Bezpieczeństwo...

Pojęcia:
- poufność (ang. secrecy) – dane są zaszyfrowane
- uwierzytelnianie (ang. authentication) - udowodnij kim jesteś
- integralność (ang. integrity) – nie można złośliwie zmodyfikować danych

Gdzie się zapewnia powyższe rzeczy w sieci komputerowej?
- nad warstwą 4 - połączenia tcp, SSL i TLS (obecnie obowiązuje TLS 1.3)
- w warstwie 3 - IPsec (patrz iproute2), zabezpiecza wszystko „nad warstwą 3”
- w warstwie 2 - wifi, WEP, WPA/WPA2
- „warstwa app” - ssh, sftp, sshfs (wyżej niż ssl)

Inne zabezpieczenia:
- bastion (wg Comera...),
  - szczególnie zabezpieczony komputer, do którego jest dostęp z zewnątrz
- zapory sieciowe, firewalls,
  - w linuxie polecenie **iptables** (tabela filter),
  - nie wpuszczamy pewnych pakietów, router nie przekazuje pewnych pakietów
- VPN = Virtual Private Network
  - sieć wirtualna zbudowana na publicznej sieci (ale zabezpieczona)

IPsec

Zabezpieczenie w warstwie 3...
Dwa nagłówki:
- AH = Auth Header – uwierz src, integr
  - zawiera: next header (proto), SPI, seq num, auth data
- ESP = Encapsualation Sec Payload – uwierz src, integr, poufność
  - zawiera: podobne pola jak AH

Security Agreement/Association, SA, logiczny kanał tworzony na początku...
- zawiera: typ prot (AH, ESP), źródłowy adr ip, 32bit ID połączenia (SPI); SA jest jednokierunkowe

Nr proto w nagłówku ip: 51 i 50

Figure 7.8-1: Position of the AH header in the IP datagram.

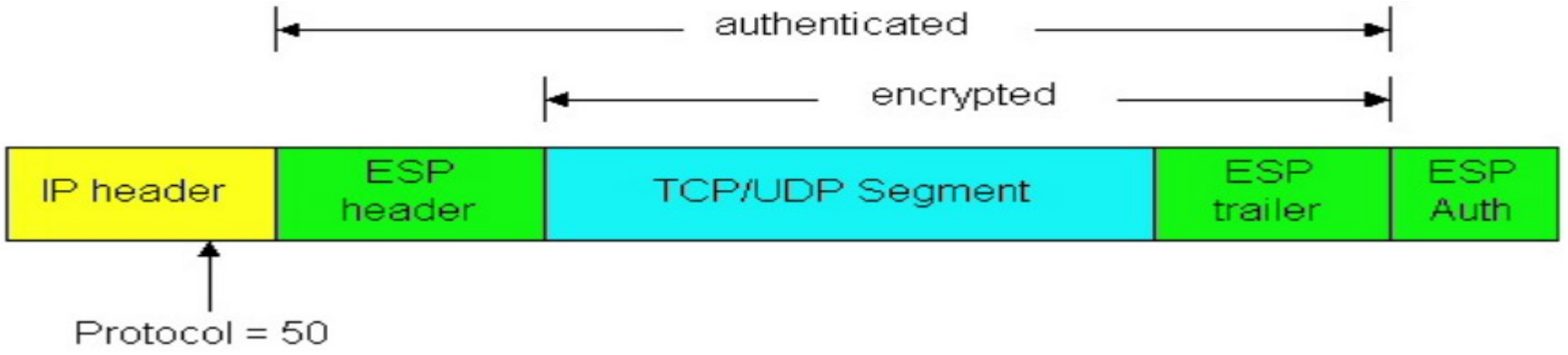


Figure 7.8-2: The ESP fields in the IP datagram.

Bastion (wg. Comera)

Bastion – maszyna między dwoma zaporami, pośrednik między internetem a siecią wewn
„zapora 1” przepuszcza pkg tylko do bastionu, „zapora 2” przepuszcza pkg tylko od bastionu;
komunikacja http intranet –> internet za pośrednictwem „http proxy” na bastionie?
Siec między zaporami: DMZ = strefa zdemilitaryzowana
Routery domowe: DMZ ma trochę inne znaczenie (?)

