Ethernet – első labor

Moldován István

Budapest University of Technology and Economics

Department of
Telecommunications and Media Informatics

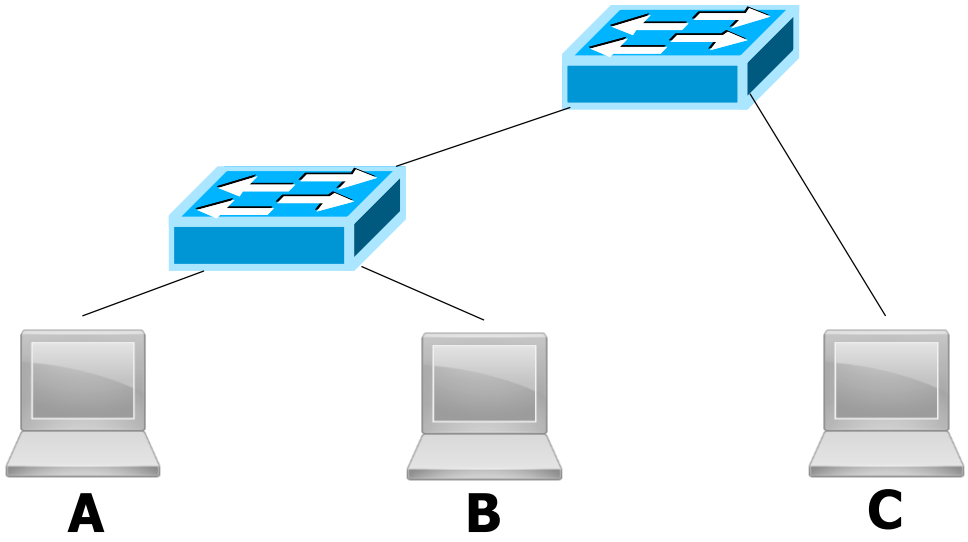
MAC címzés

- MAC címek
  - MM:MM:MM:SS:SS:SS
  - MM-MM-MM-SS-SS-SS
  - An OUI {Organizationally Unique Identifier}
    - 00:00:0A -- this is owned by Omron
    - 00-0D-4B -- this is owned by Roku, LLC
  - https://www.wireshark.org/tools/oui-lookup.html
- Broadcast: FF:FF:FF:FF:FF:FF
- Multicast

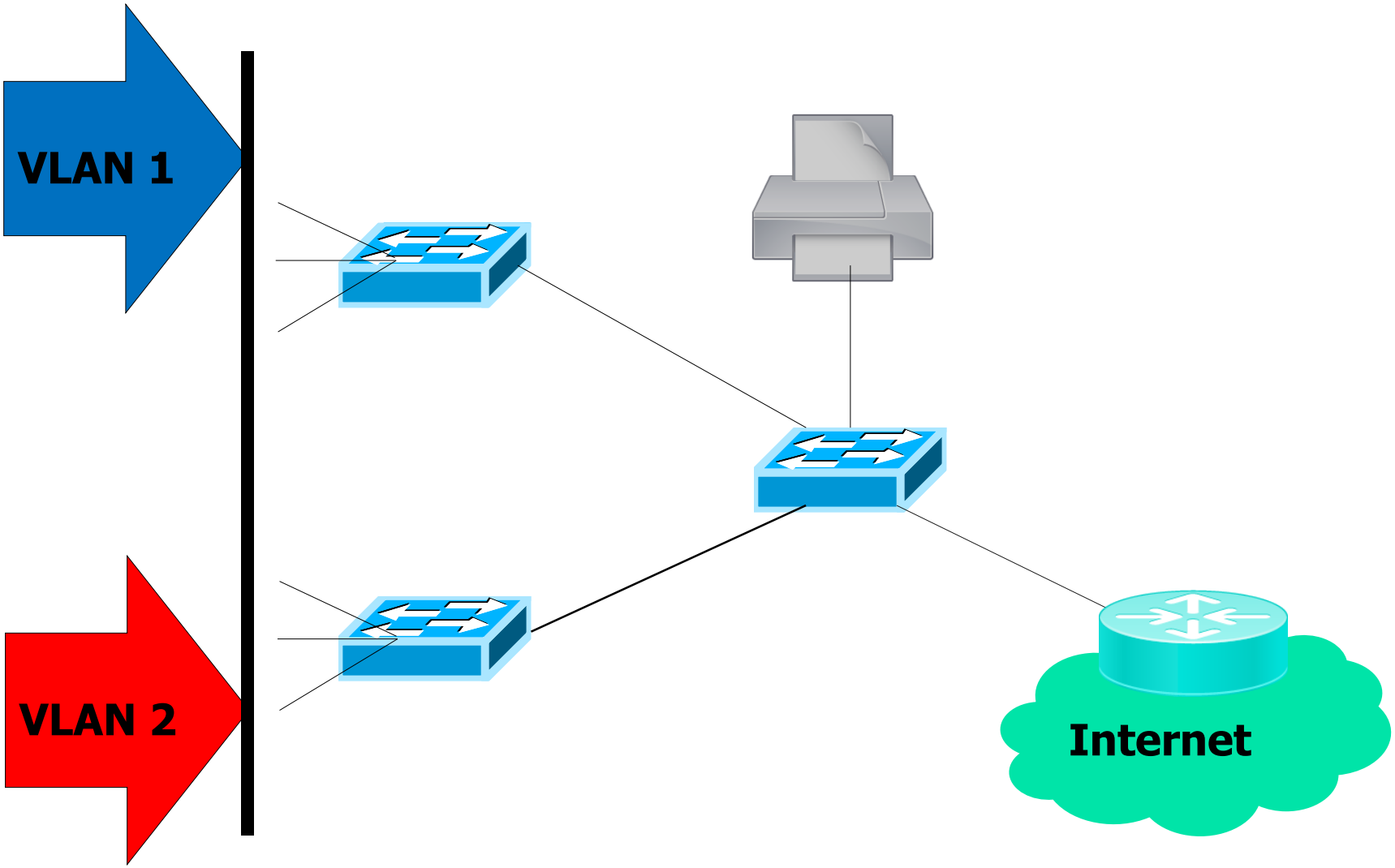
Keret formátum

- Ethernet II vs 802.3 – mikor melyik?
  - Type/length
  - packETH / wireshark

- Tanulás
  - Példa



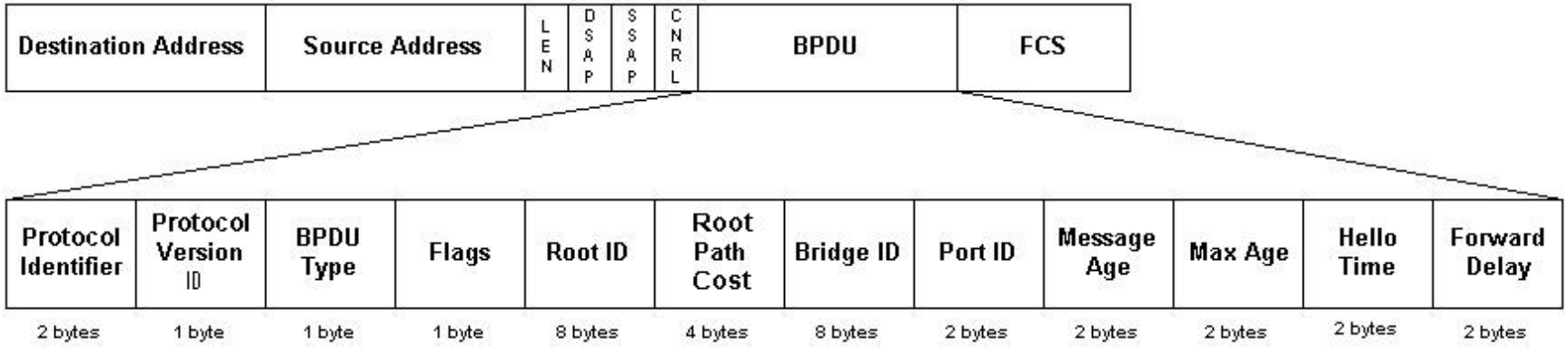
VLAN

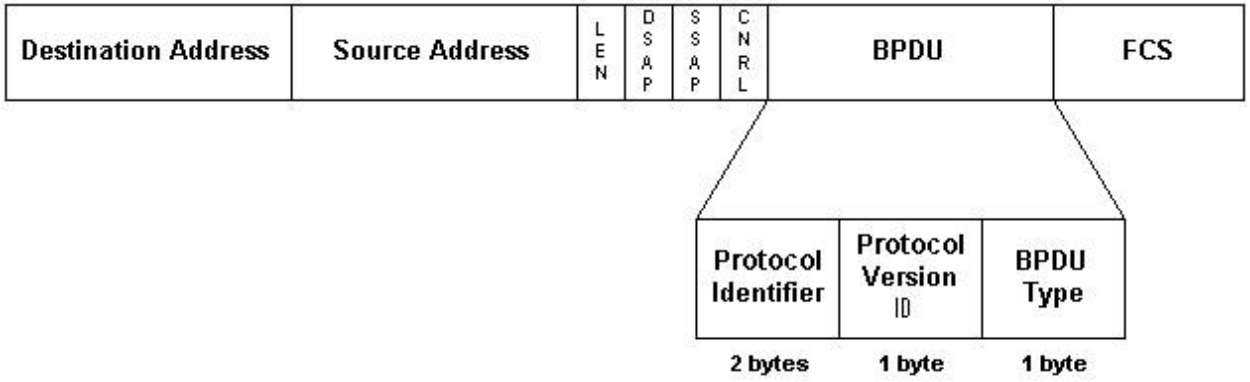


Spanning Tree

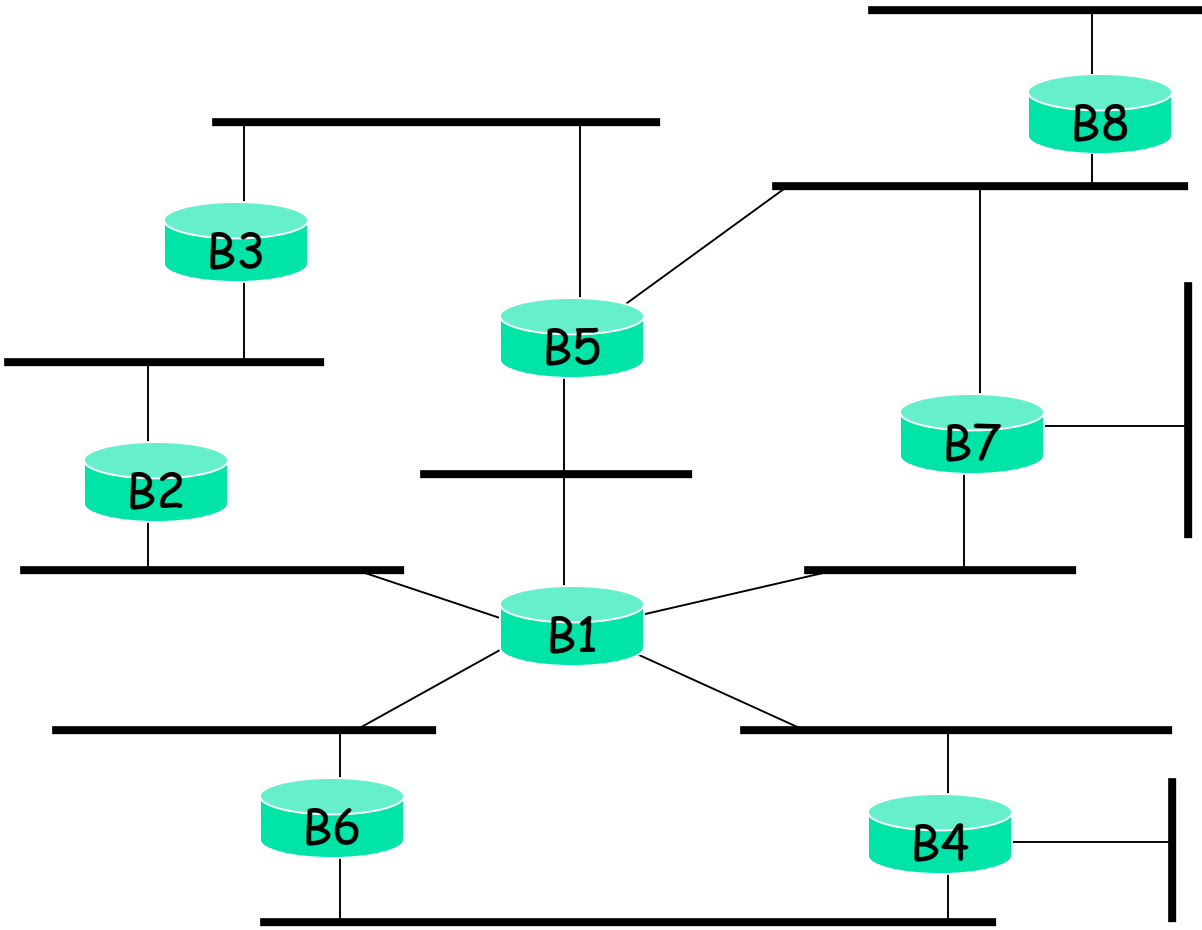
- RSTP fa kialakítása
  - BPDU-k
- Paraméterek:
  - Link speed
  - Prioritás
  - ID
- Bridge ID: Prioritás + MAC

