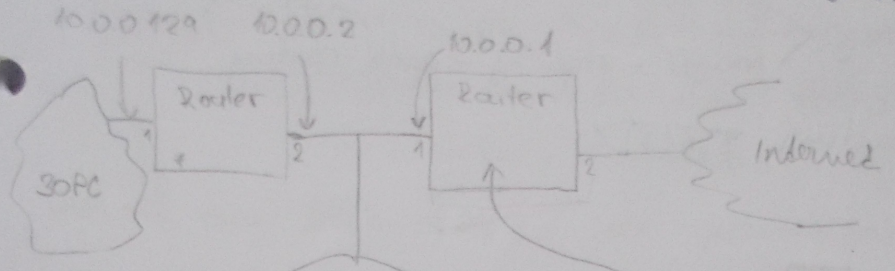
PSI-3

SMĚROVÁNÍ

25.2.

shell: netstat -r
route



10.0.0.129 ÷ 10.0.0.254
10.0.0.128/25 adresa sítě
10.0.0.255 BC

10.0.0.1 ÷ 10.0.0.126

10.0.0.0/25 adresa sítě

10.0.0.127 broadcast

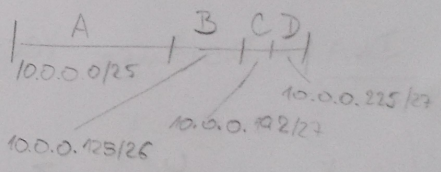
směrová tabulka

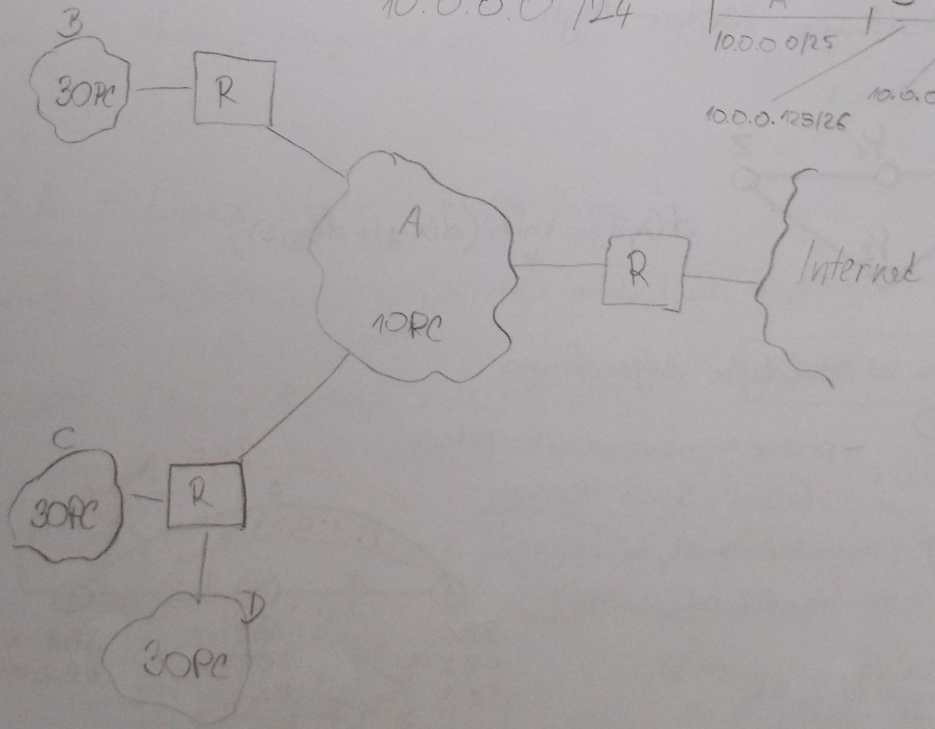
	port
10.0.0.0/24	1
0.0.0.0/0	2

*

10.0.0.128/25	1
0.0.0.0/0	2

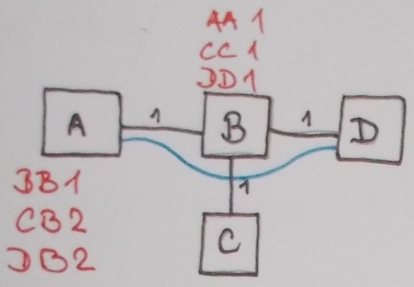
10.0.0.0/24





Směrování

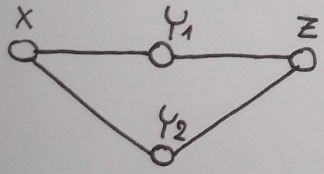
- graf routing
  - vnitřní - adaptivní
    - DVA
    - LSA
  - vnější - statický - path routing
- Source routing [B | A | D]