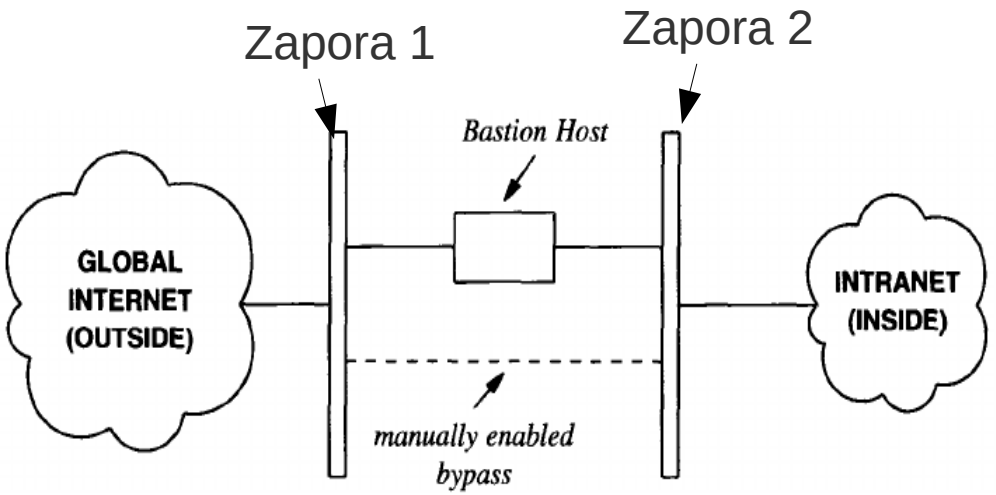


Figure 32.7 The conceptual organization of a bastion host embedded in a firewall. The bastion host provides secure access to outside services without requiring an organization to admit datagrams with arbitrary destinations.

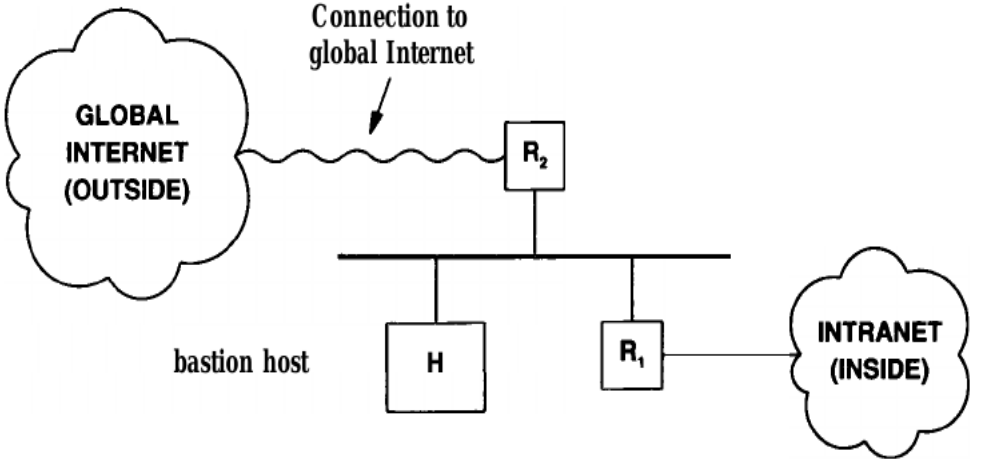
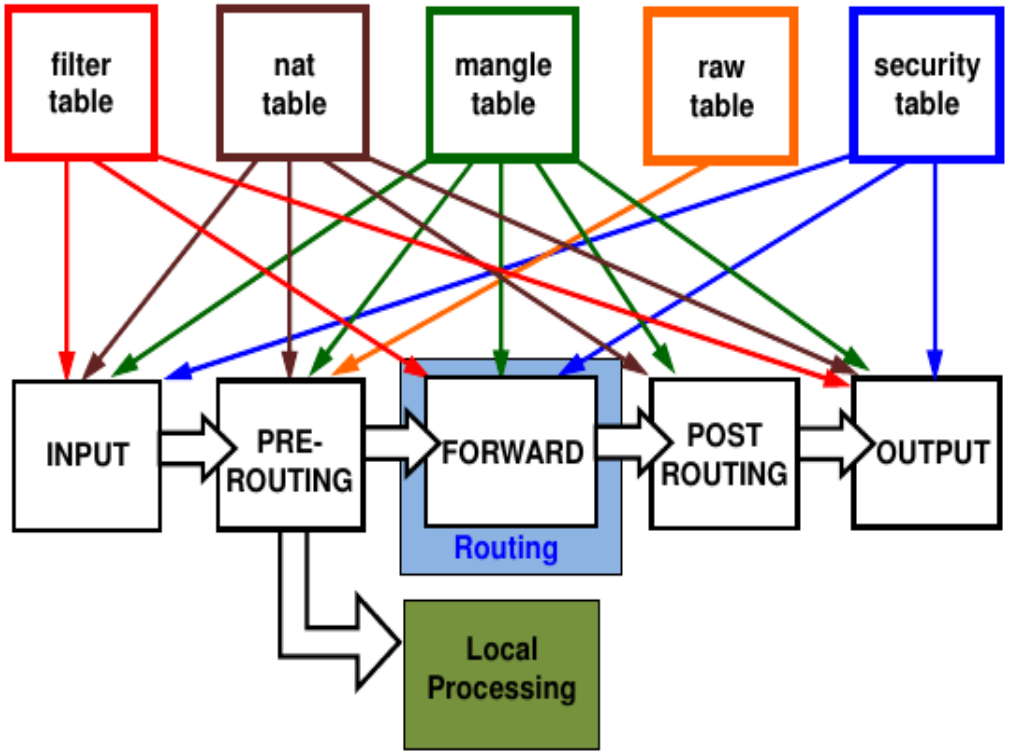


