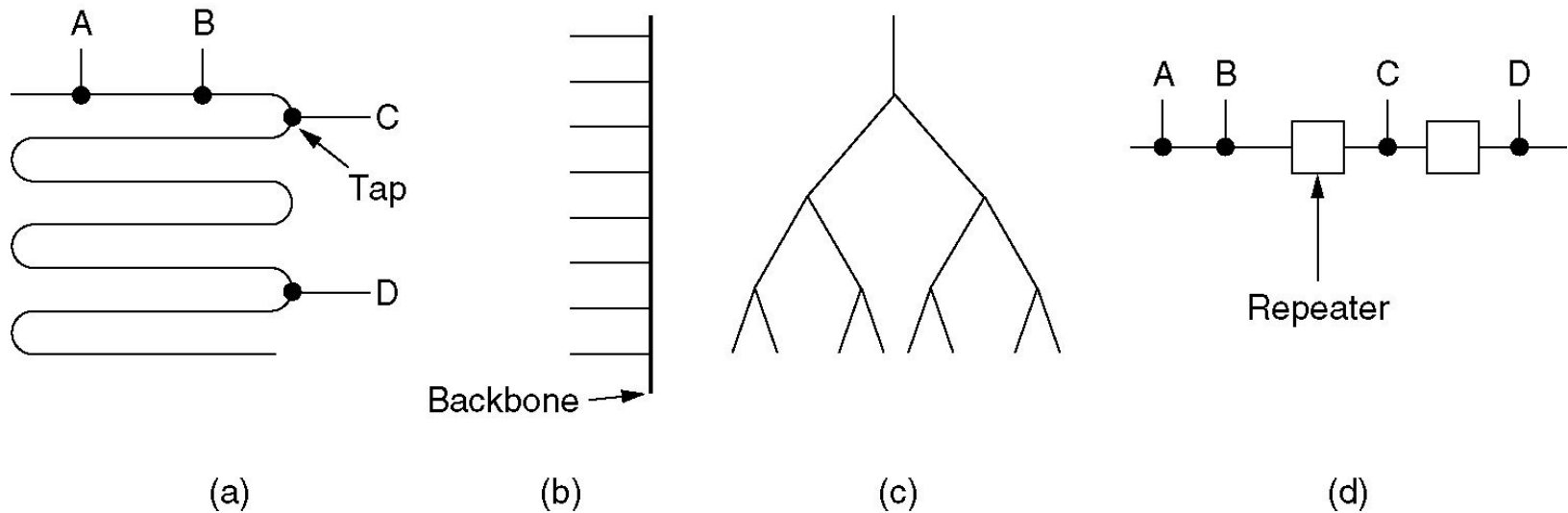
# Коаксиални кабели и хъб



Три вида Ethernet cabling.

(a) 10Base5, (b) 10Base2, (c) 10Base-T.

# Топологии на Ethernet окабеляване



(a) Шина, (b) вертикално, (c) Дървоводна, (d) Сегментирана.



# Bridge и switch

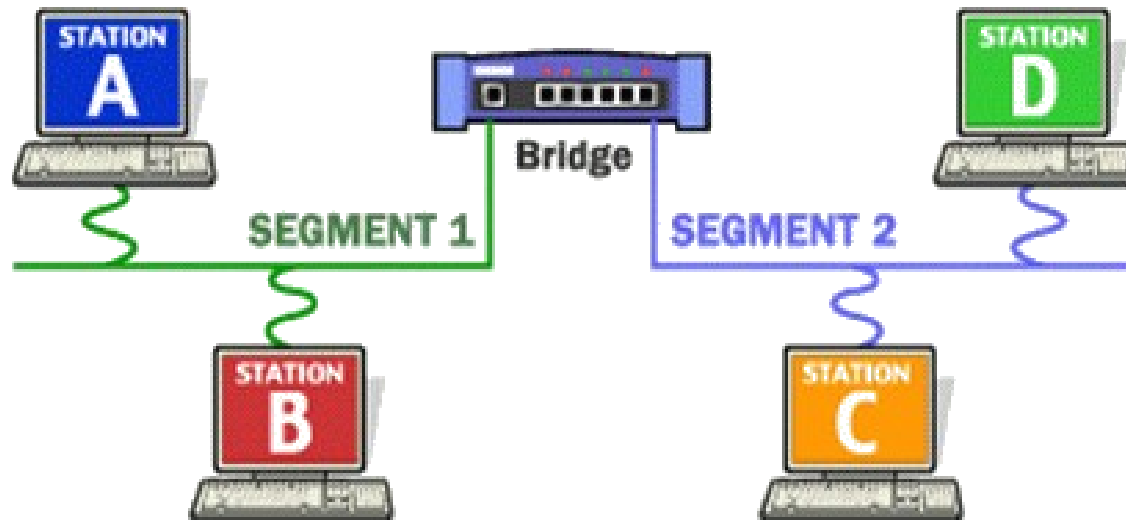
**Мостът (bridge)** работи на **канално ниво** и служи за свързване на две локални мрежи. За разлика от повторителите и хъбовете, мостът анализира получените кадри.

Той прочита адреса на получателя и по него определя към коя изходна линия да изпрати кадъра (за целта се поддържа специална таблица).

Мостът предава кадъра само към определената от него изходна линия, а не по всички изходни линии.

Подобно устройство е **превключвателят (switch)** – многопортов мост. Той също прочита адресите на постъпилите в него кадри.