Bridge BPDU-k





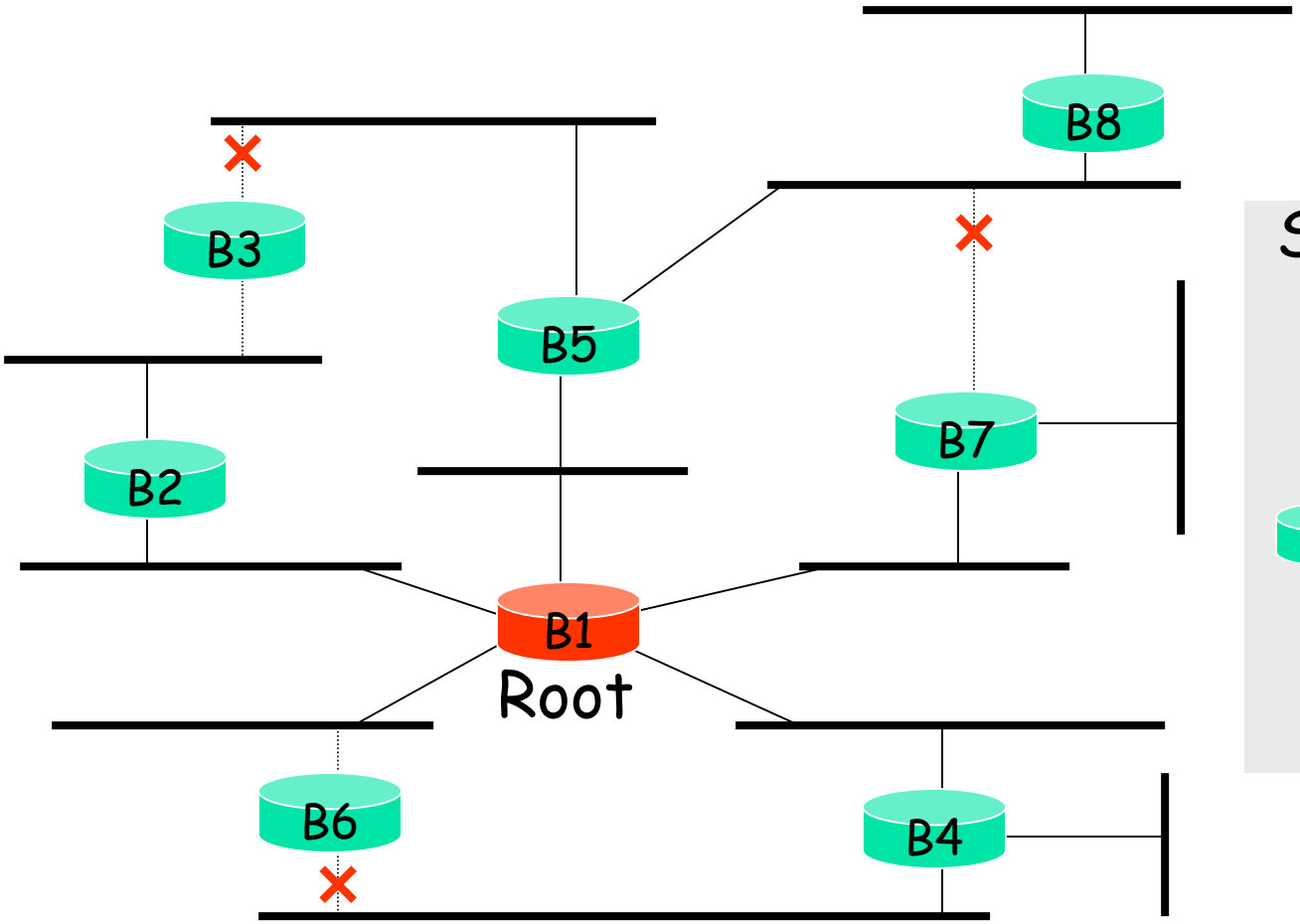
Példa – Fizikai topológia



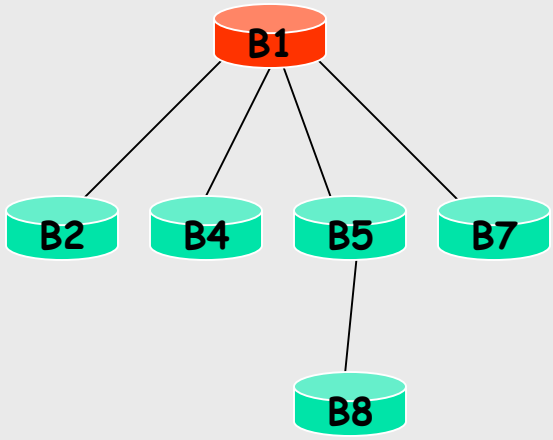
Protocol működés:

1. Root kiválasztás
2. minden LAN-ra kiválasztja a designated bridge-et, a legközelebbit a root-hoz.
3. Minden bridge a root fele a designated bridge-en keresztül küld.

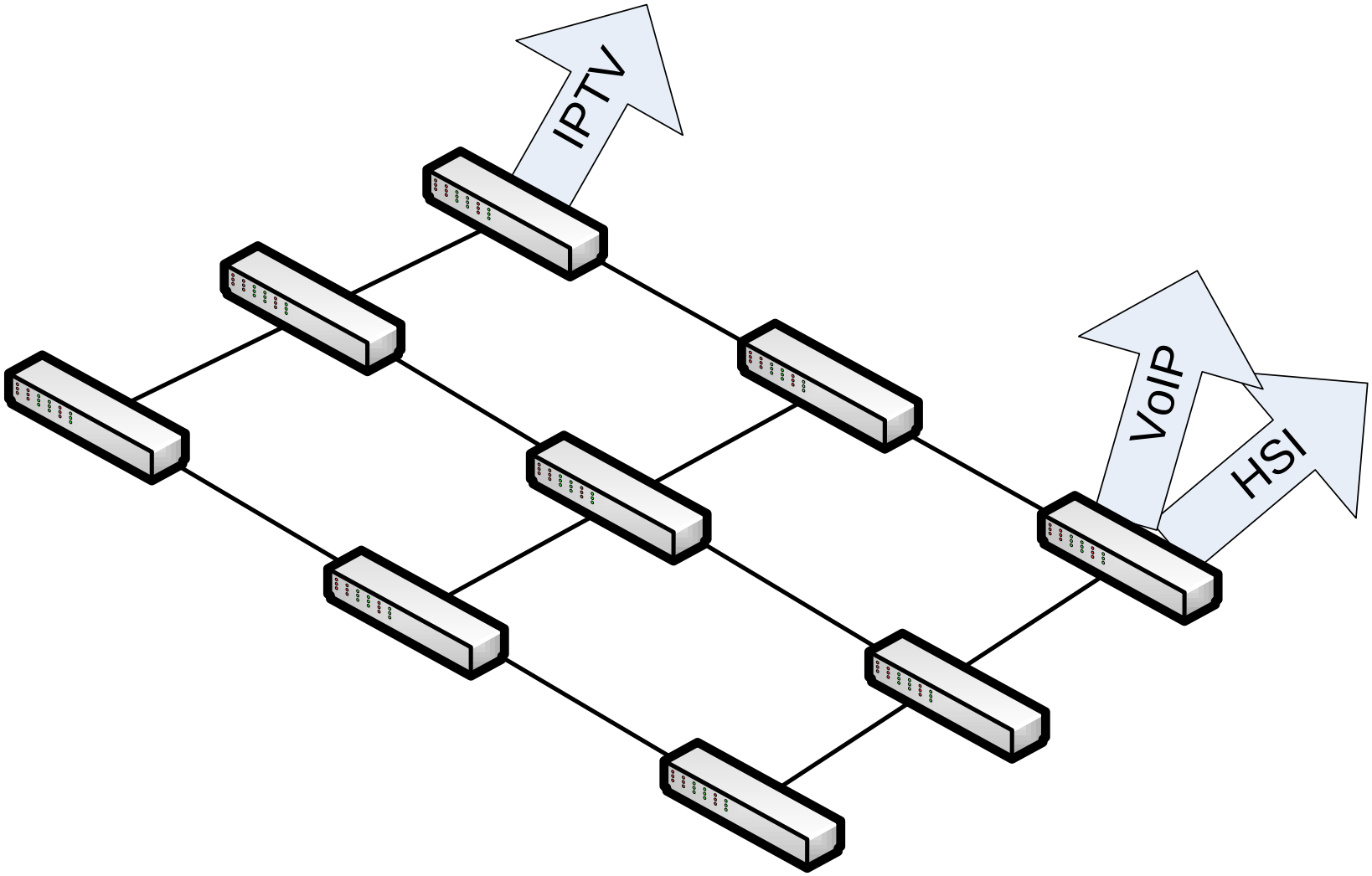
Példa – STP Topológia



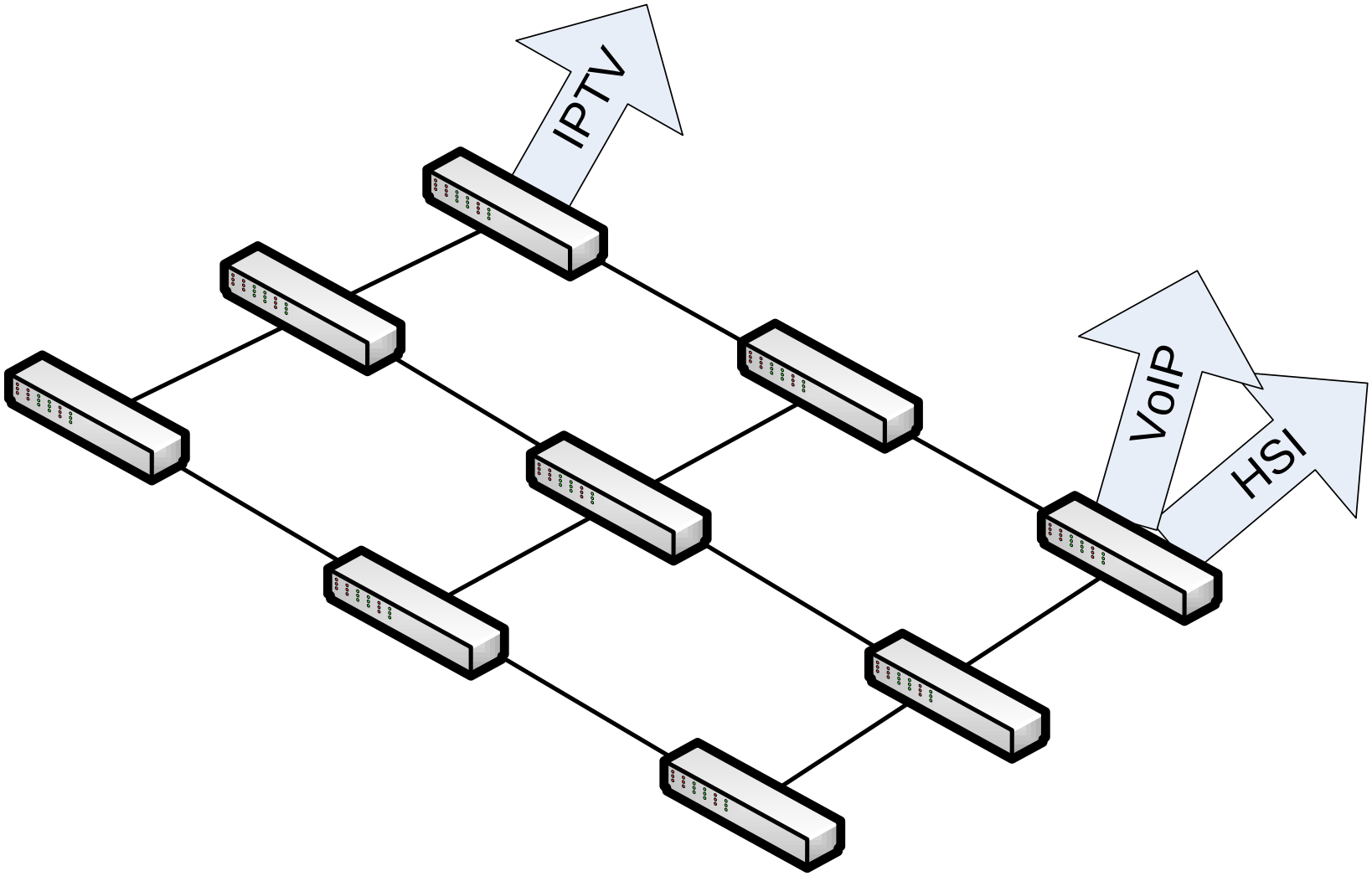
Spanning Tree:



RSTP optimális beállítás



MSTP optimális beállítás



Ethernet Switchek

- Layer 2 továbbítás – MAC címek alapján
- Megtanulja a MAC címeket akár a bridge
- Erőssége a store-and-forward működés amely egyidejű továbbítást végezhet különböző portok között
  - Nincs ütközés
  - Nagy sebességű *backplane*
- Nagyszámú interfész
  - Különböző sebességek/médiumok

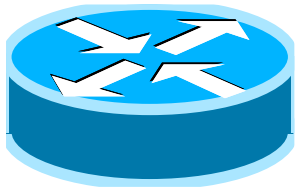
Switchek

- A nagyobb teljesítményű kapcsolók egyben bridge-ek is
  - Támogatják az STP protokollt
- Típusok
  - Nem menedzselhető:
    - irodai célokra, HUB-ok összekapcsolására kiváló
    - Nem támogatja az STP protokollt, sem a VLAN-okat (jövő óra)
  - Menedzselhető
    - VLAN és STP támogatás
    - Menedzsment interfész
  - L2/L3

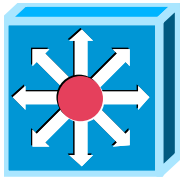
Ethernet alapú lokális hálózatok tervezése

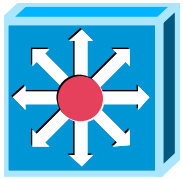
- Hierarchikusan

Router



Nagy teljesítményű, multiservice kapcsoló





SWITCH, Bridge

HUB

Kérdések?

Budapest University of Technology and Economics

Department of
Telecommunications and Media Informatics

Hálózati Technológiák és Alkalmazások

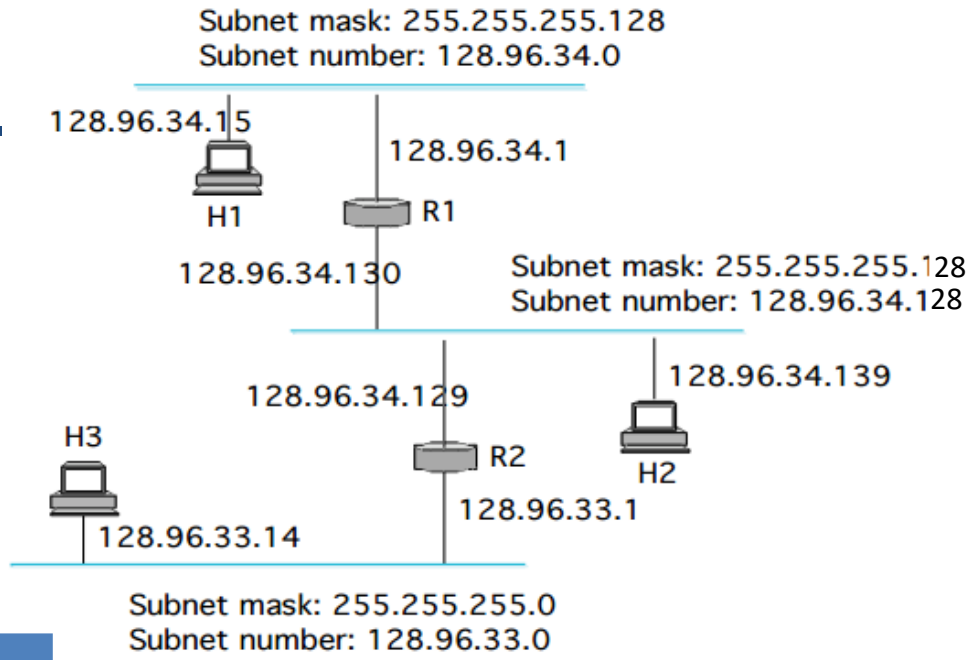
Vida Rolland
BME TMIT

2019. november 7.

1. feladat

Írjuk fel az R2 routing tábláját, a következő formátumban

Dest | NextHop | Subnet Mask

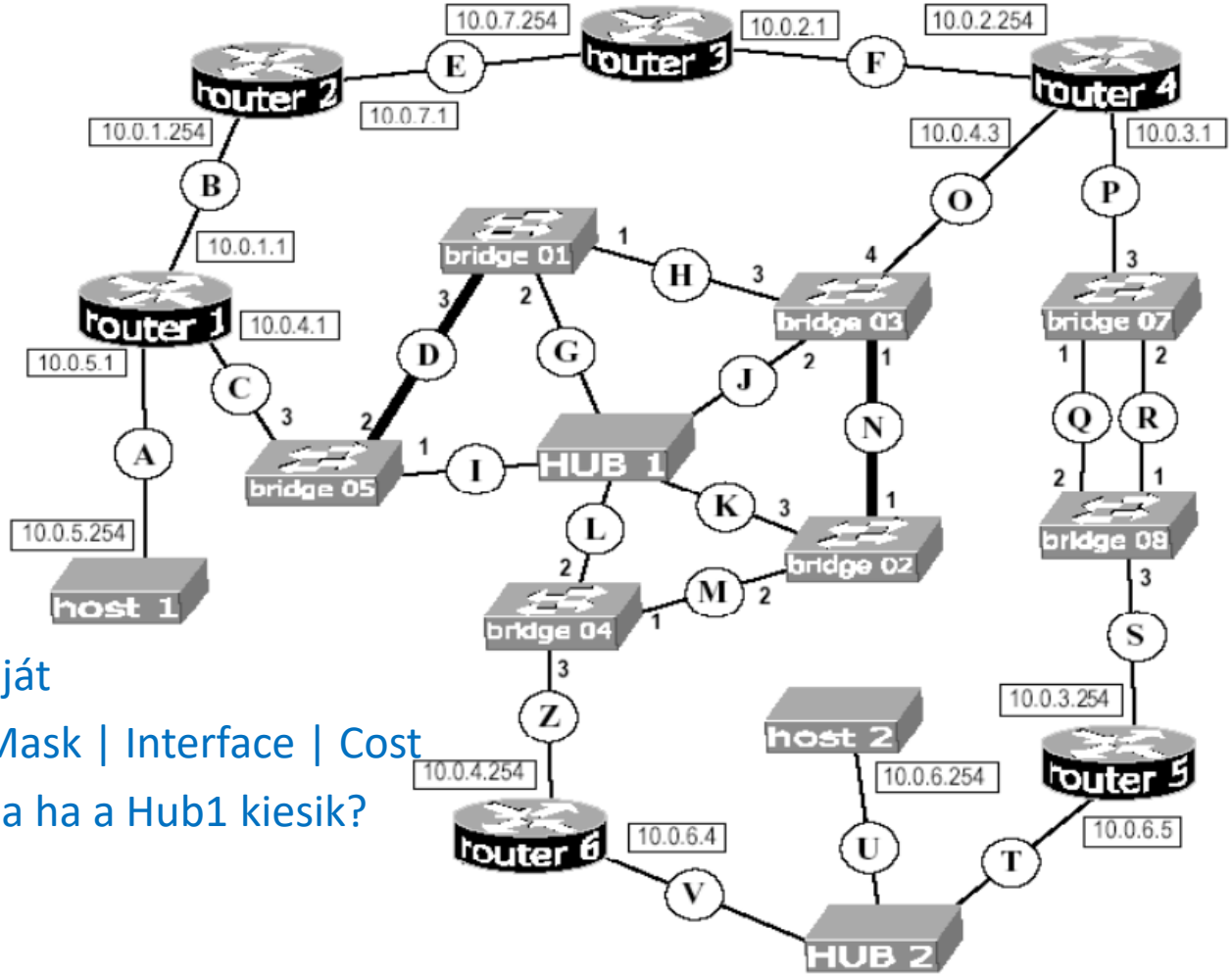


Megoldás

DestNet	Next Hop	Subnet Mask
128.96.34.128	Direct	255.255.255.128
128.96.33.0	Direct	255.255.255.0
128.96.34.0	128.96.34.130	255.255.255.128

2. feladat

- RIP routing



Írjuk fel az R1 útválasztó tábláját

DestNet | NextHop | NetMask | Interface | Cost

Hogyan változik a routing tábla ha a Hub1 kiesik?

2. feladat - megoldás

DestNet	NextHop	Netmask	Interface	HopCount
10.0.5.0	Direct	255.255.255.0	A	0
10.0.1.0	Direct	255.255.255.0	B	0
10.0.4.0	Direct	255.255.255.0	C	0
10.0.7.0	10.0.1.254	255.255.255.0	B	1
10.0.2.0	10.0.4.3	255.255.255.0	C	1
10.0.6.0	10.0.4.254	255.255.255.0	C	1
10.0.3.0	10.0.4.3	255.255.255.0	C	1

- Ha a Hub1 kiesik, nem történik semmi!
  - Léteznek alternatív útvonalak a 10.0.4.0 hálózaton

3. feladat

Az alábbi routing táblát hogyan lehet tömöríteni CIDR-t használva?

DestNet	NetMask	Interfész
194.100.0.0	255.255.255.0	I1
194.100.1.0	255.255.255.0	I1
194.100.2.0	255.255.254.0	I1
194.100.4.0	255.255.252.0	I1
194.100.8.0	255.255.248.0	I1
194.100.48.0	255.255.240.0	I1
194.100.64.0	255.255.240.0	I1

3. feladat - megoldás

DestNet	NetMask	IF
194.100.0.0	255.255.255.0	I1
194.100.1.0	255.255.255.0	I1
194.100.2.0	255.255.254.0	I1
194.100.4.0	255.255.252.0	I1
194.100.8.0	255.255.248.0	I1
194.100.48.0	255.255.240.0	I1
194.100.64.0	255.255.240.0	I1

- 194.100.0.0/24 -> 194.100.0.0-tól 194.100.0.255-ig (256 cím)
- 194.100.1.0/24 -> 194.100.1.0-tól 194.100.1.255-ig (256 cím)
- 194.100.2.0/23 -> 194.100.2.0-tól 194.100.3.255-ig (512 cím)
- 194.100.4.0/22 -> 194.100.4.0-tól 194.100.7.255-ig (1024 cím)
- 194.100.8.0/21 -> 194.100.8.0-tól 194.100.15.255-ig (2048 cím)
- Folytonos címtartományok, megegyezik az első 20 bit, 4096 cím
  - CIDR-rel 194.100.0.0/20

- 194.100.48.0/20 -> 194.100.48.0-tól 194.100.63.255-ig (4096 cím)
- 194.100.64.0/20 -> 194.100.64.0-tól 194.100.79.255-ig (4096 cím)
- Folytonos címtartomány, de 48 -> 00110000, 64 -> 01000000, nem egyezik meg az első 19 bit, nem lehet aggregálni egy /19-es címbe

3. feladat - megoldás

DestNet	NetMask	IF
194.100.0.0	255.255.255.0	I1
194.100.1.0	255.255.255.0	I1
194.100.2.0	255.255.254.0	I1
194.100.4.0	255.255.252.0	I1
194.100.8.0	255.255.248.0	I1
194.100.48.0	255.255.240.0	I1
194.100.64.0	255.255.240.0	I1

- Tömörített CIDR routing tábla

DestNet	IF
194.100.0.0/20	I1
194.100.48.0/20	I1
194.100.64.0/20	I1

4. Feladat

Az alábbi routing táblát hogyan lehet tömöríteni CIDR-t használva?