Figure 32.8 A firewall implemented with two routers and a bastion host. One of the routers has a connection to the rest of the Internet.

Zapory/ firewalls (iptables)




```
iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
  # wpuszczamy pakiety tcp, dport 8080
iptables -A INPUT -j DROP
  # odrzucamy wszystkie pakiety
```

Table	Function	Chain
filter (default)	Packet filtering / firewall	INPUT
		FORWARD
		OUTPUT
NAT	Network Address Translation	PREROUTING
		INPUT
		OUTPUT
		POSTROUTING
mangle	Packet modification	PREROUTING
		INPUT
		FORWARD
		OUTPUT
		POSTROUTING
security	Mandatory Access Control	INPUT
		FORWARD
		OUTPUT
raw	Bypass "**conntrack**" for corner cases	PREROUTING
		OUTPUT

Table-1: Iptables Tables & Chains

iproute2

Następca net-tools...
   to są komendy „ip” i „tc”
   oraz wsparcie w kernelu
Co można zrobić
przy pomocy iproute2 ?
   to co w net-tools,
   kształtowanie ruchu (qdisc),
   tunele, ipsec,
   zaawansowany routing,
   routing multicastowy,
Docs:
   Linux, Adv-Routing-HOWTO
Porównanie iproute2
  i net-tools …

NET-TOOLS COMMANDS	IPROUTE COMMANDS
arp -a	ip neigh
arp -v	ip -s neigh
arp -s 192.168.1.1 1:2:3:4:5:6	ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1
arp -i eth1 -d 192.168.1.1	ip neigh del 192.168.1.1 dev eth1
ifconfig -a	ip addr
ifconfig eth0 down	ip link set eth0 down
ifconfig eth0 up	ip link set eth0 up
ifconfig eth0 192.168.1.1	ip addr add 192.168.1.1/24 dev eth0
ifconfig eth0 netmask 255.255.255.0	ip addr add 192.168.1.1/24 dev eth0
ifconfig eth0 mtu 9000	ip link set eth0 mtu 9000
ifconfig eth0:0 192.168.1.2	ip addr add 192.168.1.2/24 dev eth0
netstat	ss
netstat -neopa	ss -neopa
netstat -g	ip maddr
route	ip route
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0 i	ip route add 192.168.1.0/24 dev eth0

iproute2 / qdisc

Interf sieciowy (łącze) posiada kolejkę ramek do wysłania...
Qdisc = „dyscyplina kolejki”, czyli zasada działania kolejki
Polecenie iproute2 wyświetlające qdisc dla eth0, kolejka typu „pfifo_fast”

```
root# tc qdisc show dev eth0
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
```

Dyscyplina ta posiada 3 kolejki „bands” o priorytetach 0 (max pri),1,2
respektuje bity TOS pkg ip, na tej podstawie kwalifikuje pkg do odp kolejki...