# Bridge и switch



# Bridge и switch

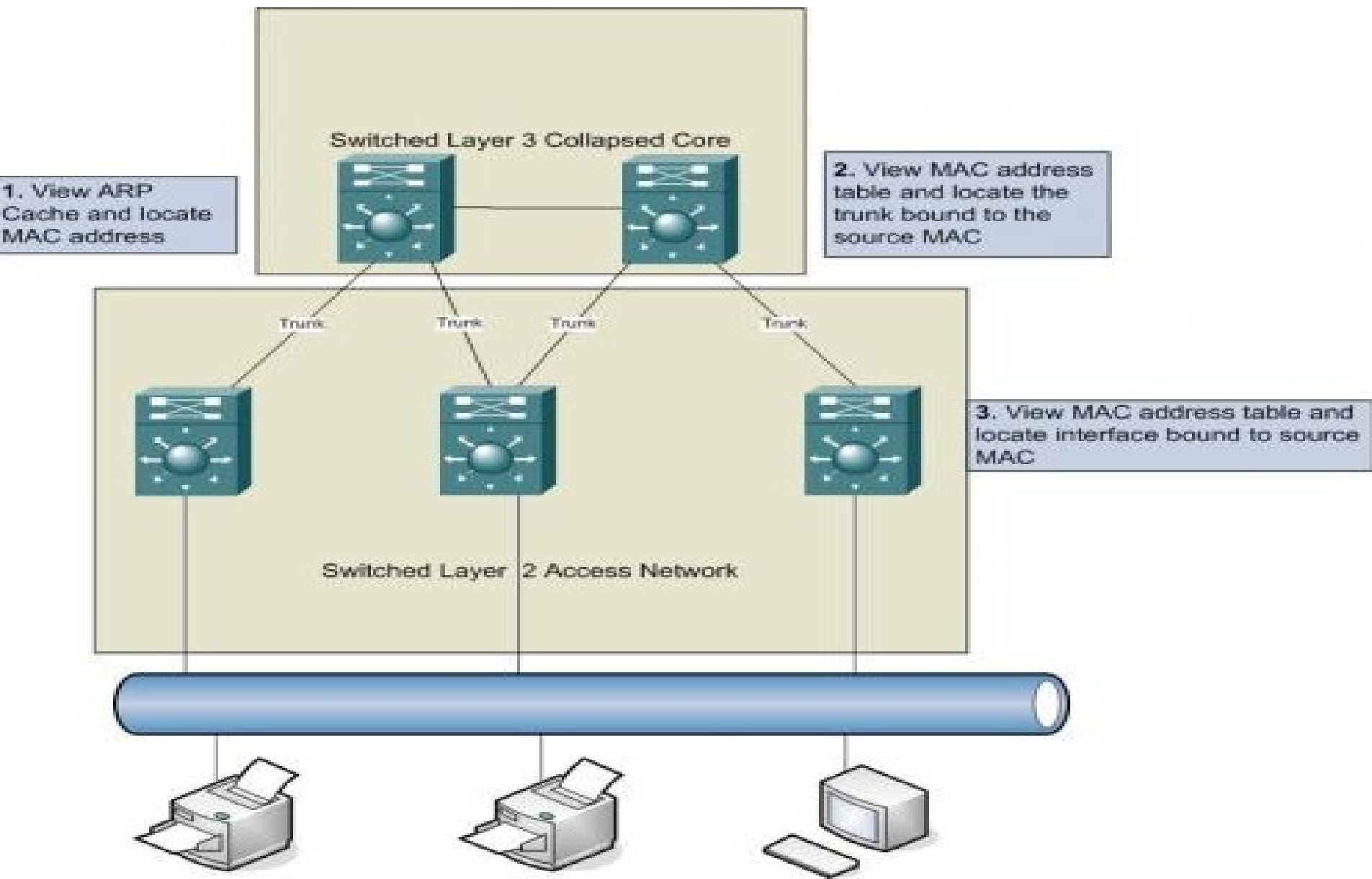
Всяка линия (порт) е самостоятелна и представлява **отделен колизионен домейн**. Това се нарича още **микросегментиране**.

Но бродкастите се разпространяват по всички портове, т.е той е **един бродкаст домейн**.

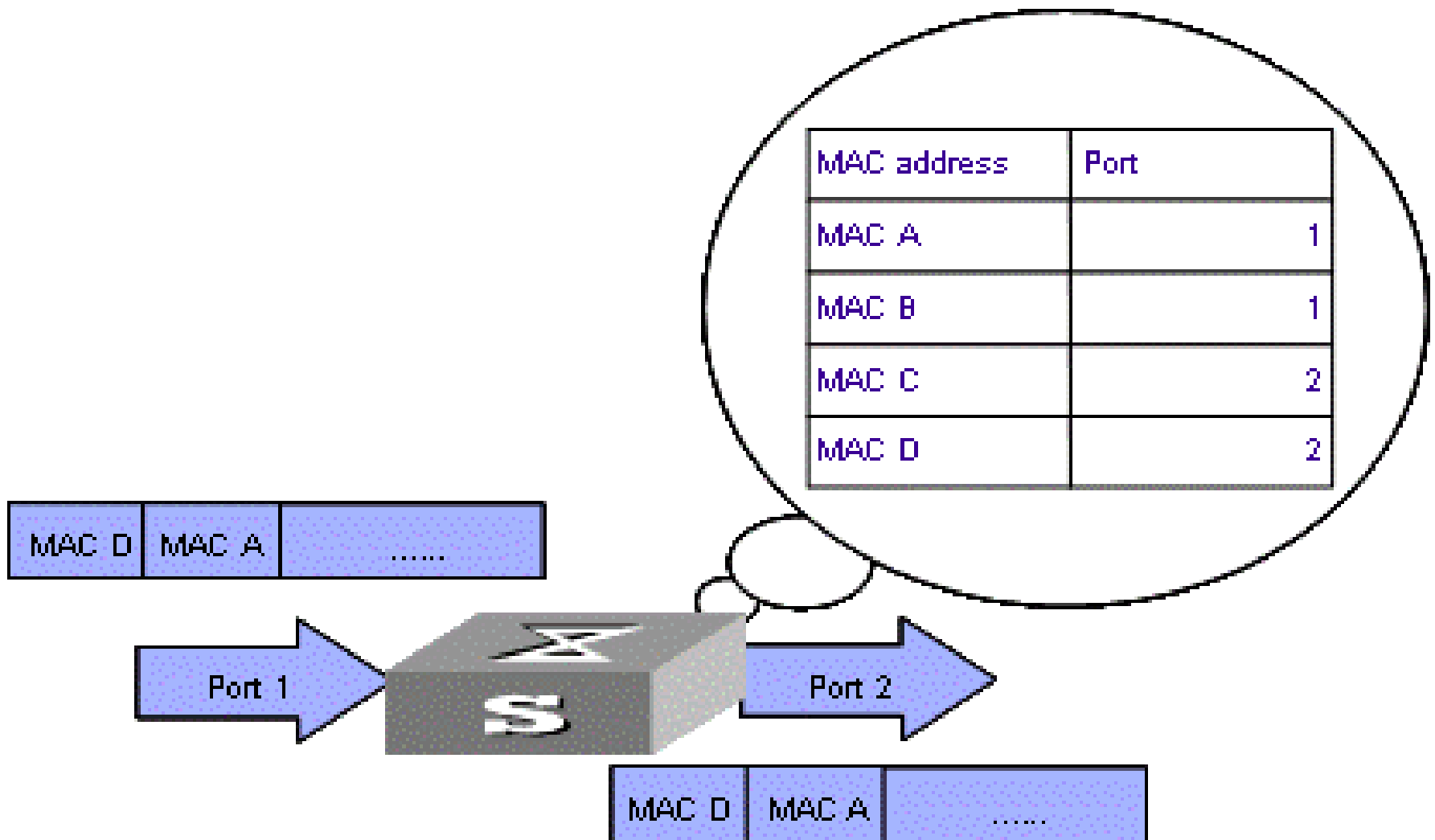
При превключване между сегменти кадри не могат да бъдат изгубени поради колизии.

За целта превключвателя трябва да има достатъчно буферно пространство за да може да се препращат кадрите.

# Switched Ethernet



# MAC Address Table



# Три режима на превключване

С пълно буфериране (**store and forward**). В буферната памет се записва целия кадър и чак след това се превключва към изходния порт. Внася се закъснение и изисква повече памет.

**Cut-through** – Суичът прочита адреса на получателя при получаване на кадъра. Започва прехвърлянето към изходящия порт, преди да получи пълния кадър. Така се намалява закъснението. Имаме **две форми на cut-through**:

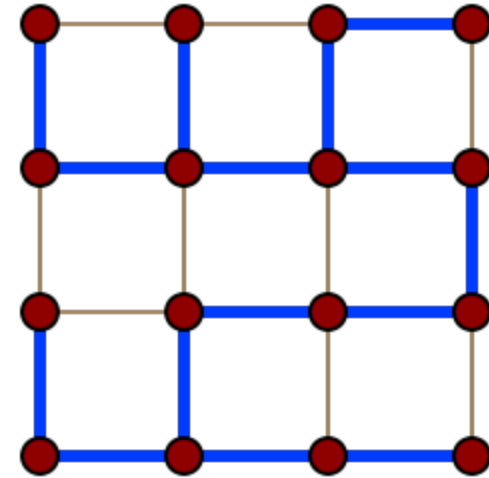
**Fast-forward** – С най-ниско закъснение, веднага превключва кадъра след приемане на адреса на получателя. Има проблеми с откриването на грешки.