DestNet	NetMask	IF
200.0.0.0	255.255.192.0	A
200.0.64.0	255.255.192.0	A
200.0.128.0	255.255.128.0	A
200.1.0.0	255.255.0.0	A
193.0.2.0	255.255.255.0	B
193.0.3.0	255.255.255.0	B
193.0.4.0	255.255.255.0	B
193.0.5.0	255.255.255.0	B

4. feladat - megoldás

DestNet	NetMask	IF
200.0.0.0	255.255.192.0	A
200.0.64.0	255.255.192.0	A
200.0.128.0	255.255.128.0	A
200.1.0.0	255.255.0.0	A
193.0.2.0	255.255.255.0	B
193.0.3.0	255.255.255.0	B
193.0.4.0	255.255.255.0	B
193.0.5.0	255.255.255.0	B

- 200.0.0.0/18 -> 200.0.0.0-tól 200.0.63.255-ig (16384 cím)
- 200.0.64.0/18 -> 200.0.64.0-tól 200.0.127.255-ig (16384 cím)
- 200.0.128.0/17 -> 200.0.128.0-tól 200.0.255.255-ig (32768 cím)
- 200.1.0.0/16 -> 200.1.0.0-tól 200.1.255.255-ig (65536 cím)
- Folytonos címtartomány, megegyező első 15 bit, 131072 cím
  - CIDR-rel 200.0.0.0/15

- 193.0.2.0/24 -> 193.0.2.0-tól 193.0.2.255-ig (256 cím)
- 193.0.3.0/24 -> 193.0.3.0-tól 193.0.3.255-ig (256 cím)
- 193.0.4.0/24 -> 193.0.4.0-tól 193.0.4.255-ig (256 cím)
- 193.0.5.0/24 -> 193.0.5.0-tól 193.0.5.255-ig (256 cím)
- 193.0.2.0 és 193.0.3.0 folytonos címtartomány, megegyező első 23 bit -> 193.0.2.0/23
- 193.0.4.0 és 193.0.5.0 folytonos címtartomány, megegyező első 23 bit -> 193.0.4.0/23

4. feladat - megoldás

- Tömörített CIDR routing tábla

DestNet	IF
200.0.0.0/15	A
193.0.2.0/23	B
193.0.4.0/23	B

DestNet	NetMask	IF
200.0.0.0	255.255.192.0	A
200.0.64.0	255.255.192.0	A
200.0.128.0	255.255.128.0	A
200.1.0.0	255.255.0.0	A
193.0.2.0	255.255.255.0	B
193.0.3.0	255.255.255.0	B
193.0.4.0	255.255.255.0	B
193.0.5.0	255.255.255.0	B

5. Feladat (IMSc)

- Egy ISP rendelkezik a 194.48.0.0/16 címtartomány felett
- 3 szervezetnek (A, B és C) 2048, 8192 illetve 4096 címre van szüksége
- Az A és C szervezetek az 1. interfészen keresztül érhetőek el, a B szervezet a 2. interfészen

- Optimizáljuk CIDR-t használva a címkiosztást a szervezeteknek. Hogyan fog kinézni az útválasztó tábla?

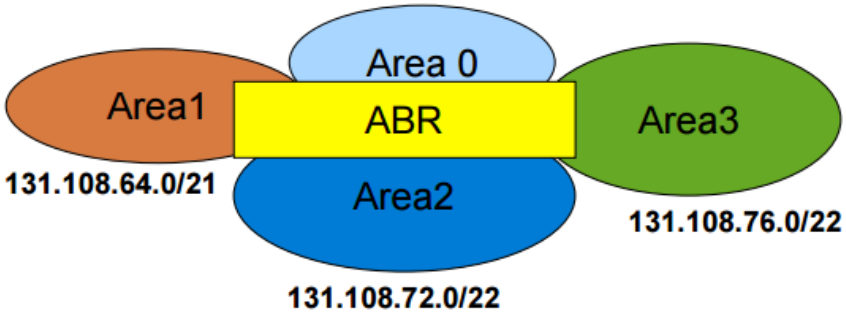
6. feladat

- Egy X cégnek legyen a címtartománya 131.108.64/20
- A tartományt a cég 3 egysége között kell felosztani, a következőképpen:
  - Az A egységnek max. 2000 címre van szüksége
  - A B és C egységeknek max 1000-1000 címre van szüksége

- Javasoljunk egy megfelelő hálózati architektúrát, feltételezve hogy OSPF-et használunk IGP protokollként
- Hogyan osztjuk fel a címtartományt?

6. feladat - megoldás

- Címtartomány mérete: 4096 cím -> OK
- 3 OSPF tartomány (+ area 0), egységenként egy
- Címkiosztás:
  - A: 2048 cím -> /21 -> 131.108.64.0/21 (131.108.64.0-tól 131.108.71.255-ig), area1
  - B: 1024 cím -> /22 -> 131.108.72.0/22, (131.108.72.0-tól 131.108.75.255-ig), area2
  - C: 1024 cím -> /22 -> 131.108.76.0/22 , (131.108.76.0-tól 131.108.79.255-ig), area3



5. feladat - megoldás

- A és C egy interfészen, B egy másikon -> legalább két bejegyzés a routing táblában
- A -> 2048 cím -> /21
- C -> 4096 cím -> /20
- Nem lehet aggregálni, mert nem töltenek ki egy teljes /19-es tartományt

- A -> 194.48.0.0, subnet mask 255.255.248.0 , 194.48.0.0-tól 194.48.7.255-ig, IF 1
- C -> 194.48.16.0, subnet mask 255.255. 240.0 , 194.48.16.0-tól 194.48.31.255-ig, IF 1
- 2048 címes blokk az A és C között fenntartva egy későbbi lehetséges aggregálásra

- B -> 8192 cím -> /19
- B -> 194.48.32.0, subnet mask 255.255.224.0 , 194.48.32.0-tól 194.48.63.255-ig, IF 2

Szállítási réteg (L4)

Gyakorlat

Budapest University of Technology and Economics

Department of
Telecommunications and Media Informatics

A gyakorlat célja

- A TCP-t nagyon sok környezetben használják
- A főbb mechanizmusok ismerete fontos
  - A programozónak
  - A hálózati szakembernek
    - Wired
    - Wireless
  - Lassan az ipari környezetbe is beszivárog

Portok

- Kapcsolat fogalma
  - 5-tuple
- Kapcsolat kiépítése
  - UDP - nincs
  - TCP
    - 3 way handshake
    - Miért van rá szükség?

Portok egy gépen

- Netstat parancs

```
TCP    127.0.0.1:1906         localhost:1907     ESTABLISHED
TCP    192.168.1.147:53699    13.77.87.52:https      ESTABLISHED
TCP    192.168.1.147:53703    91.190.216.57:12350    ESTABLISHED
TCP    192.168.1.147:53737    64.4.23.152:40008      ESTABLISHED
TCP    192.168.1.147:53759    108.177.96.188:5228    ESTABLISHED
TCP    192.168.1.147:53772    40.77.226.192:https    ESTABLISHED
TCP    192.168.1.147:54512    a104-96-129-73:https   CLOSE_WAIT
TCP    192.168.1.147:54513    a104-96-129-73:https   CLOSE_WAIT
TCP    192.168.1.147:54514    a104-96-129-73:https   CLOSE_WAIT
```

UDP

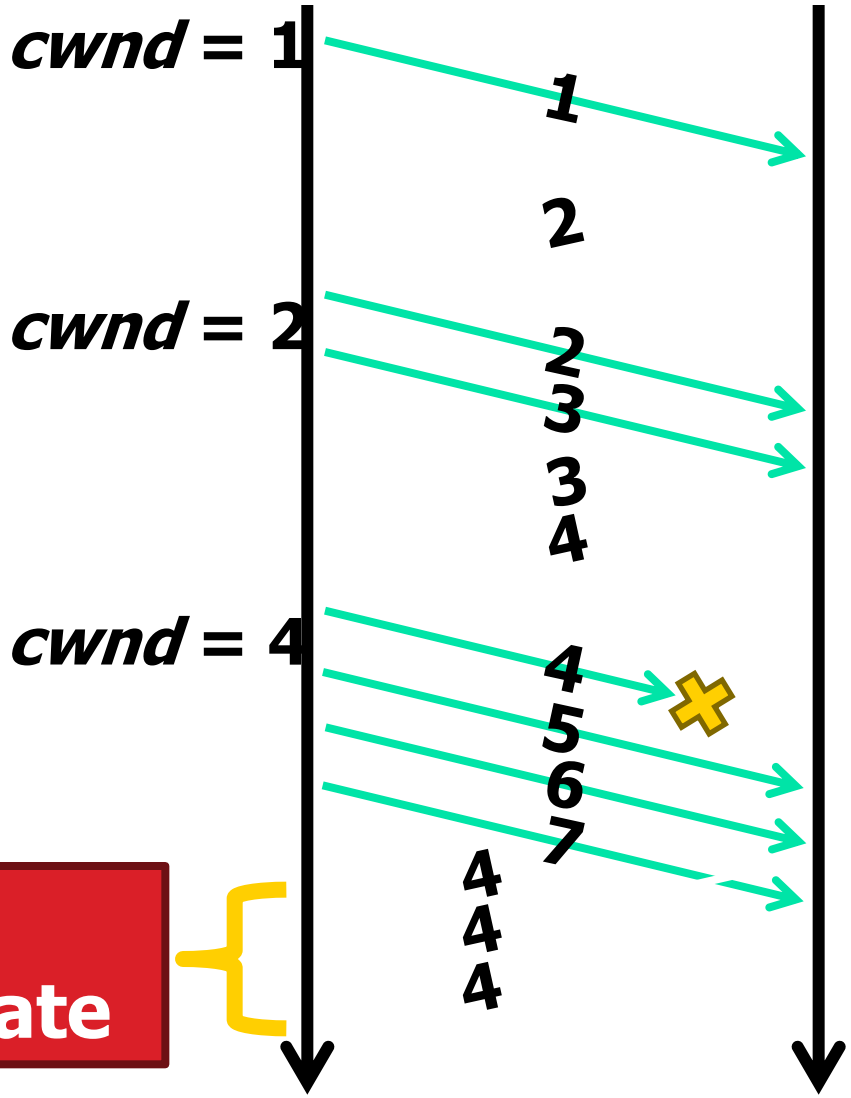
- Adatküldés: szegmensek
- Hibakezelés
  - ICMP: port nem elérhető
  - Loss: nincs visszajelzés
- Sávszélesség, késleltetés

The Evolution of TCP

- TCP Tahoe
  - Kezdeti verzió
- A TCP 1974-ben volt kitalálva!
  - Manapság rengeteg változata van a TCP-nek
- Kezdeti, elterjedt: TCP Reno
  - Tahoe, plus…
  - Fast retransmit
  - Fast recovery