Směrovač
- směrovací tabulka
- směrovací algoritmy

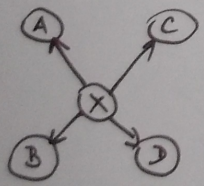
Směrování paketů

DVA - Distance Vector Algh.

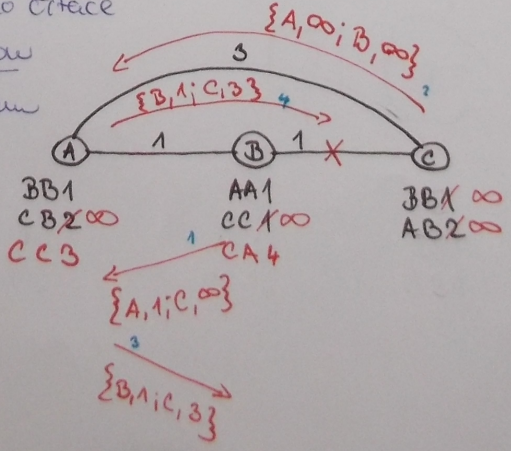


$$d(x,z) = \min_{+} \left(d(x,y) + d(y,z) \right)$$

Bellman - Fordův algoritmus



- problém nekonečného čítače
- ⇒ řešení: Split Horizon (neposílám zpět, to, co jsem se naučil od jiného)



⇒ řešení 2: Triggered Update (změnu pošlu dál mžitě)

⇒ řešení 3: Hold Down

PSI-3

RIP v1 : - max 16

(UDP 1520) - zprávy po 30s (pouze vector → cíl, vzdálenost)
- výpadek po 6 zpráv/ale 180s
- ∞ zůstalo v tab. 120s

UDP má min. 512B ⇒ 25 cest (vše v jednom paketu → konzistence)

RIP v2: - nevjal se, přišel pozdě, kompatibilita s RIPv1

= max neomezeně
- ověřování
- adresa sousedního uzlu

Algoritmus DVA:

nová > stará → zahodí
- stejný zdroj → použije se
- jiný zdroj → zahodí se

nová = stará → zahodí se

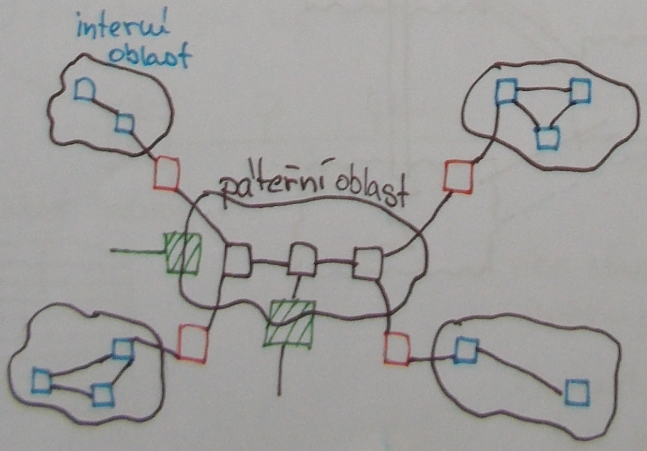
nová < stará → použije se

LSA - Link State Algorithm

- používá Dijkstrův alg. (potřebuje znalost celé topologie)

OSPF (Open Sortest Path First)

OSPF ≡ IP



- hraniční router
- páteřní router
- interní router
- hraniční směrovač autonomní oblasti

1) vnitřní cesta
2) vnější cesta

metrika: $\frac{f_{max}}{f}$

- OSPF podporuje: - více cest (přepínání)
  - různé typy směrování
- OSPF - výpočetně náročnější než RIP, ale méně datově náročný