**Fragment-free** - Филтрира кадри (фрагменти), претърпели колизии, най-често срещаните грешки. Обикновено това са кадри с дължина, по-малка от 64 байта. Т.е. прочита първите 64 бита, за да определи дали това не е колизионен фрагмент, преди да започне превключването.

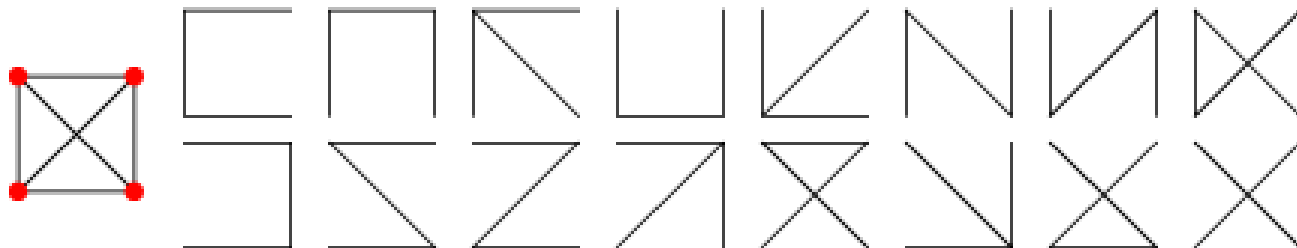
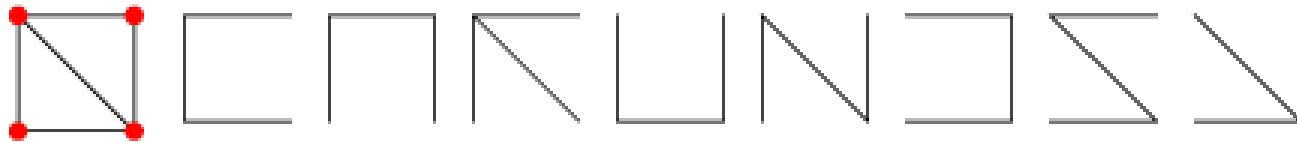
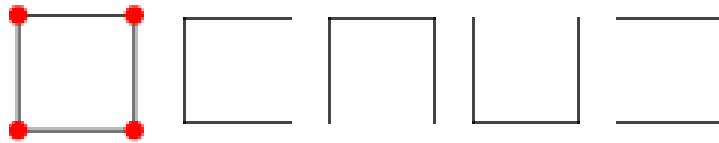
# Spanning Tree (математика)

spanning tree (разперено дърво) на граф  $G$ :  
сбор от клони на  $G$ , които  
формират дърво,  
разперващо се *spanning* от  
всеки връх.

Т.е всеки връх е в дървото,  
но няма зацикляне (*no loops*).

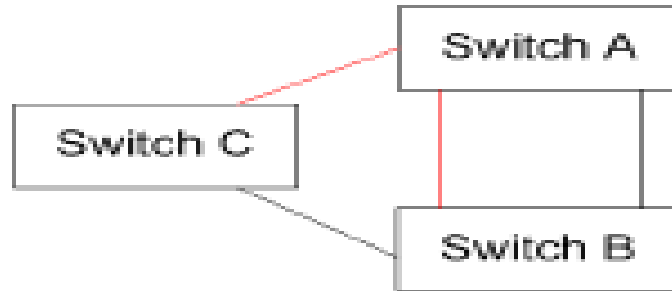


# Spanning Tree (математика)





# Spanning Tree. Защо?



Резервираност в топологията:

**PROS:** откзоустойчивост, по-висока производителност.

**CONS:** switched LAN: **broadcast storms**: задръстване на MAC таблицата и др.

Хост изпраща broadcast кадър. Разпространява се по всички портове с изключение на входящия.

broadcast се “завърта” през паралелните uplink портове – до безкрайност, защо

layer 2 няма TTL да го спре. Решението е: Spanning Tree Protocol.

# Spanning Tree Protocol (STP)

**Spanning Tree Protocol (STP)** е протокол на 2 слой по модела на OSI, който гарантира **топология без зацикляне** в Switched LAN. Базира се на алгоритъма на Radia Perlman, който е работил за Digital Equipment Corporation.

Позволява да се включват резервни пътища, които автоматично да се активират при авария в основните без опасност от зацикляне.

Зациклянето в тези мрежи е опасно заради липсата на механизъм TTL, както ще видим в IP протокола на 3 слой.

**STP** се дефинира в стандарта **IEEE 802.1D**.

# STP - стоимости

Скорост (Data rate)	(STP Cost – 802.1D-1998)	(802.1t-2001)
4 Mbit/s	250	5,000,000
10 Mbit/s	100	2,000,000
16 Mbit/s	62	1,250,000
100 Mbit/s	19	200,000
1 Gbit/s	4	20,000
2 Gbit/s	3	10,000
10 Gbit/s	2	2000

# Spanning Tree - алгоритъм

Spanning Tree алгоритъмът изчислява път без зацикляне.

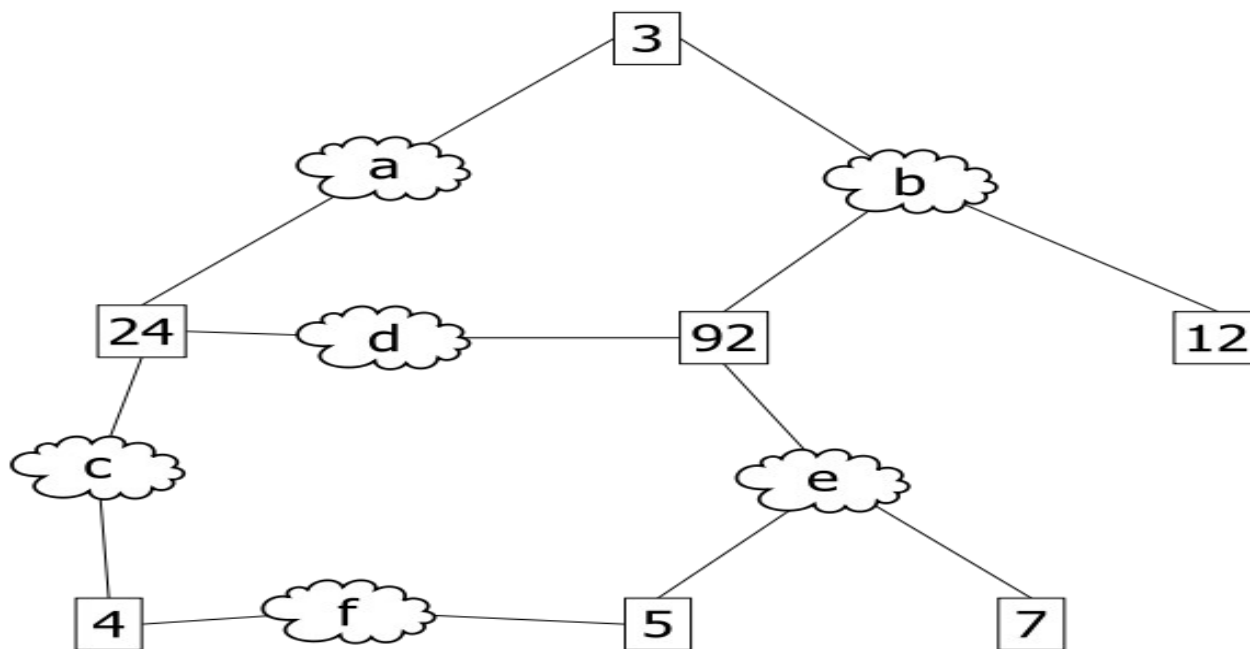
Първоначално всички портове са блокирани. Отнема около 50 s, докато започнат да превключват.