TCP Reno: Fast Retransmit

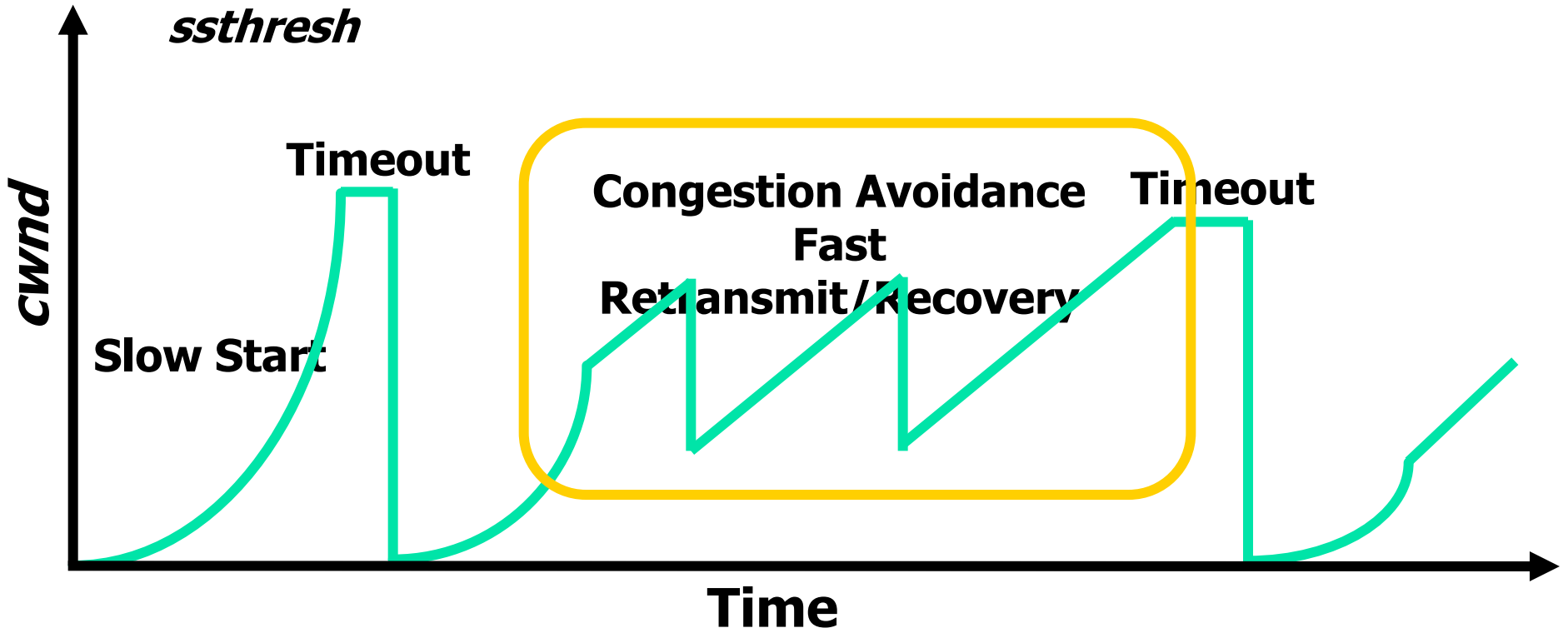
- **Problem: in Tahoe, if segment is lost, there is a long wait until the RTO**
- **Reno: retransmit after 3 duplicate ACKs**



TCP Reno: Fast Recovery

- After a fast-retransmit set *cwnd* to *ssthresh/2*
  - i.e. don't reset *cwnd* to 1
  - Avoid unnecessary return to slow start
  - Prevents expensive timeouts
- But when RTO expires still do *cwnd* = 1
  - Return to slow start, same as Tahoe
  - Indicates packets aren't being delivered at all
  - i.e. congestion must be really bad

Fast Retransmit and Fast Recovery



- At steady state, *cwnd* oscillates around the optimal window size
- TCP always forces packet drops

Many TCP Variants…

- Tahoe: the original
  - Slow start with AIMD
  - Dynamic RTO based on RTT estimate
- Reno: fast retransmit and fast recovery
- NewReno: improved fast retransmit
  - Each duplicate ACK triggers a retransmission
  - Problem: >3 out-of-order packets causes pathological retransmissions
- Vegas: delay-based congestion avoidance
- And many, many, many more…

TCP in the Real World

- What are the most popular variants today?
  - Key problem: TCP performs poorly on high bandwidth-delay product networks (like the modern Internet)
  - Compound TCP (Windows)
    - Based on Reno
    - Uses two congestion windows: delay based and loss based
    - Thus, it uses a *compound* congestion controller
  - TCP CUBIC (Linux)
    - Enhancement of BIC (Binary Increase Congestion Control)
    - Window size controlled by cubic function
    - Parameterized by the time T since the last dropped packet

TCP ablak

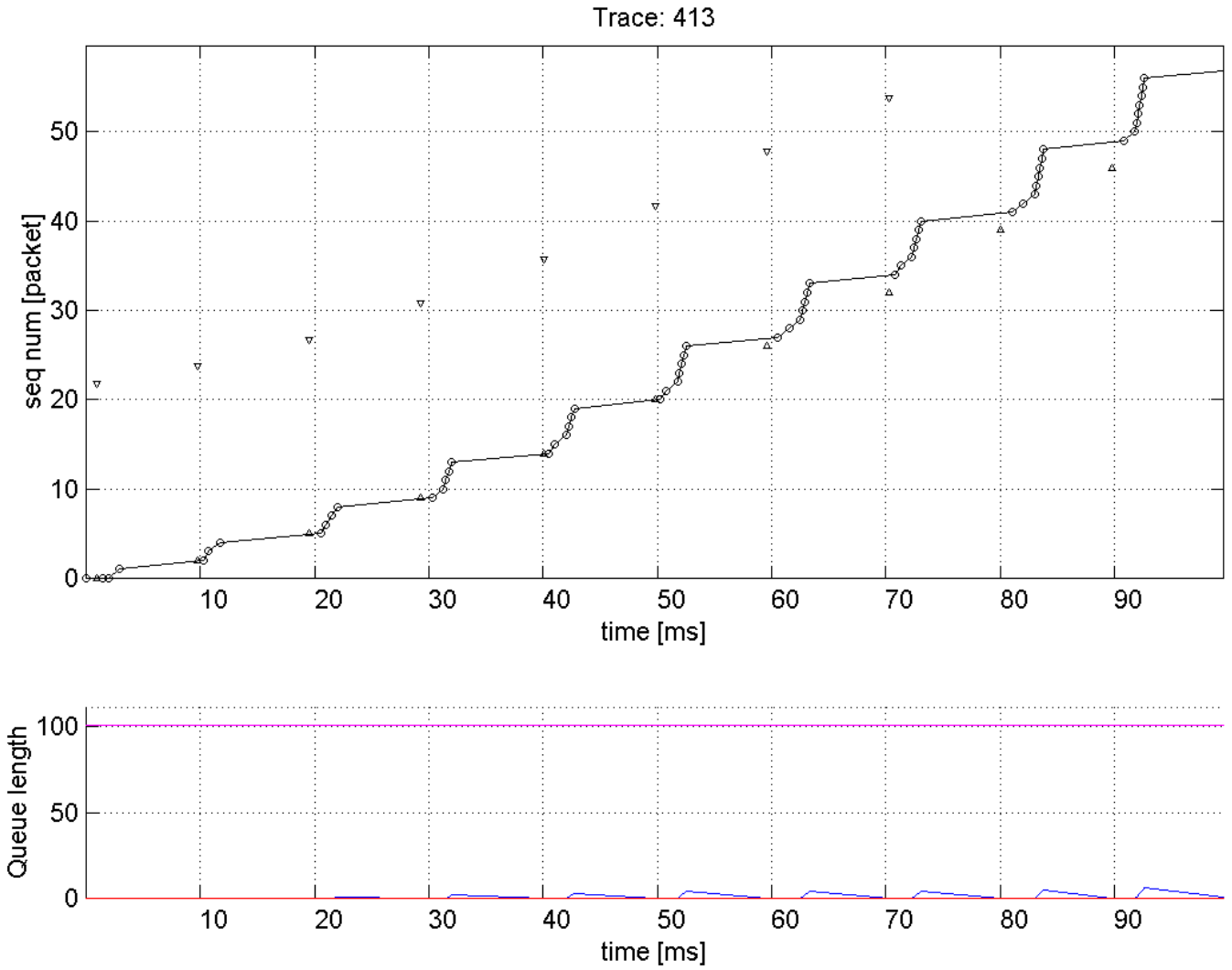
- Bandwidth delay product
- 8k – korlátos sávszélesség
  - Telítődés
- 64K
  - jobb
  - Window scale option

TCP opciók - példa

No.	Time	Source	Destination	Protocol	Length	Info
4	0.168986	192.168.0.11	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
5	0.221892	fe80::d0f9:8c1:d62f:eb63	ff02::1:3	LLMNR	86	Standard query 0x7e01 A isatap
6	0.000117	192.168.0.11	224.0.0.252	LLMNR	66	Standard query 0x7e01 A isatap

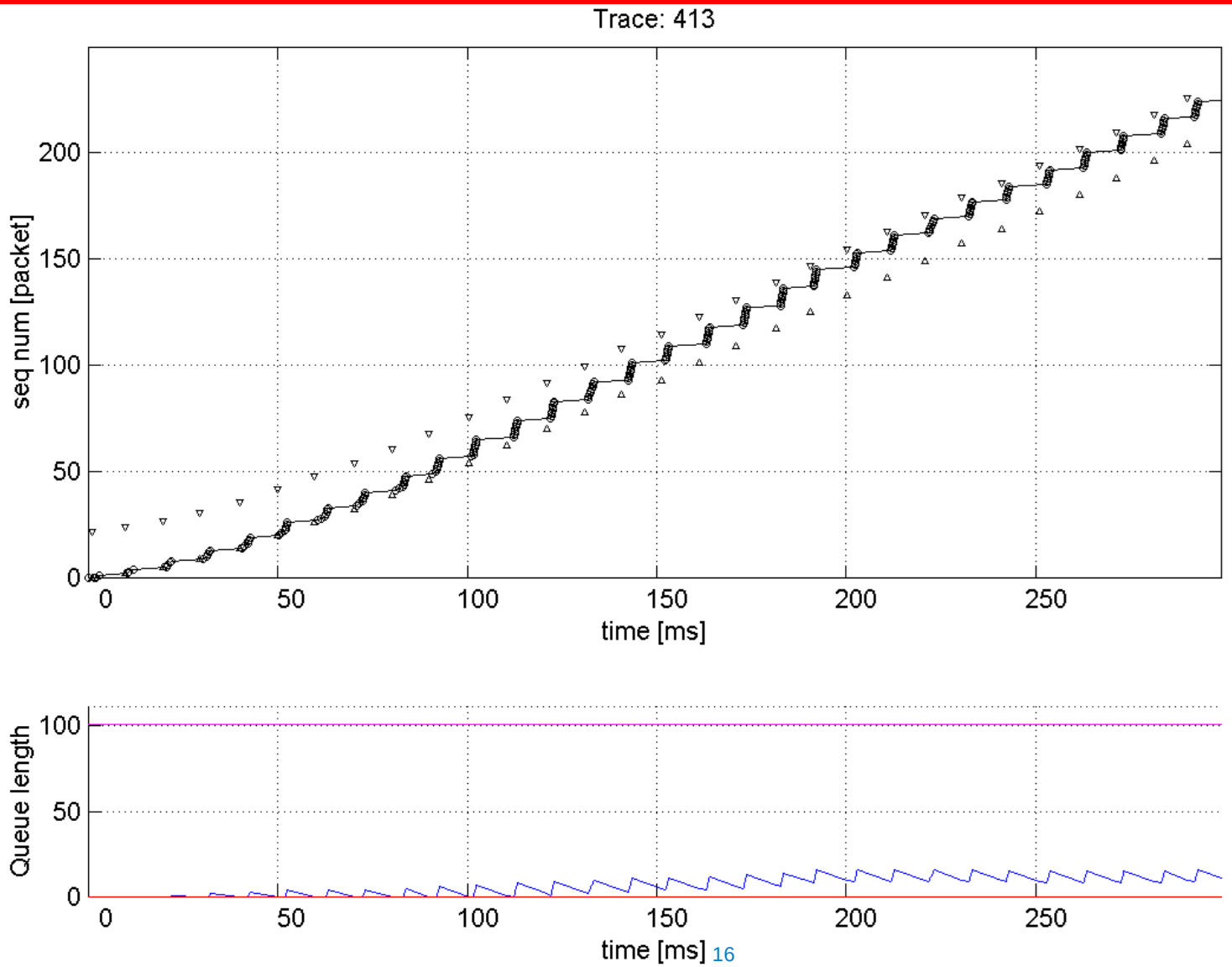
```
▷ Frame 12: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: Wistron_2d:ab:ba (00:1f:16:2d:ab:ba), Dst: 3Com_03:04:05 (00:01:02:03:04:05)
▷ Internet Protocol Version 4, Src: 192.168.0.11, Dst: 192.168.0.168
◢ Transmission Control Protocol, Src Port: 29385, Dst Port: 22, Seq: 0, Len: 0
    Source Port: 29385
    Destination Port: 22
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    [Next sequence number: 0    (relative sequence number)]
    Acknowledgment number: 0
    1000 .... = Header Length: 32 bytes (8)
  ▷ Flags: 0x002 (SYN)
    Window size value: 8192
    [Calculated window size: 8192]
    Checksum: 0x822a [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ◢ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (NOP), SACK permitted
      ▷ TCP Option - Maximum segment size: 1460 bytes
      ▷ TCP Option - No-Operation (NOP)
      ▷ TCP Option - Window scale: 2 (multiply by 4)
      ▷ TCP Option - No-Operation (NOP)
      ▷ TCP Option - No-Operation (NOP)
      ▷ TCP Option - SACK permitted
  ◢ [Timestamps]
      [Time since first frame in this TCP stream: 0.000000000 seconds]
      [Time since previous frame in this TCP stream: 0.000000000 seconds]
```