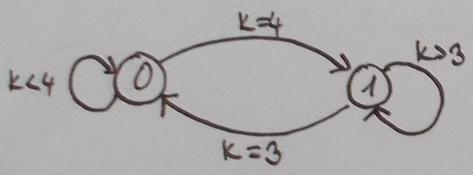
VNĚJŠÍ SMĚROVÁNÍ

EGP - Exterior Gateway Protocol

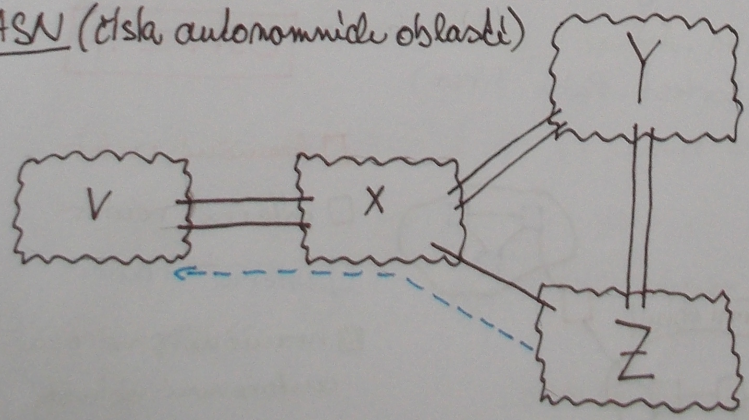
BGP - Border Gateway Protocol

- používá path routing (definuje kudy se směruje)
- nosný je protokol IP
- alg. zjišťování života souseda: k ≥ N

N = 8



ASN (čísla autonomních oblastí)



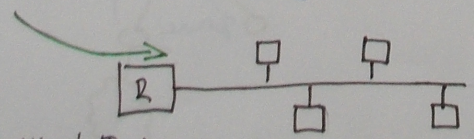
PSI-4 **INTERNET MULTICAST** 4. 3. 2015

Replikovaný unicast
- zpráva se pošle na server a ten ji rozešle

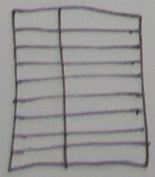
Multicast
- zpráva pro skupinu (mají skup. IP adresu)
- zdroj nemusí být ve skupině

① Ethernetová karta na PC má několik adres (MAC)
- individuální adresu (začíná 0) IP ⇒ MAC
- broadcastovou adresu (samé 1) (pro boot, ARP)
- několik skupinových adres (začíná 1)
- režimy:
  1) blokovaná (nic neprijíma)
  2) přijímá svojí fyz. adresu
  3) fyz. adresa + broadcast
  4) 3) + limited multicast (ty, kteří jsou def. přímo na kartě)
  5) 3) + multicast
  6) všedmo (tzv. promiskuitní režim)
  - vhodné pro režimy switch, AP, ...

② Protokol pro reg. hostů (IGMP = Internet Group Management Protocol)



Multicast IP adresy



IGMPv1: výzva 224.0.0.1 →
stanice po náhodné době odpoví 224.0.X.y ←

více stanic se stejnou MCAST IP adresou
1) plán odpovědi (každá stanice různě)
2) pokud odpoví někdo dříve než já, tak už sám odpovídat nemusím

IGMP v2: přidává možnost odhlášení

odhlášení

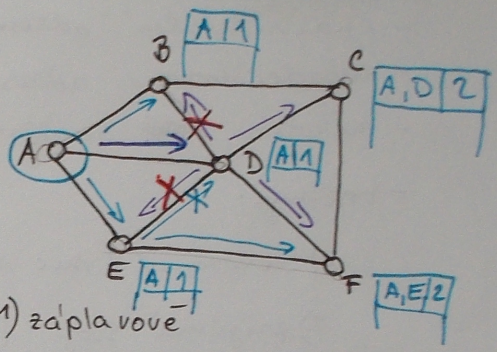
výzva pro dotaz skupinou adresu

IGMP v3: může si vybrat od kterého zdroje chce poslouchat multicast zprávy

③ Protokoly pro směrování

- interní:
- externí:

~~[illegible]~~



1) záplavové

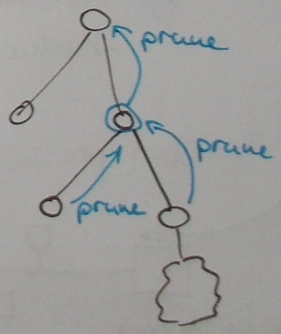
② vytvořit doručovací strom

- alg: reverse path forwarding

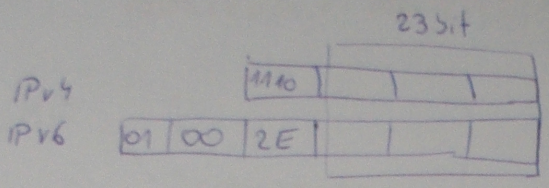
3) CBT (Core Base Tree)

vylepšení interního směrování:

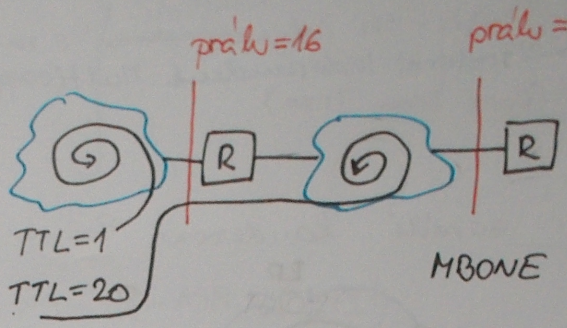
- join (připojení ke stromu)
- prune (odtržení)



7SI-4



- rozsah doručování
  - podle TTL



  - administrativní omezení

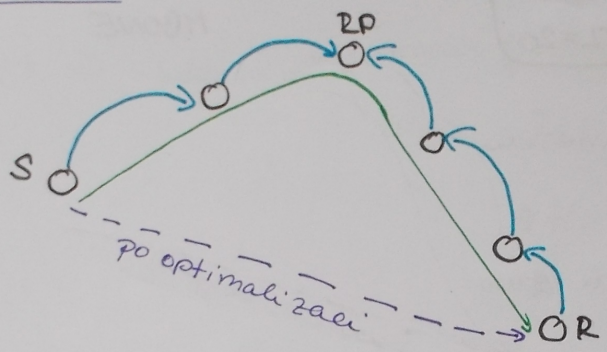
- rozdělení skupinových adres:

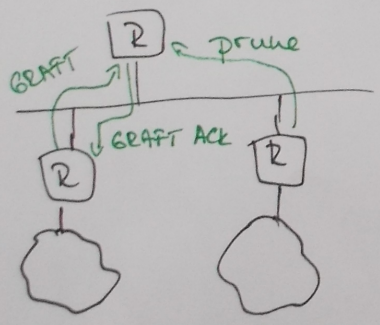
224.0.0./24	– lokální
224.0.1/24	– globální
224.1/16	– ~~SM~~ ST Multicast Groups
233/8	– GLOP block (233.X.Y.0) (pro autonomní oblasti)

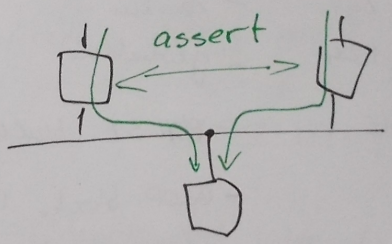
Protokoly pro skupinové směrování

- DVMRP (
- MOSPF (Multicast OSPF)
- PIM
  - PIM-DS (postupně se ořezává)
  - PIM-SM (client musí o zasílání požádat)
  - Protocol Independent Multicast
- CBT (Core Base Tree)

PIM-SM: zavádí Randevous Point (RP)







PSI-5

PROSTŘEDKY PRO ŘÍZENÍ POČ. SÍTÍ/CH

- ICMP — Ping
  \ Traceroute
- 'WireShark' (Packet Sniffer)
- TFTP (přenos dat)

} SNMP — identifikace proměnných
  \ přenos dat — kódovací metody (GET, SET)

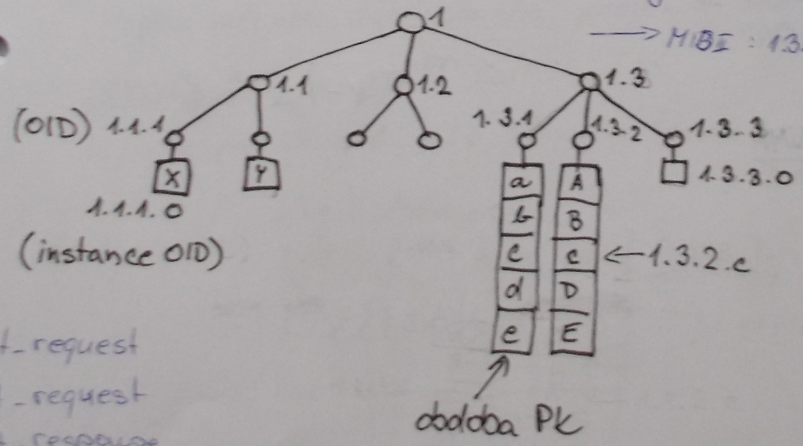
SNMP - kódování — popis - ASN.1 (SMI)
  \ zápis - BER (Basic Encode Rule)