**Стъпка 1 : Избор на Root Bridge** – с най-нисък приоритет или най-ниско bridge ID (MAC адрес)

**Стъпка 2 : Избор на Root Ports** – От алтернативните пътища се избират тези с най-малка стойност до Root Bridge. RPs водят към root bridge.

**Стъпка 3 : Избор на Designated Ports** – Порт, който праща и получава трафик от Root Bridge – с най-ниска стойност до Root Bridge. DPs водят от root bridge към клоните на дървото.

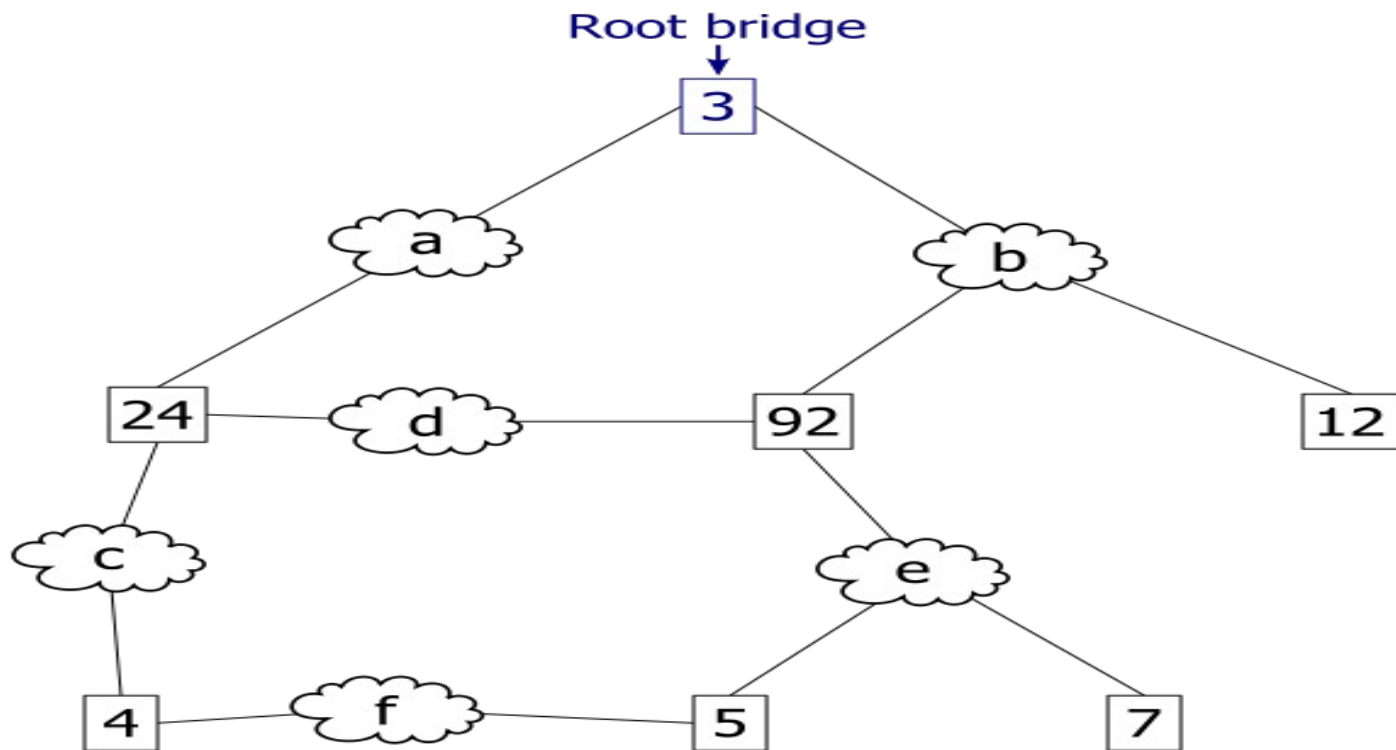
# Пример



Номерираните кутийки - **bridge ID**.

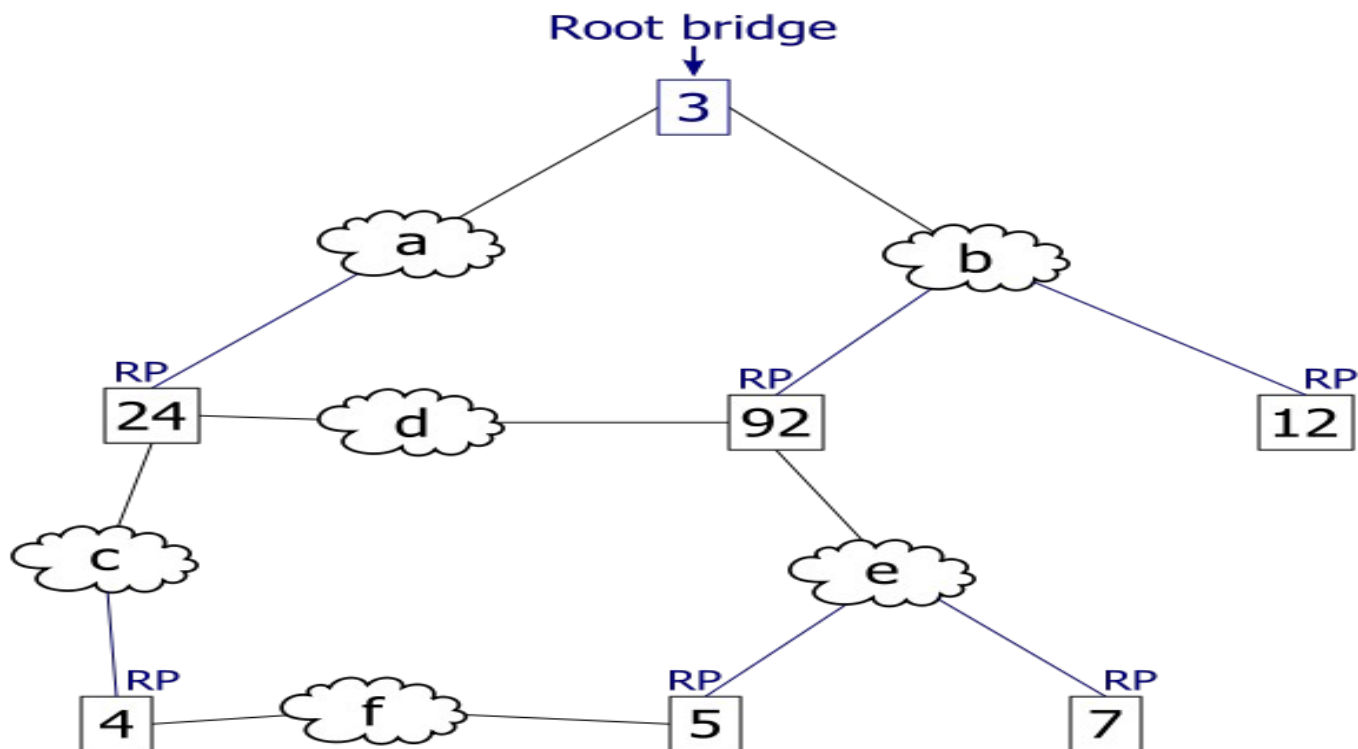
Номерираните облаци – **мрежови сегменти**.