No.	Algo.	Sender	Traffic	# conn.	Receiver	Kbytes/sec	# ovfl.
413	None	Linux 2.1	one.1024	1	Linux 2.0	1097.7 (gw)	0



Megjegyzések

- A fogadó csak egy ACK-t küld egy csomagcsoportra
- Minden ACK érkezése a küldőnél egy csomagbörszt küldését eredményezi
- A küldő a *cwnd*-t minden egyes ACK érkezésekor eggyel növeli
- A sorok hossza növekszik a börsztök érkezésekor, a börsztök méretei pedig növekednek az idővel
- ACK érkezések 10 ms-ként

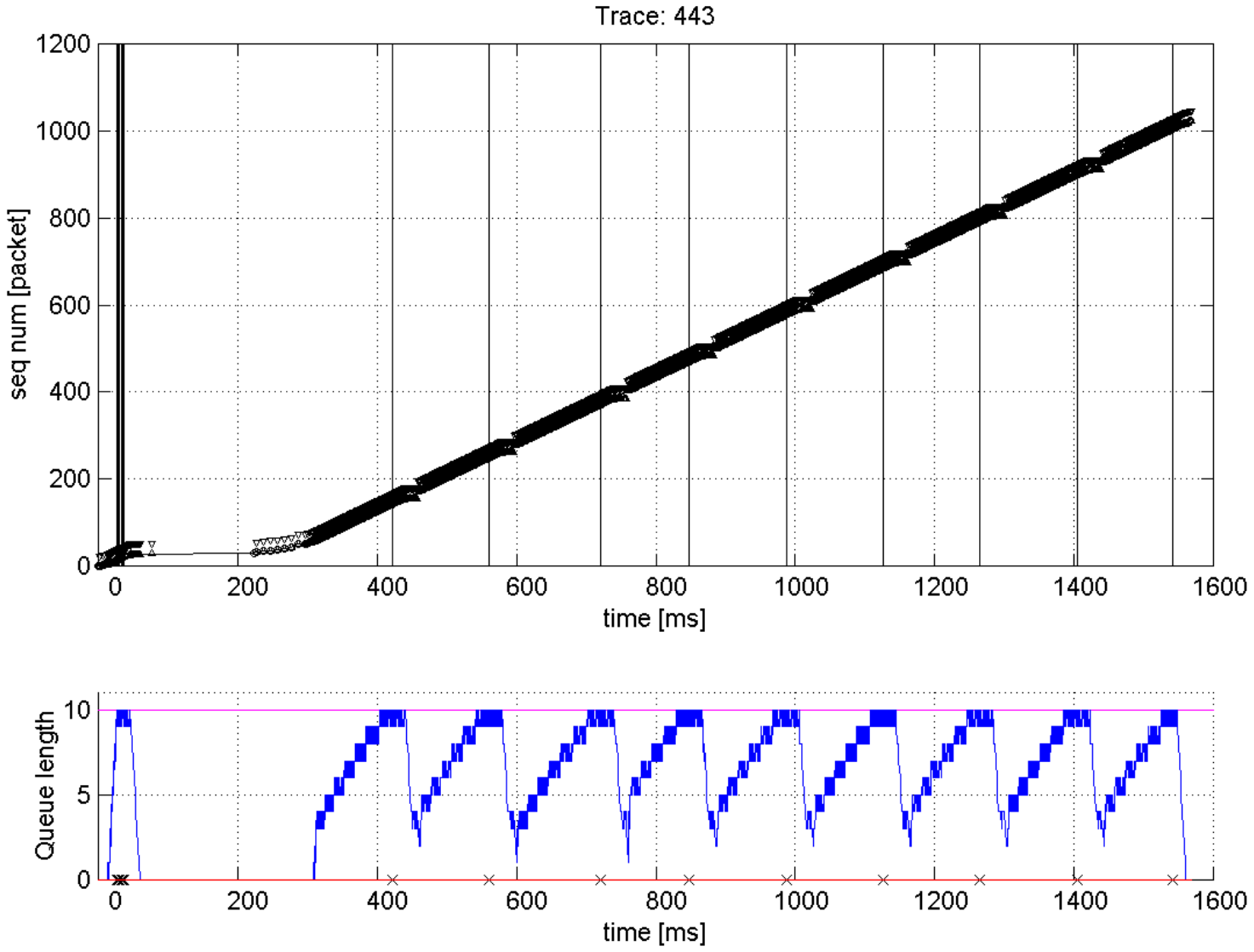

Trace: 413
seq num [packet]
time [ms]
Queue length
time [ms]
0
50
100
150
200
250

Megjegyzések

- Kb 200ms után, a küldő eléri a fogadó meghirdetett ablakméretének felső határát
- A sorhosszúság kb. 15 körül stabilizálódik
- A sorok hossza a börsztök érkezésekor növekszik – ezeket a fogadó egy-egy csomagcsoportot nyugtázó ACK-ra küldött
- A sorok hossza fokozatosan csökken függően a 10 Mbps-os Ethernet kapacitásától

Börsztös forgalom

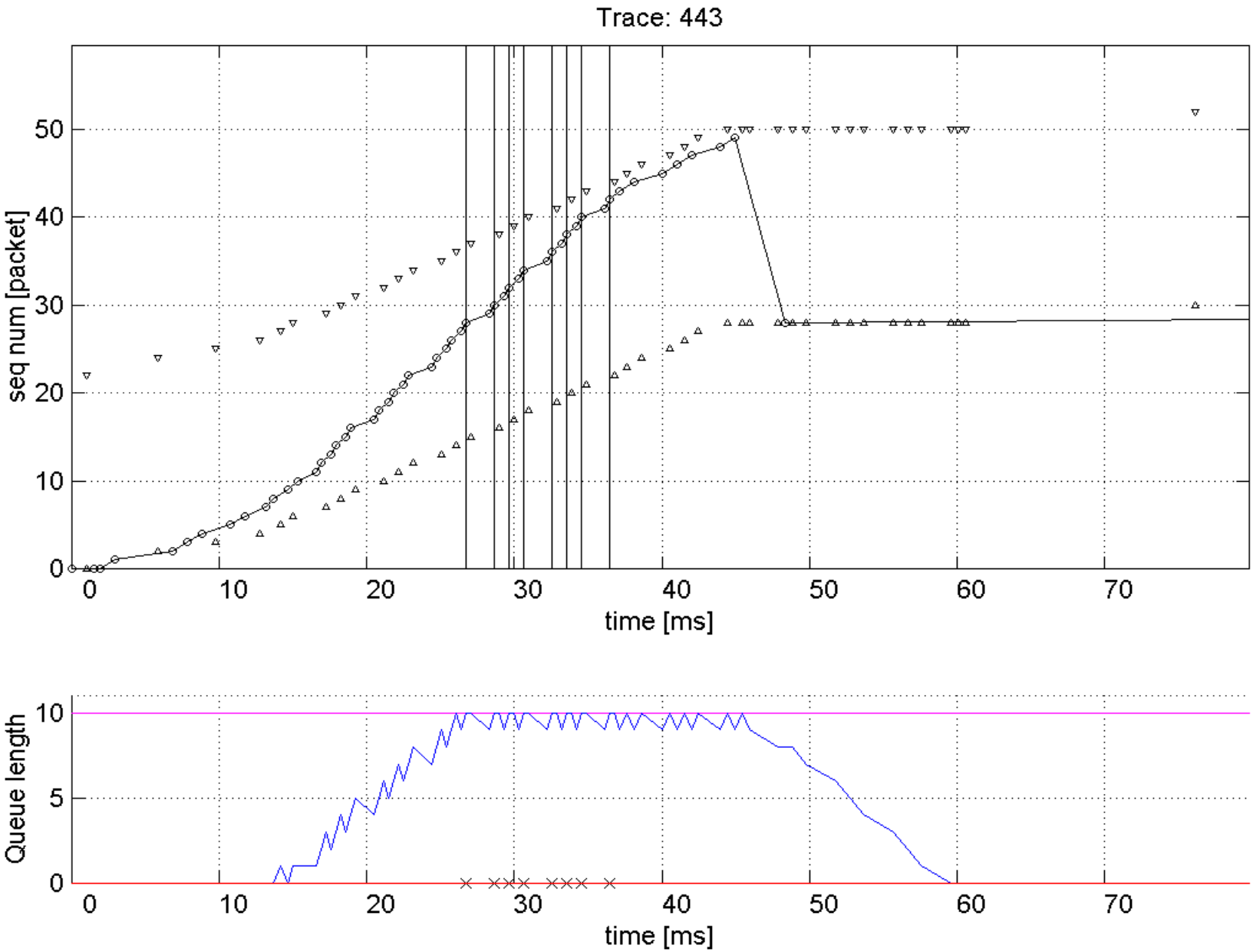
- Ablak – gyors felfutás
  - Hatása: többszörös loss
  - Timeout
- Sok timeout – lassú kapcsolat
  - Főleg a kapcsolat elején nagy gond
  - Inkább fast retransmit kellene


Trace: 443
seq num [packet]
time [ms]
Queue length
time [ms]

Megjegyzések

- A slow start börsztös csomagvesztéssel és az adás szüneteltetésével ér véget
- Periódikus vesztések láthatók, melyeket a küldő gyorsan javít – a vesztések nem okoznak jelentős teljesítménycsökkenést
- A sorhosszban periódikus minta van 300 ms után: lineáris növekedés, egy vesztés, egy esés. Ez mutatja a congestion avoidance mechanizmus működését a küldő oldalon: a vesztés észrevételekor csökkenti a congestion window értékét, majd ismét lineárisan (additívan) növeli