tlv: type | length | value

SNMP-ident. prom.: identifikátory řešeny pomocí různých stromů

ISO | ITU | ISO-ITU

OID (Obj. ID): 1.3.6.1.2 (iso.org.dod.internet.mgmt)



→ MIB: 1.3.6.1.2.1

get_request
set_request
get_response
get_next_request
trap (asynchronní)

⇒ zavedena metoda get_next

get_next 1.1.1
return 1.1.1.0 X

Proxy agenti

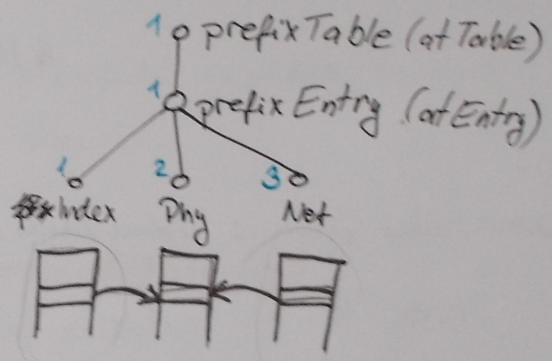
11.3.2015

- něco : konverze protokolu
  - data
- používají se tak, kde není dostupný SNMP nebo data strukturu
  MIB

Reprezentace dat

- přístupová práva OID:
  - read-only
  - read-write
  - write-only
  - not-accesible

MIB – definice tabulky:



příklad

Net: 147.228.67.1

Index: 1

Phy: 1.1.2 | . 1 . 147.228.67.1

OID — instance OID

TLV – type length value

Mechanizmus SNMP

{ ver, community, type, id, err { { OID, zvol. OID }, { OID, zvol. OID } } }

TLV 2.1.0

TLV 6.6, Public

TLV TLV

TLV

Sekvence 16

délka 16

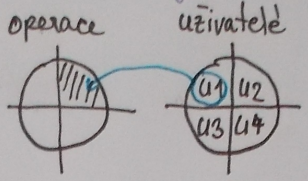
- ověřování pomocí community name

PS1-5 11.3.2015

SNMP v.2c

- v verze 1 rodila bezpečnost, neznalost některých nástrah takových dat, celá množina dat má pouze jeden přístup

- ve verzi 2 : šifrování, víceúrovňový přístup, hierarchické řízení mezi mouit stanicí a agenty jsou ještě mezilehlé stanice pro omezení zátěže

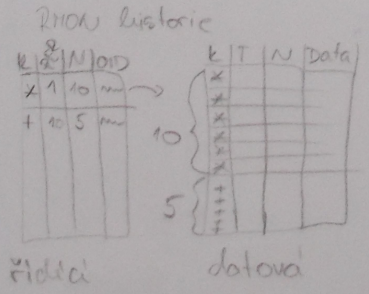


SNMPv3

- vrcholná verze, skládá se z modulů → z kterých se tvoří agent i monitorovací stanice

RMON (Remote MONitoring)

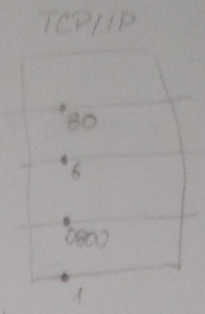
- nadstavba SNMP (spíš jeho rozšíření)
- místo agentů používá sondy (jsou více autonomní)
  - ↳ pro přenos dat používá protokol SNMP
  - ↳ nejčastěji jako karta do SWITCHE
  - ↳ pouze pro Ethernet nebo TokenRing (pouze fyzická úroveň)



k - klíč
PER - perioda vzorkování
N - počet záznamů
T - čas zaznamenání
N - číslo vzorku