# Избор на root bridge



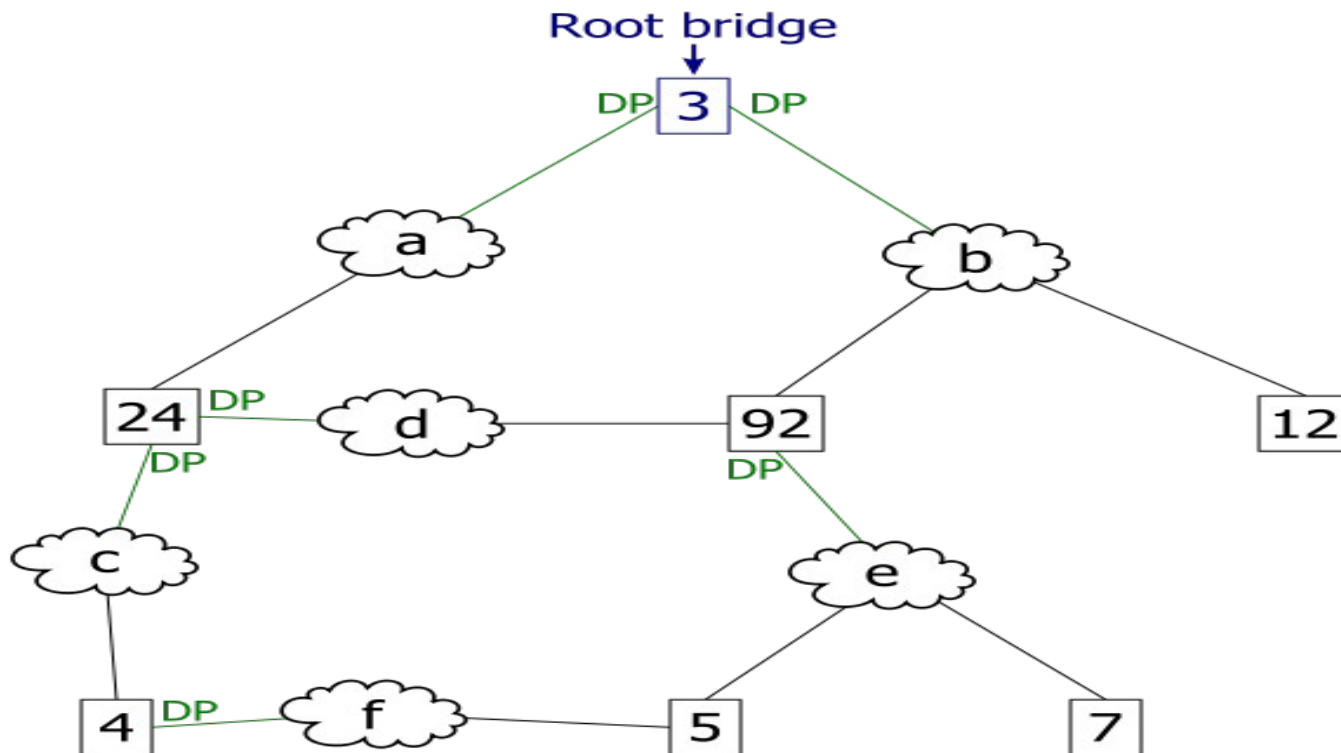
Най-малкият bridge ID е 3

# Избор на root port



Предполагаме, че **стойността** на всеки сегмент е **1**. Най-късият път от **bridge 4** до **root bridge** минава през **сегмент c**.

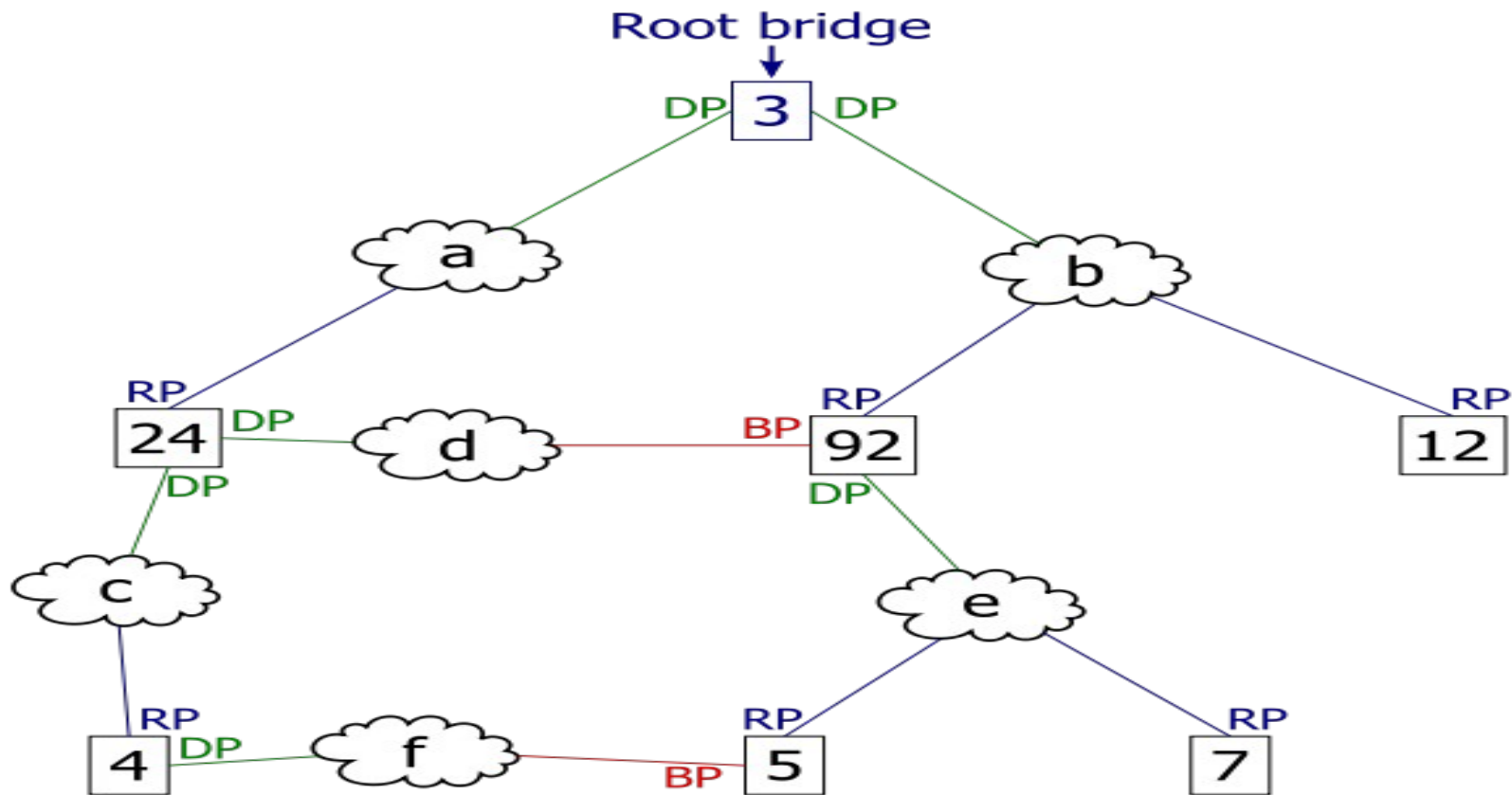
# Избор на designated port



Най-късият (с най-малка стойност) път до root от мрежов сегмент **e** минава през bridge 92.