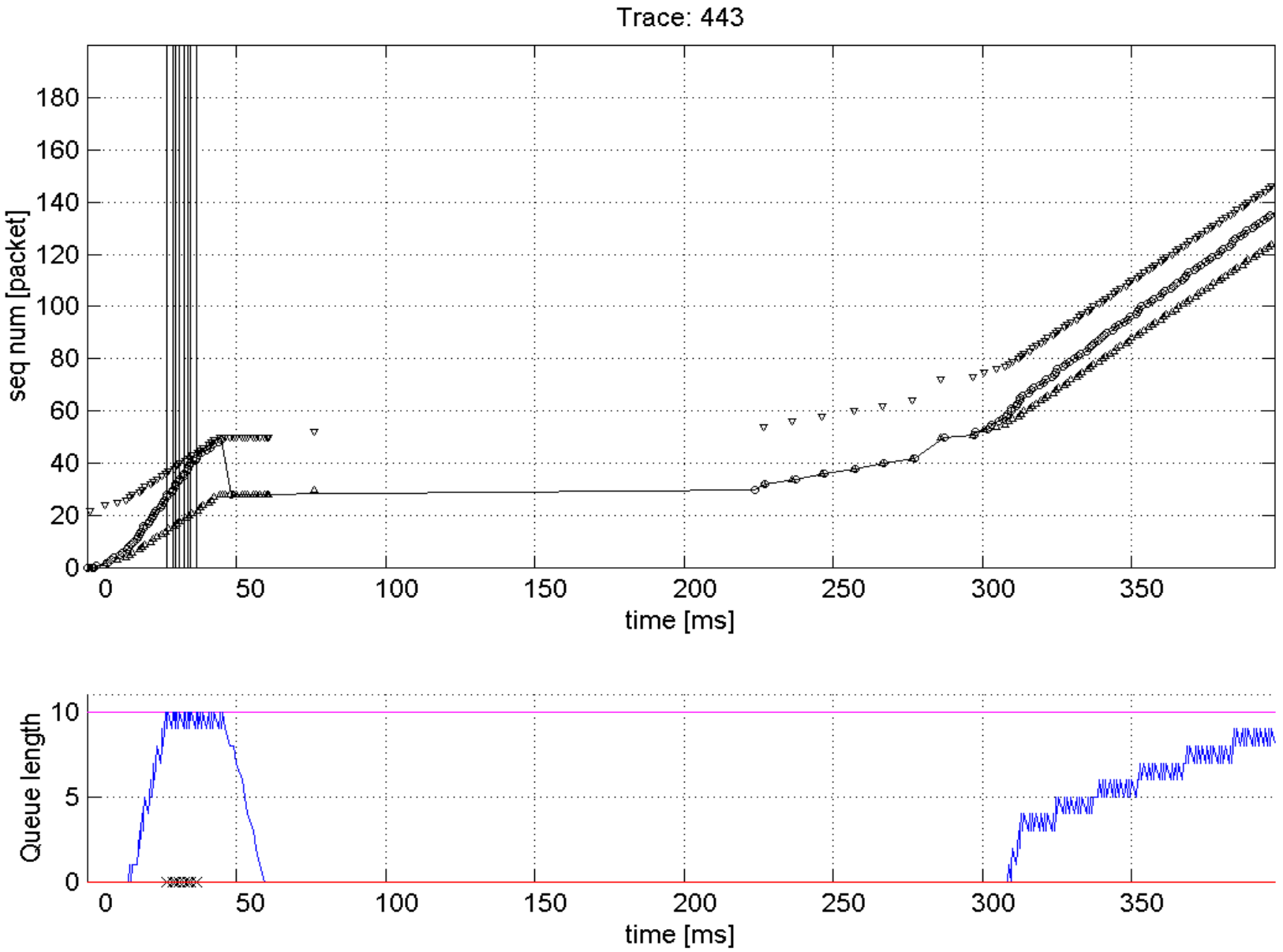
Az első 80 ms



Magyarázatok

26-36ms	A fogadó felé a sorok túlcsordulnak, a csomagok eldobásra kerülnek
37-45ms	A fogadó felé menő csomagok továbbításra kerülnek, a fogadó fogadja ezeket a csomagokat, de nem nyugtázza őket, mert néhány korábbi csomag is hiányzik
44-47ms	A fogadó duplikált ACK-t küld
48ms	A küldő újraküldi az első nemnyugtázott csomagot (fast retransmit).
49-61ms	A fogadó továbbra is duplikált ACK-kat küld
76ms	A fogadó nyugtázza az újraküldött csomagokat

200 ms környéke

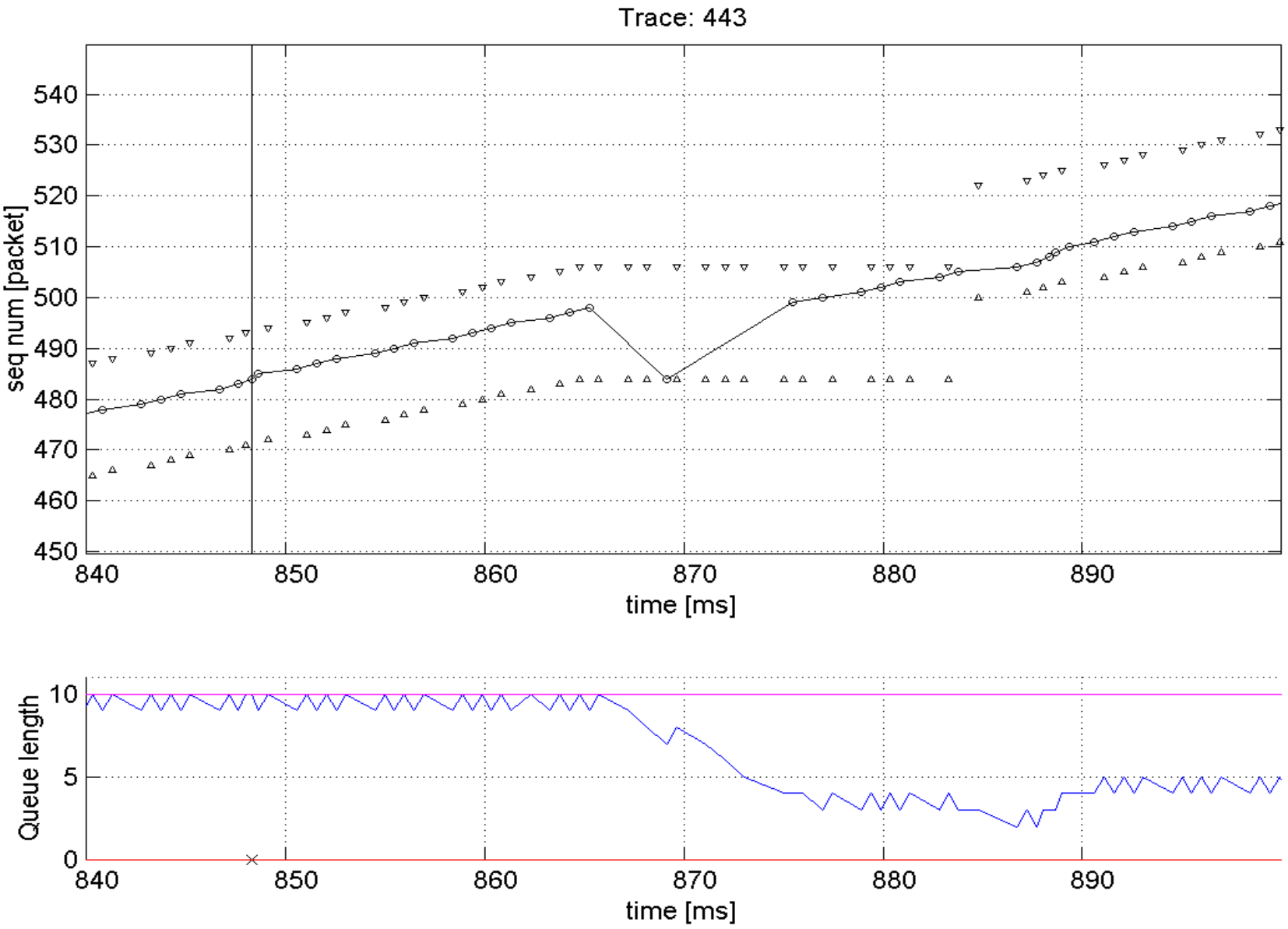


Magyarázatok

225ms	A küldő időzítője lejár és küld egy következő nemnyugtázott csomagot. Az időztés kb 200 ms
225-275ms	A küldő továbbít egy csomagot rögtön, amint a fogadó nyugtázta az előzőt. A küldő nem növeli a congestion window méretét, amíg újraküldést végez
275ms	A küldő minden elküldött adatára kap nyugtát (Az ACK számok ugrása). Slow start kezdődik.
330ms-	A gyors (exponenciális) növekedése a nemnyugtázott csomagoknak megáll és congestion avoidance szakasz következik – ez látszik a sorhossz lineáris növekedéséből

- Börsztös csomagvesztés történt
- Az első vesztést a küldő vette észre a duplikált ACK-ból, és gyors újraküldés következett
- A következő vesztéseket szintén a küldő vette észre az időzítők lejáratakor – emiatt slow start indult
- Ez a mechanizmus börsztös vesztéseknél gyakori

870 ms környéke



Magyarázatok

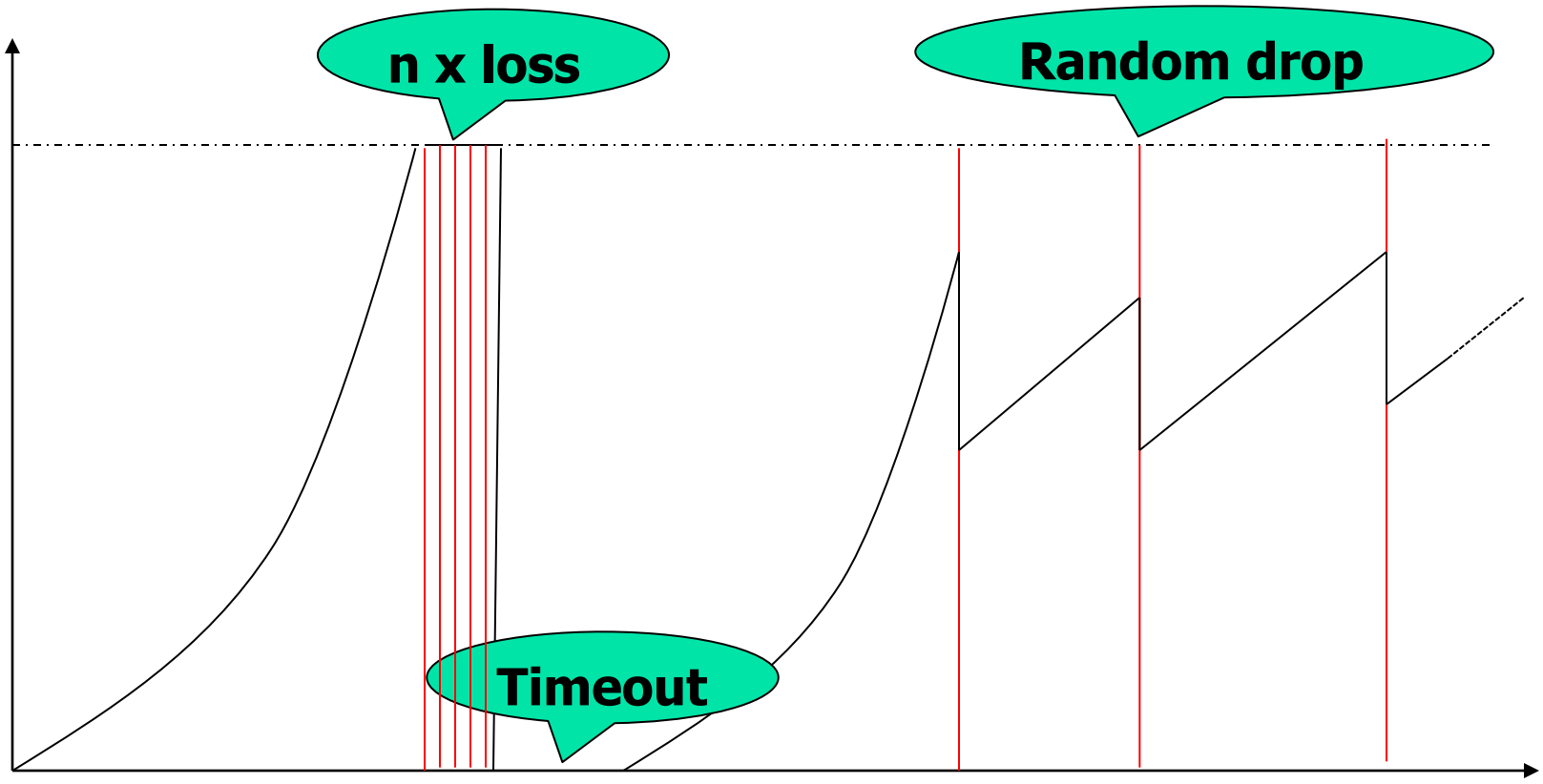
848ms	Egy csomag veszett el a közbenső sorok megtelése miatt
848-864ms	A fogadó a korábbi csomagokat nyugtázza
864-868ms	A fogadó a további csomagokat nem tudja nyugtázni, mert egy hiányzik. Emiatt a fogadó duplikált ACK-t küld
869ms	A küldő 3 duplikált ACK-t kap, majd gyors újraküldést hajt végre
875-884ms	A küldő folytatja a csomagok küldését, mivel nem érte el a fogadó advertised window méretét még.
885ms	Egy ugrás látható az ACK numberek között – a fogadó megkapta a hiányzó csomagot, és az azutániakkal együtt nyugtázta azt.
885ms-	A küldő folytatja a csomagok továbbítását (fast recovery történt: slow start nem következik most). A gyors újraküldés nem okozott nagy teljesítmenycsökkenést

Csomagvesztés – jó vagy rossz?