RMON 2

- síťová nebo aplikační úroveň



délka
↓
16 . 0.0.0.1 ; 0.0.8.0 ; 0.0.0.6 ; 0.0.0.80,

+ 4.0.0.0
↑ délka speciální atributy

Neurses → knihovna pro TUI

PS1-6

18.3.15

způsoby přenosu

- dělení do bloků - fragmentace
- interpretovat — současně
  - výběr (html, plaintext, obrází)
- posloupnost bloků
- bloky rozdělují stringové separátory
- zpráva i přílohy jsou jsou bloky (text + 2 obrázky = 3 bloky)

outlook — ~~webmail~~ zabezpečení

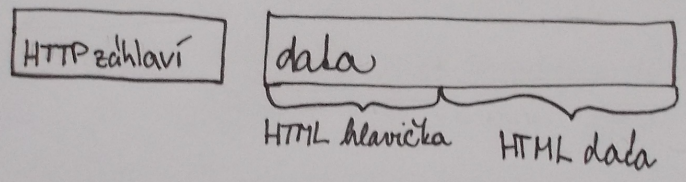
pop
imap

- pomocí SSL

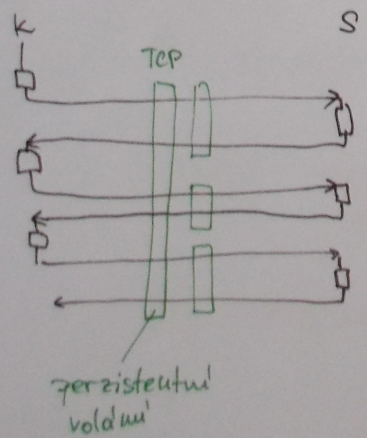
HTTP

- GET, HEAD, POST, PUT, DELETE, OPTION, TRACE, CONNECT

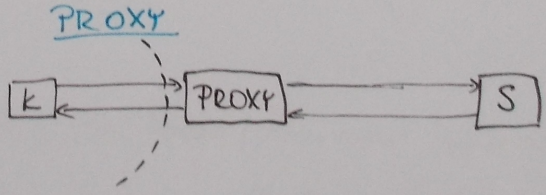
formát zprávy



přenos

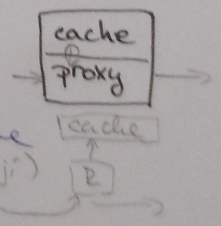


PROXY



CACHE (vyrovnávací paměť)

- lokální
  - v lokální síti
  - transparentní cache (není vidět, nekonfigurují ji)



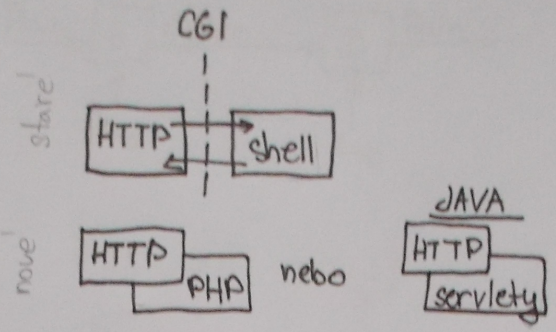
- potřeba lepší a více cache ⇒ CDN

COOKIES

- HTTP je bezestavový → stavovost zařízena pomocí cookies

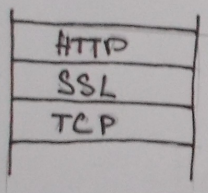
Stránky

- statické
- dynamické
- aktivní



interpretace na straně klienta - Java Script, Java Aplety

Bezpečnost



přihlášení
- anonymní
- neanonymní

- malu certifikát a je zaručeno, že komunikuji s tím serverem se kterým jsem chtěl

URL

obecný tvar:

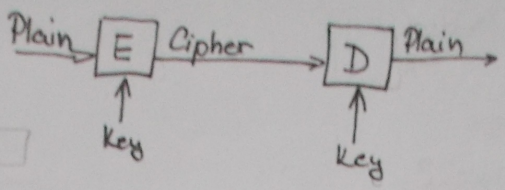
schema://uživatel:heslo@stroj:port/cesta/dokument?parametry#odkaz_uvnitř_dokumentu

telnet:// → vyvolá klienta pro telnet

rmi://stroj:port/názevSlužby

ŠIFROVÁNÍ a BEZPEČNOST

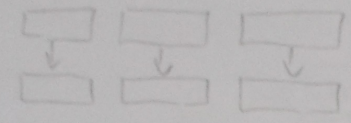
- algoritmy – symetrické (DES, 3DES, IDEA, AES)
  (56b) (112b) (128, 192, 256b)