# Spanning Tree - резултат



Активни портове, които не са root port или designated port са блокирани (blocked port).

# Виртуални ЛМ (Virtual LANs)

VLAN е комутирана мрежа, която е **логически сегментирана** по някакви функции и не се влияе от физическото разположение на потребителите (по етажи, сгради и т.н.).

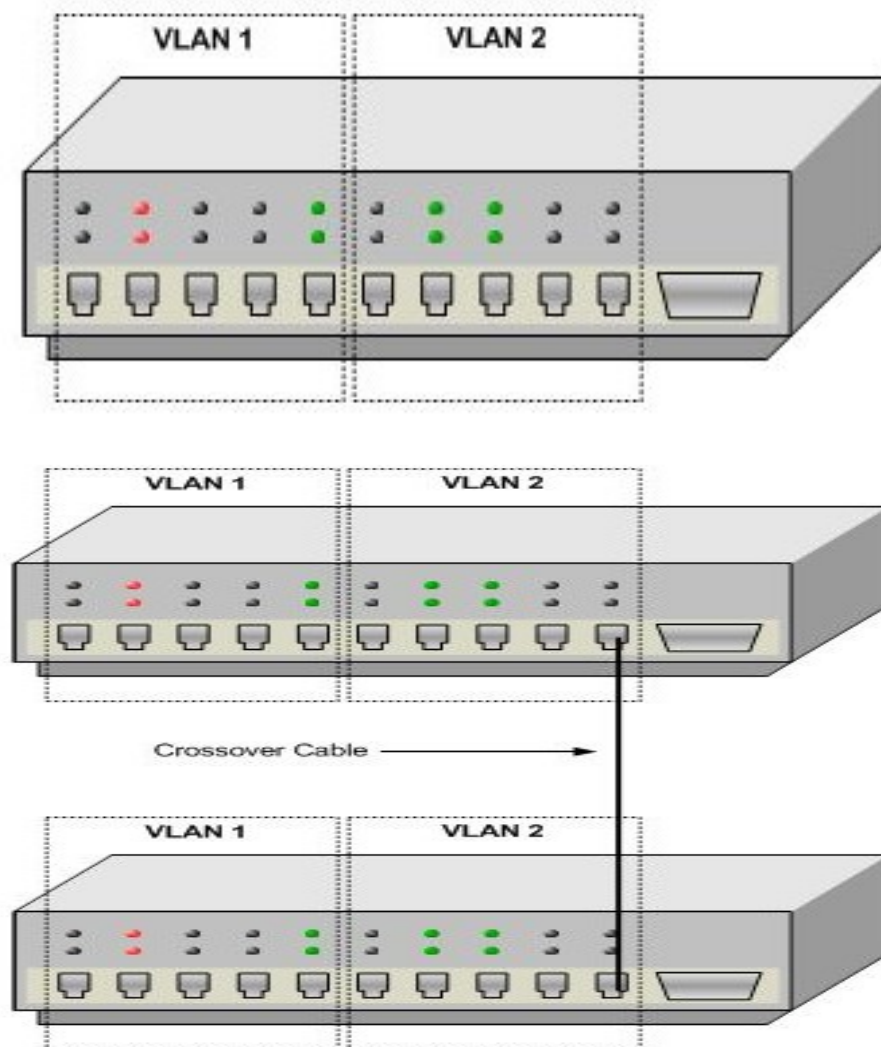
Един VLAN представлява един **broadcast domain**.

**Сигурност.** Потребителите на VLAN<sub>i</sub> нямат достъп до машините на VLAN<sub>j</sub>. Това може да стане **единствено и само през рутер**.

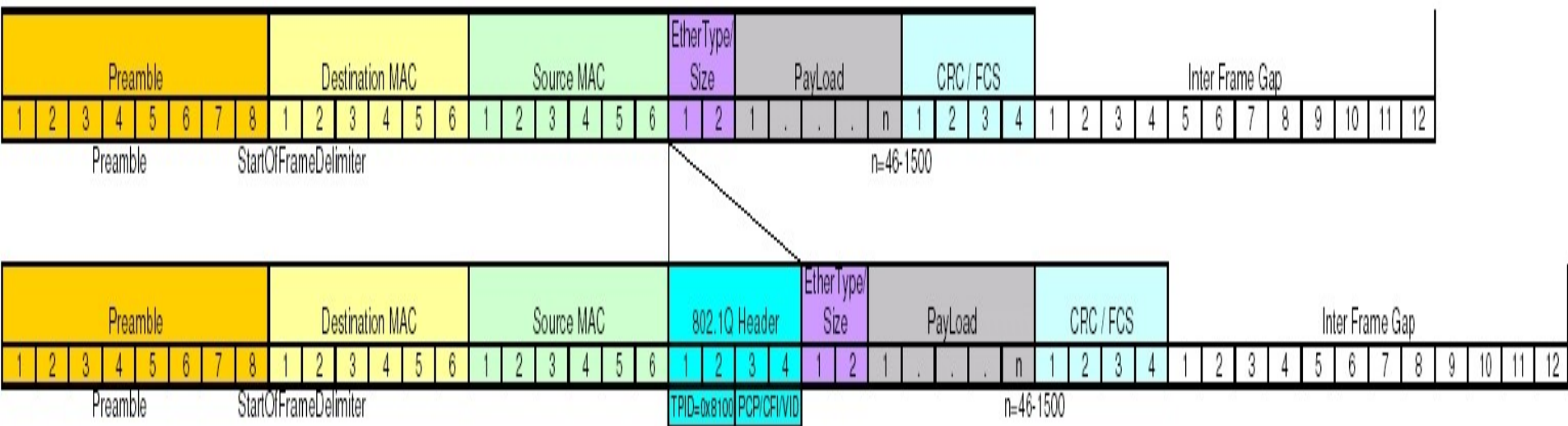
**Гъвкавост.** Опростява местене, добавяне, премахване на потребителски машини.

Един порт на суич може да се присвои статично или динамично към VLAN.