- Példa: DSL 10Mbit/s vonal, 2 lehetőség
  - a) x% loss vs
  - b) FEC – hibajavító kód, fix 20% adat
- TCP transport, letöltések: melyik jobb?
- a) 10Mbps – x% loss (pl. BER=10^{-6})
  - ⇒1 csomag 1500 byte, 12000bit, minden 83.3 csomag elvész
- b) 8Mbps
  - a hibajavító kód 20% veszteség mindig

RED – Random Early Drop

- Buffer menedzsment a routerekben
  - Egy dobás jobb mint egy timeout



TCP - Wireless

- Probléma:
  - Wireless – eleve van loss
    - de az nem túlcsordulás, nem szűk keresztmetszet!
    - TCP rosszul értelmezi, visszaveszi a cwnd-t
  - Megoldások - többféle
    - WTCP – proxy
    - SACK

Köszönöm a figyelmet

- Vége -

Budapest University of Technology and Economics

Department of
Telecommunications and Media Informatics

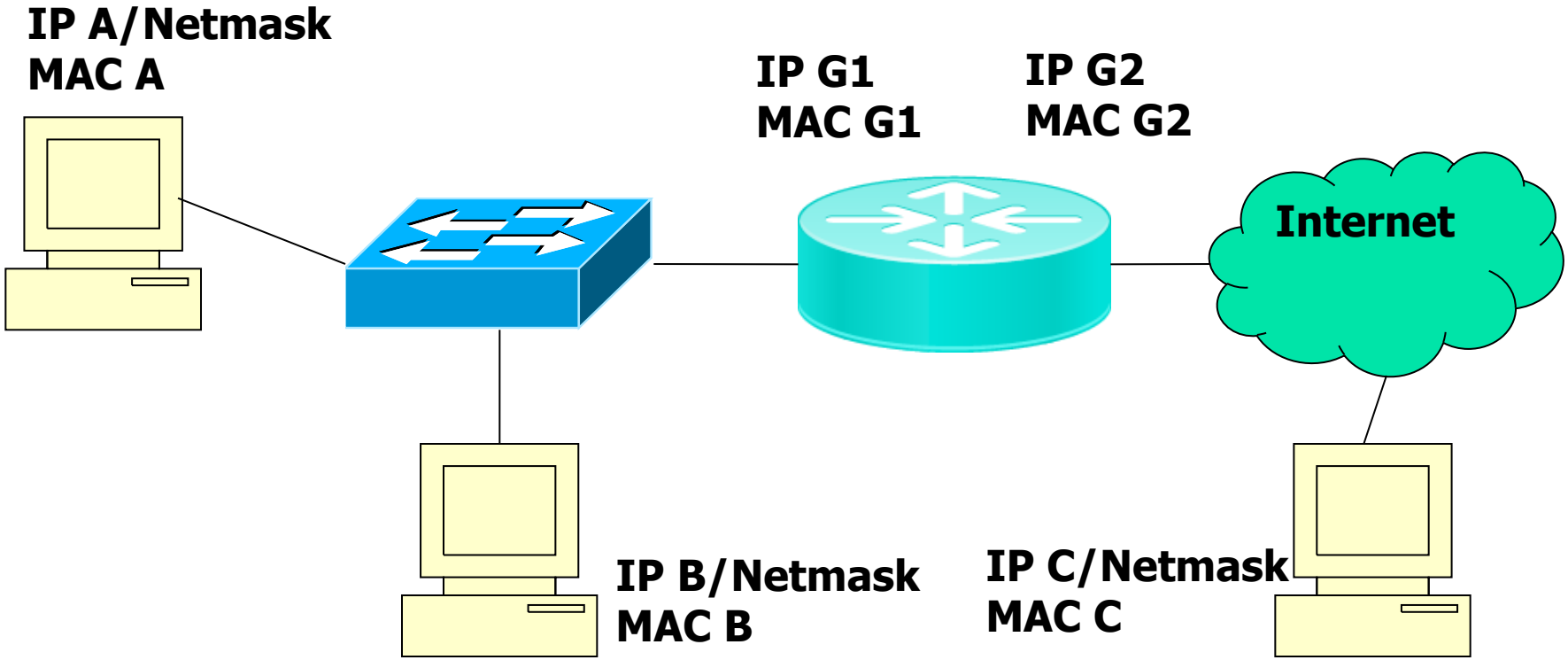
Ethernet/IP címzés - gyakorlat

Moldován István

moldovan@tmit.bme.hu

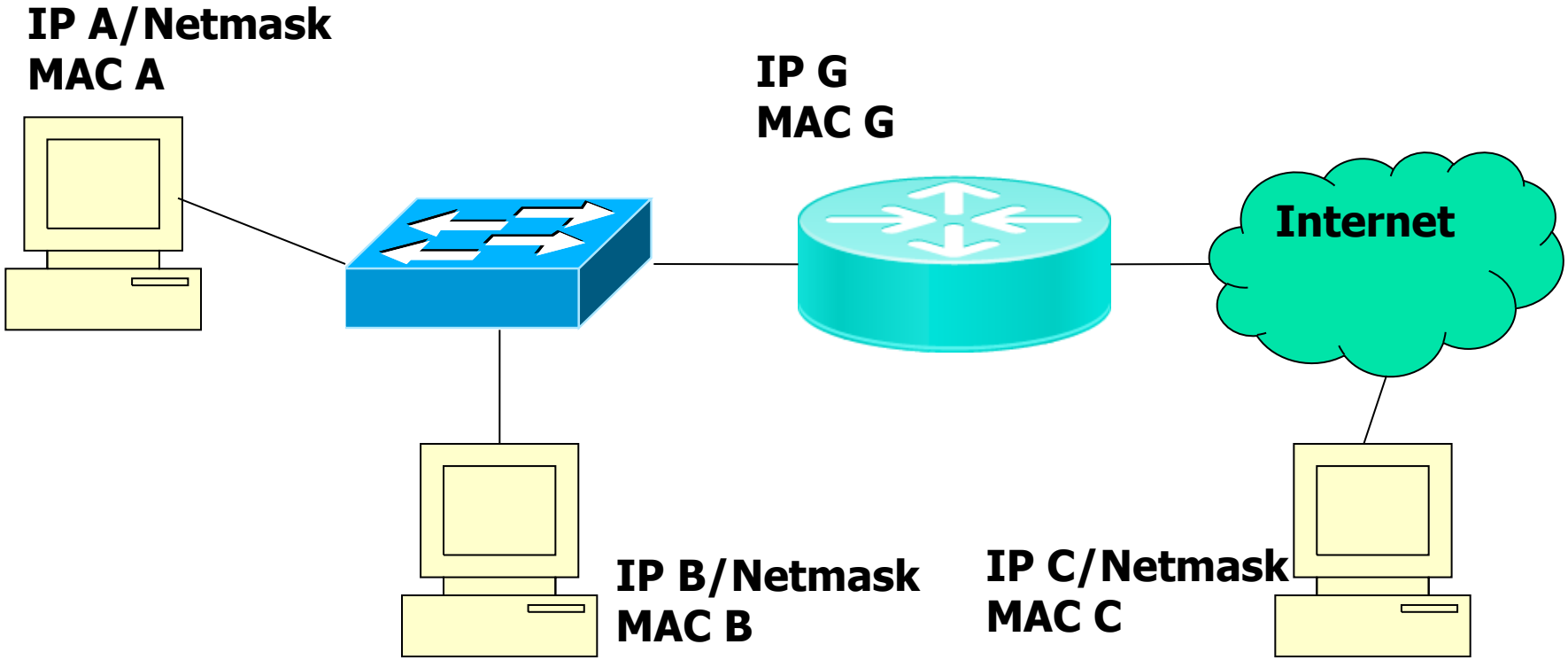
BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
TÁVKÖZLÉSI ÉS MÉDIAINFORMATIKAI TANSZÉK

IP+Ethernet működés - Címek



IP A, B, G1: ugyanaz a subnet
IP G2, C: más subnet

IP+Ethernet működés



1. A->B – ugyanaz a subnet, cél MAC: B
2. A->C – nem lokális háló, GW-en keresztül – cél MAC: G

MAC cím feloldása: ARP

Köszönöm a figyelmet!

ZH feladatok

Budapest University of Technology and Economics

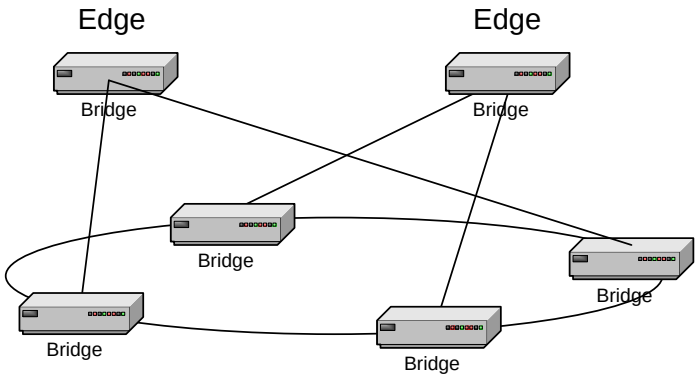
Department of
Telecommunications and Media Informatics

ZH

- 10 kérdés, fele gyakorlati
- Elméleti kérdések:
  - Röviden kifejtendőek
    - Pl1: Mi a VLAN és hogyan működik?
    - PL2: Mi az FDMA? Működhet-e párhuzamosan TDD-vel? A választ indokoljátok!
  - Gyakorlati
    - Levezetendő
    - Gyakorlatokon tárgyalt témákból

Gyakorlati kérdés példák - 1

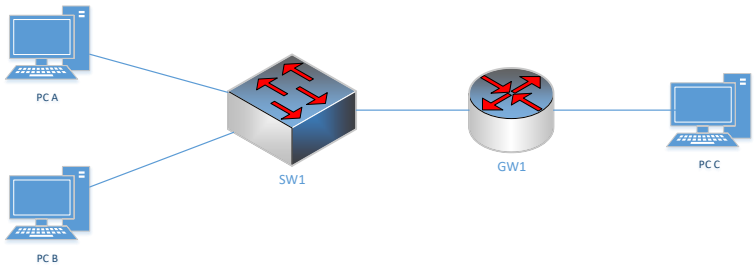
- STP/MSTP



- Adott topológia, rajzoljátok fel a fát
  - RSTP
  - MSTP esetén

Gyakorlati példák - 2

- Csomag továbbítás az alábbi hálózaton
  - Ethernet, ARP, IP



Gyakorlati példák - 3

- IP címzés
  - IP címek – Netmask, egy hálózaton vannak?
- Routing tábla
  - Longest prefix match
- CIDR
  - tömörítés

Kérdések?

Budapest University of Technology and Economics

Department of
Telecommunications and Media Informatics