UWAGA: obecnie pole tos ma inne znaczenie!!!
patrz: wikipedia Type_of_service, RFC 2474, DS field + ECN
(linux: dopiero w 2011 to zauważono, bit „min cost” jest zły...)

Binary	Decimcal	Meaning
1000	8	Minimize delay (md)
0100	4	Maximize throughput (mt)
0010	2	Maximize reliability (mr)
0001	1	Minimize monetary cost (mmc)
0000	0	Normal Service

TOS	Bits	Means	Linux Priority	Band
0x0	0	Normal Service	0 Best Effort	1
0x2	1	Minimize Monetary Cost	1 Filler	2
0x4	2	Maximize Reliability	0 Best Effort	1
0x6	3	mmc+mr	0 Best Effort	1
0x8	4	Maximize Throughput	2 Bulk	2
0xa	5	mmc+mt	2 Bulk	2
0xc	6	mr+mt	2 Bulk	2
0xe	7	mmc+mr+mt	2 Bulk	2
0x10	8	Minimize Delay	6 Interactive	0
0x12	9	mmc+md	6 Interactive	0
0x14	10	mr+md	6 Interactive	0
0x16	11	mmc+mr+md	6 Interactive	0
0x18	12	mt+md	4 Int. Bulk	1
0x1a	13	mmc+mt+md	4 Int. Bulk	1
0x1c	14	mr+mt+md	4 Int. Bulk	1
0x1e	15	mmc+mr+mt+md	4 Int. Bulk	1

2 kolejki priorytetowe... zasada działania:
jeśli nie ma pkg z high pri to wysyłaj z low prio

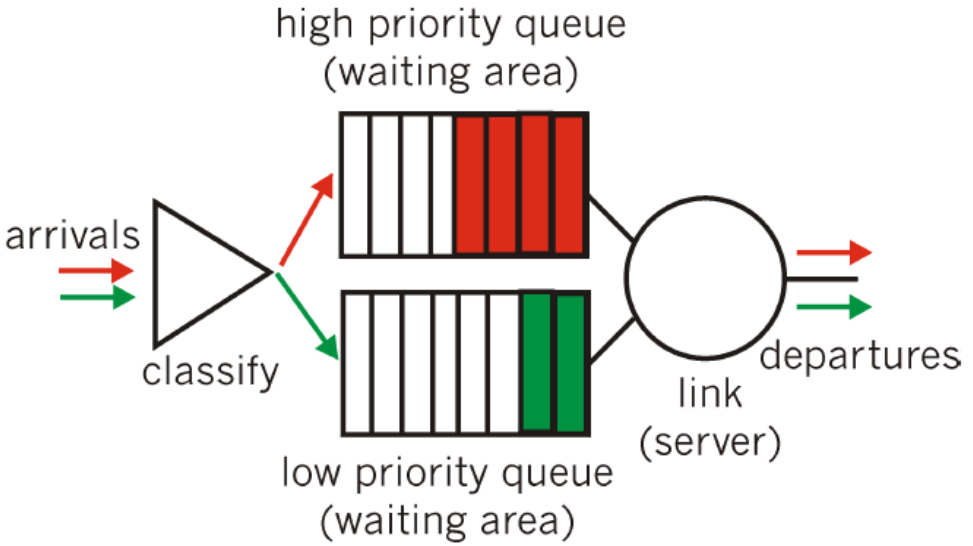


Figure 6.6-3: Priority queuing model

iproute2 / qdisc

Inne typy dyscypliny kolejek:

- mq (?), używane przez wlan0... ?
- SFQ (Stochastic Fairness Queue), „RR między połączeniami/strumieniam + hashi”
- TFB (Token Bucket Filter), „1 dziurawe wiadro”
- CBQ (Class Based Queue), pakiety są „klasyfikowane”, frakcja przepustowości na klasę
- HTB (Hierarchical Token Bucket), ”drzewo dziurawych wiader”, lepsze niż CBQ?