**Trunk** портове за връзка между суичове.



# VLANs – 802.1Q Tag



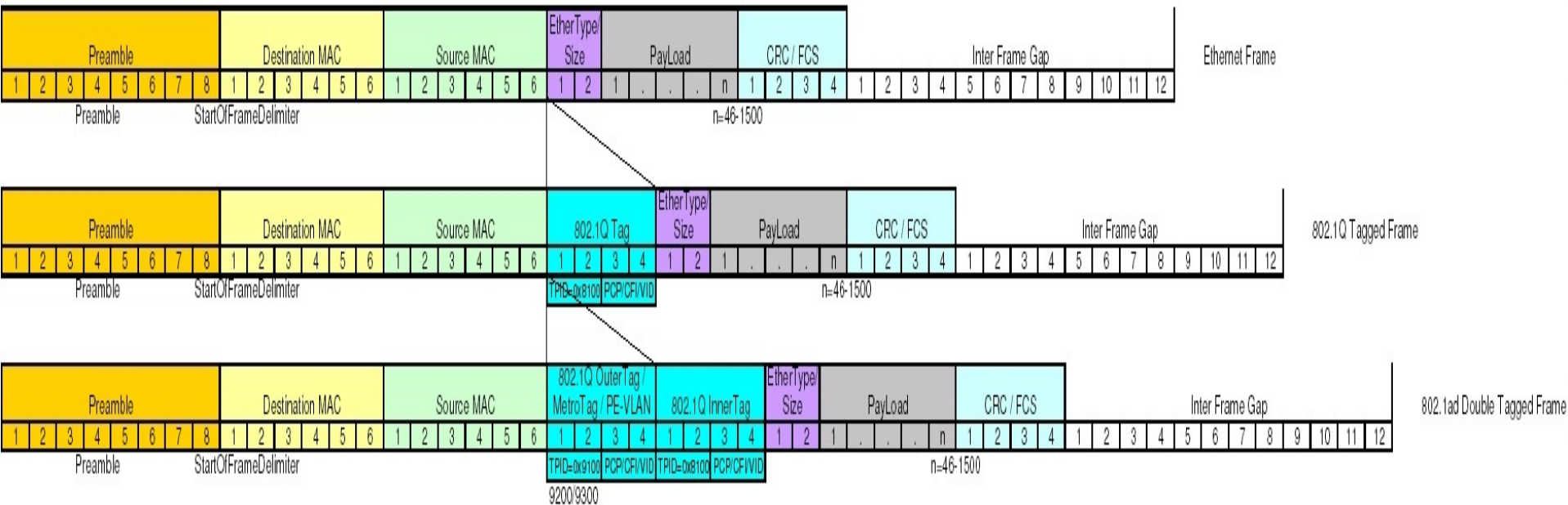
**Tag Protocol Identifier (TPID):** 16-битово поле: 0x8100 (IEEE 802.1Q)

**Priority Code Point (PCP):** 3-бита - IEEE 802.1p приоритет: 0 (най-ниско) до 7

**Canonical Format Indicator (CFI):** 1-бит: “0” за Ethernet суичове

**VLAN Identifier (VID):** 12 бита. ако е “0”, кадърът не във VLAN; позволява до 4094 VLAN-а. VLAN 1 резервирана за управление.

# VLANs – (QinQ)



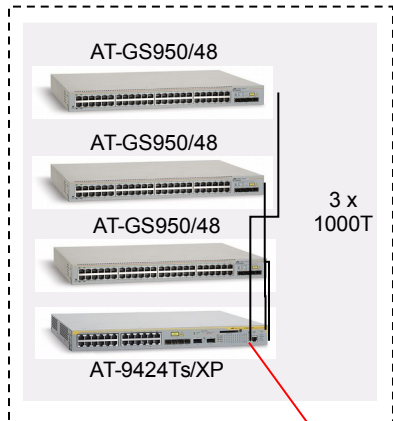
**Double-tagging (QinQ)** се използва от ISPs и MAN оператори, както и техните клиенти, да прокарват вътрешни VLAN-и през външен VLAN.

Външен tag предхожда **вътрешен tag**.

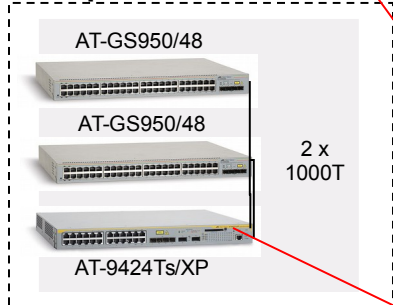
TPID - hex 9100, 9200 или 9300 за външния; но **802.1ad** определя **88a8** за външни тагове.

# Физическа топология на мрежа

**2 ет. – север, 155 р.м.**

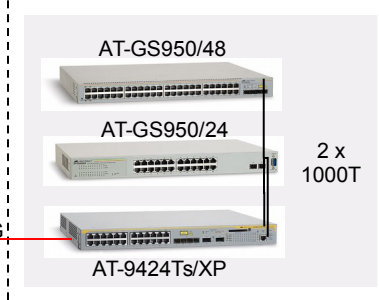


**1 ет. – север, 66 р.м.  
+ 46 р.м. в мазе**

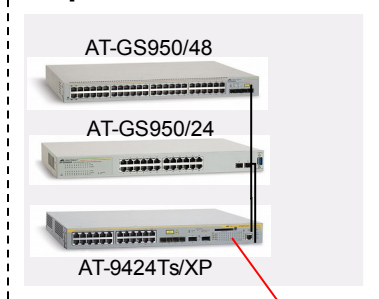


8 x 10G  
към 3, 4 и 5 ет.

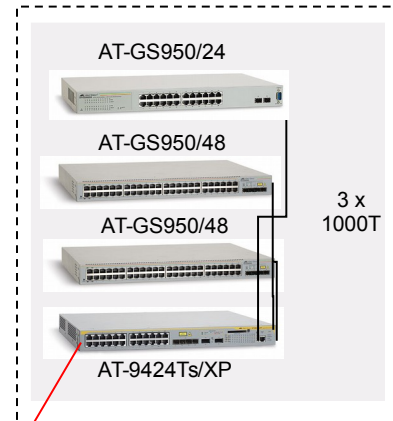
**2 ет. - център, 84 р.м.**



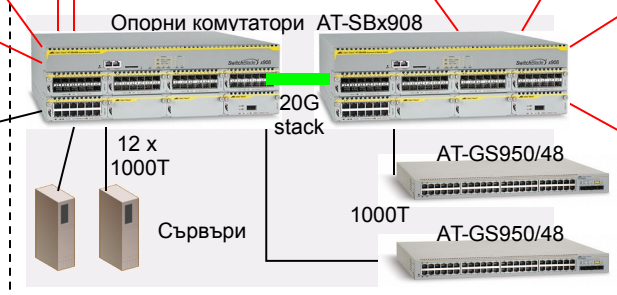
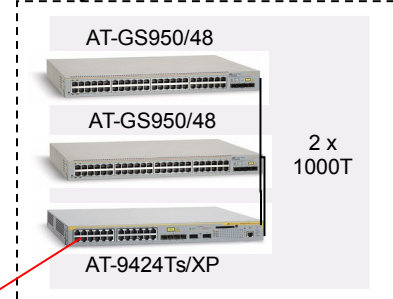
**1 ет. - център 76 р.м.  
+ 2 р.м. в мазе**