gdzie są opisy qdisc? np. https://man7.org/linux/man-pages/man8/tc-sfq.8.html

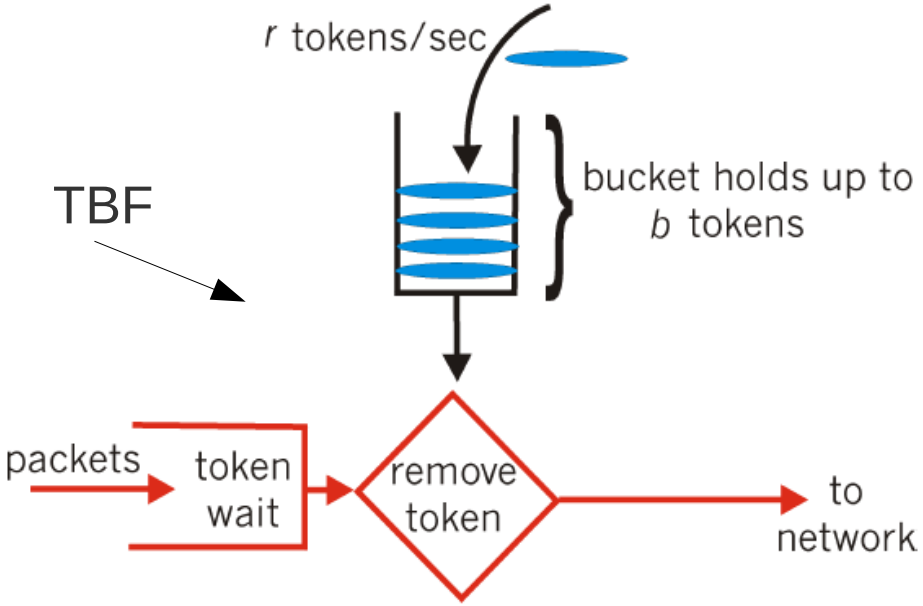


Figure 6.6-7: The Leaky Bucket Policer

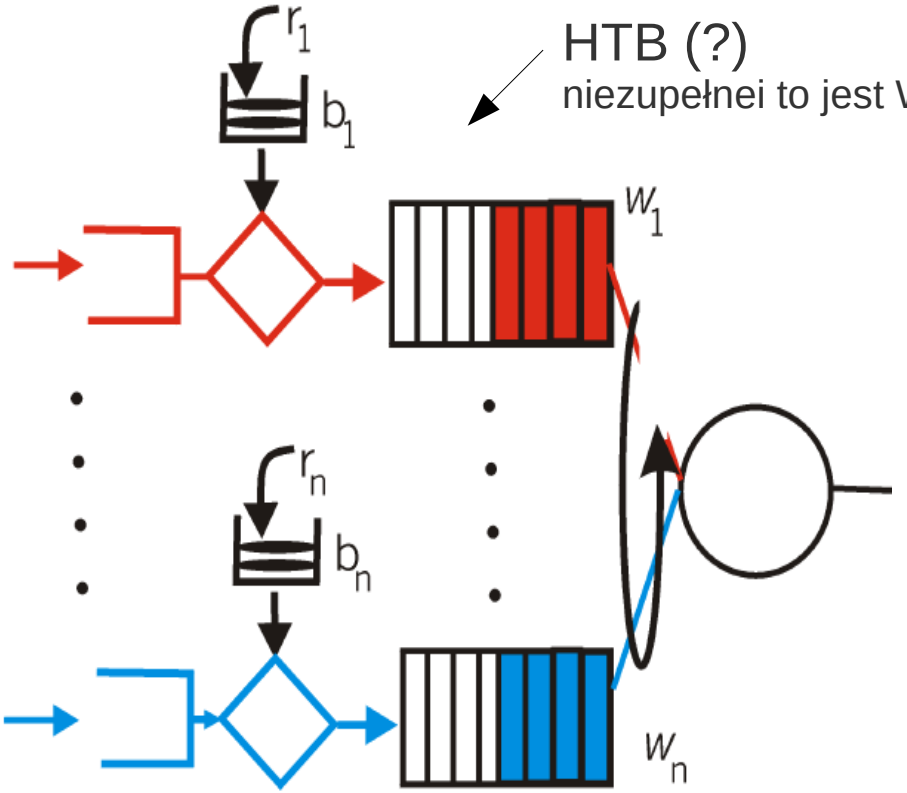


Figure 6.6-8: *n* multiplexed leaky bucket flows with WFQ scheduling

WFQ = Weighted Fair Queue

iproute2 / qdisc

Po co właściwie te „dyscypliny kolejkowe” ? (przypomnienie)
- chcemy inaczej traktować ruch voip niż ftp...
- obrona przed atakami: „dos”, np. zalewanie udp, SFQ vs fifo...
- klasyfikacja/ oznaczanie pkg + qdisc łączy routerów = **DiffServ**

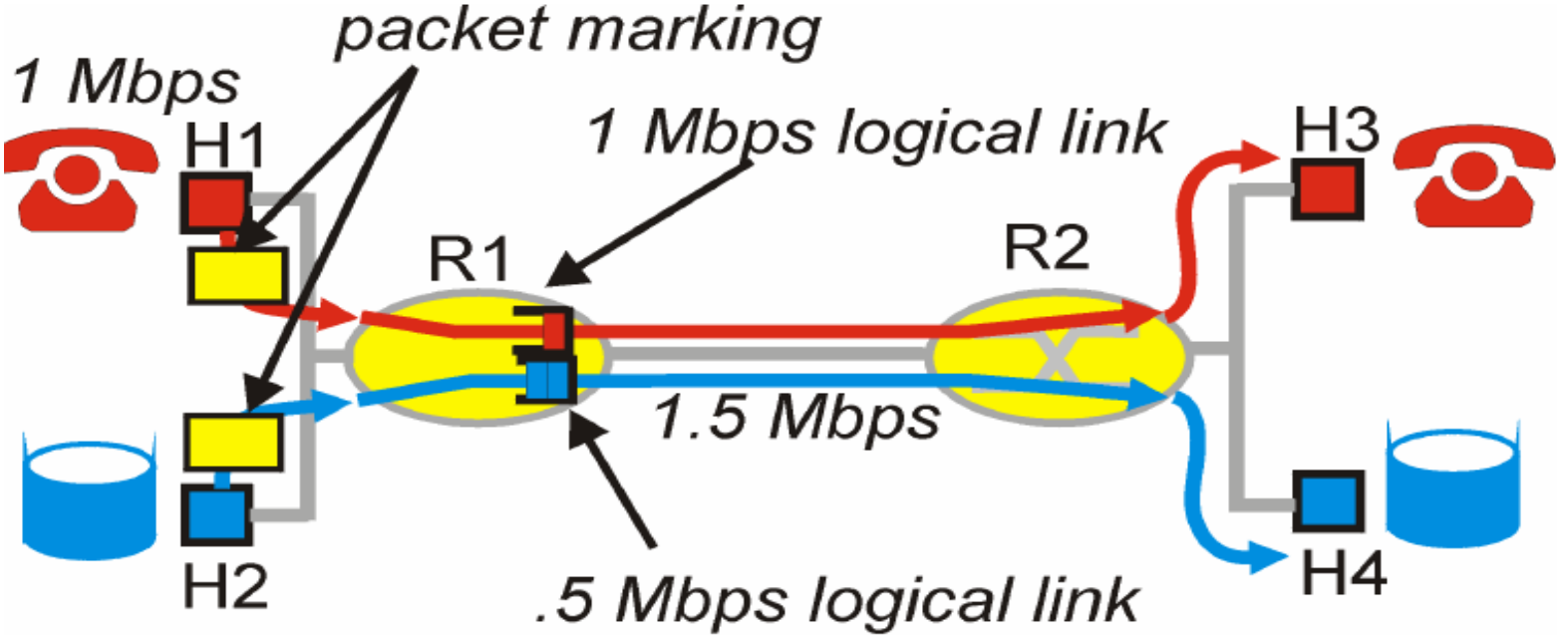


Figure 6.5-4: Logical isolation of audio and ftp application flows

Kłopotliwe pojęcia dotyczące qdisc (wg MH):
- classful vs classless:
  classful = jest klasyfikacja pkg, klasy są różnie traktowane
- shaping vs scheduling:
  scheduling= zmiana kolejności pkg, shaping = opóźnianie pkg