**2 ет. – север, 119 р.м.**

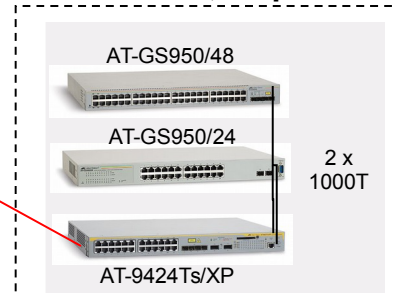


**1 ет. - юг, 109 р.м.  
+ 8 р.м. в мазе**

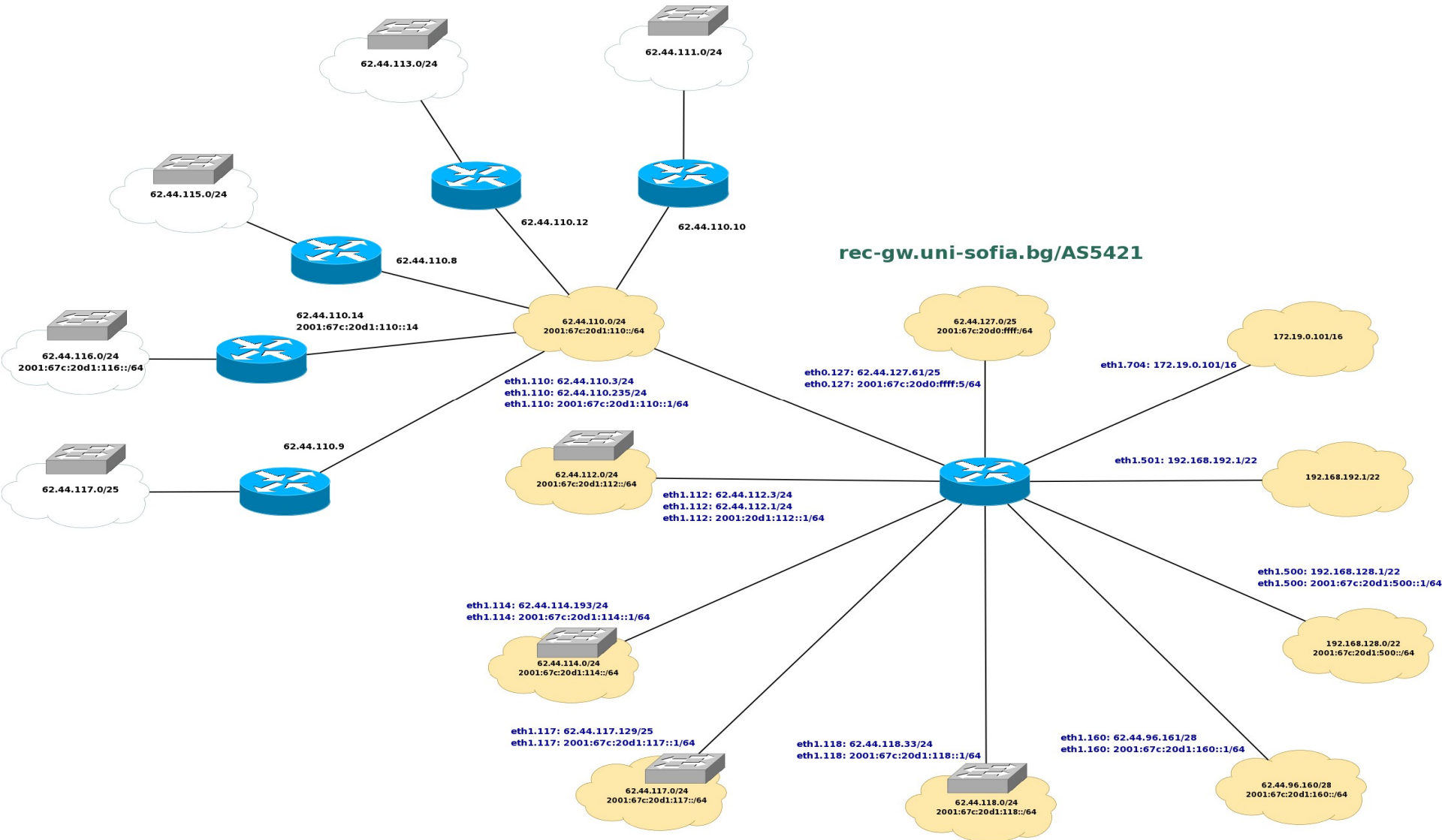


**1 ет., стая 234, 74 р.м.**

**библиотека, 88 р.м.**



# Логическа топология на същата мрежа с VTENs



# Конфигуриране на 802.1q VLAN в Red Hat Enterprise Linux и CentOS

## 1. Активиране на поддръжката на 802.1q VLAN в системата

Поддръжката на стандарта 802.1q се осигурява от модула 8021q на ядрото. Лесно може да се провери дали е наличен:

**\$ /sbin/modinfo 8021q**

```
filename:          /lib/modules/2.6.32-  
220.2.1.el6.i686/kernel/net/8021q/8021q.ko  
version:           1.8  
license:           GPL  
alias:             rtnl-link-vlan  
srcversion:        2763A7682EB87280D0F56F2  
depends:            garp  
vermagic:          2.6.32-220.2.1.el6.i686 SMP  
mod_unload modversions 686
```



# Конфигуриране на 802.1q VLAN в Red Hat Enterprise Linux и CentOS

```
Vi /etc/sysconfig/network
```

```
VLAN=yes
```

За да може да се извършва ръчна и автоматична настройка на 802.1q VLAN интерфейсите в системата:

```
# yum install vconfig
```

**2. Как се активираме поддръжката на 802.1q VLAN в работещи системи без да прекъсваме мрежовите функции**

Да заредим модула за поддръжка на 802.1q VLAN в ядрото на операционната система:

```
# /sbin/modprobe 8021q
```

```
$ grep 8021q /proc/modules
```

```
8021q 25929 1 cxgb3, Live 0xf8a3c000
```



# Конфигуриране на 802.1q VLAN в Red Hat Enterprise Linux и CentOS

## 3. Конфигуриране на 802.1q VLAN интерфейс

### **/etc/sysconfig/network-scripts/ifcfg-eth1.100**

DEVICE=eth1.100

HWADDR=00:00:CD:4A:55:6A

BOOTPROTO=none

IPADDR=10.10.10.1

NETMASK=255.255.255.0

ONBOOT=yes

REORDER\_HDR=yes

### **# ifup eth1.100**

Added VLAN with VID == 100 to IF -:eth1:-

# Carrier Ethernet

Как Ethernet да се ползва от телеком операторите:

**Ethernet over SDH/SONET.**

**Ethernet over MPLS.** Ethernet върху IP/MPLS мрежи. Ethernet се транспортира като “псевдожици” - MPLS label switched paths (**LSPs**) вътре в MPLS “тунел”. Поддържа връзки **точка-точка** (Virtual Private Wire Service - **VPWS**) и **многоточкови** (Virtual Private LAN service – **VPLS**).

**Конвенционална ("чиста") Ethernet.** Прилага 802.1w - Rapid Spanning Tree Protocol за връзки точка-точка.

# Carrier Ethernet 2.0



**Боб Меткалф**, сега съветник в MEF (**Metro Ethernet Forum**), обяви второ поколение Carrier Ethernet (**CE 2.0**).  
“... възможност за опериране с **до 8 услуги** (за сравнение CE 1.0 предлага само 3),  
2 от тях са разпределени в направленията E-Line, E-LAN, E-Tree и E-Access”