Obecně

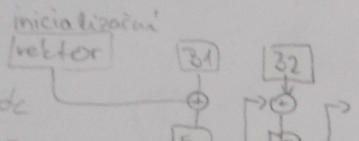
metody:

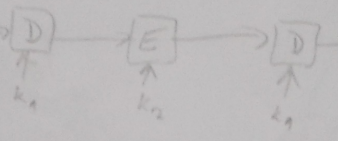
ECB



CBC

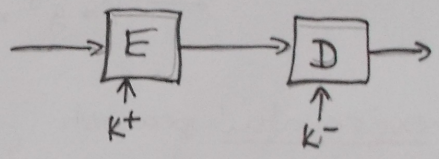
Chain Block Cipher





DES

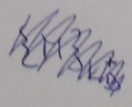
- asymetrické (RSA: 2k-4k)
  - pro distribuci šifr. klíčů
  - \ pro el. podepisování
  - k^+ – veřejný
  - k^- – tajný



- hashovací funkce – SHA1, SHA2, SHA256 (128B)
  - neexistuje inverzní funkce
  - \ MD5 (160B)

$\exists\ h(M_1) = h(M_2)$

- operace
  - šifrování: $\{M\}_k$, $\{M\}_{k^+}$

A → B : 

$\{M\}_k \{k\}_{k^+_B}$

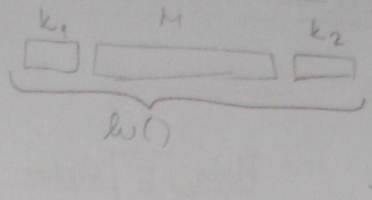
- zpráva šifruji symetrickým
- klíč potom asymetrickým veřejným klíčem příjemce

-podpis

$A \to B: \{M\} \to h(M)$

$\{M\}\{h(M)\}_{k_A^-}$

MAC



-výměna klíčů

$\{k\}_{k_B^+} \longrightarrow$

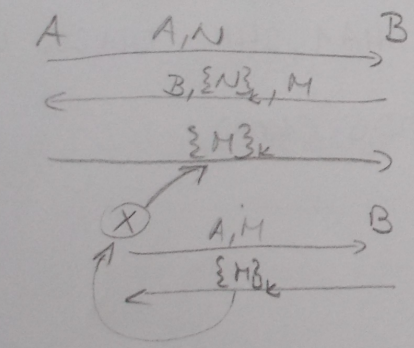
Diffie-Hellman

$g^x \bmod N \to X, g, N$

$\leftarrow y = g^y \bmod N$

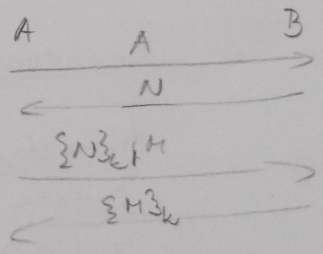
$\Rightarrow X^y \bmod N = y^x \bmod N = K$

-ověřování pravosti



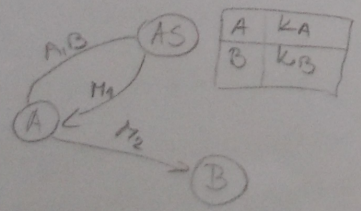
možnost MITM útoku

přímé ověřování



správné ověření

nepřímé (přes autorizační server)



$M_1: \{k_{AB}, \{k_{A,B}\}_{k_B}\}_{k_A}$

PSI-7
Lekce 9

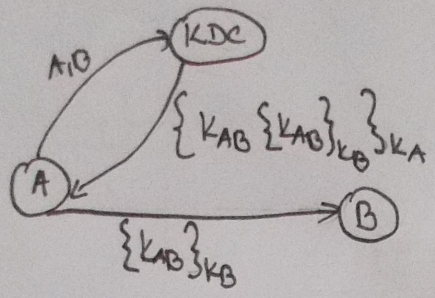
OVĚŘOVÁNÍ A PROTOKOLY

25.3.15

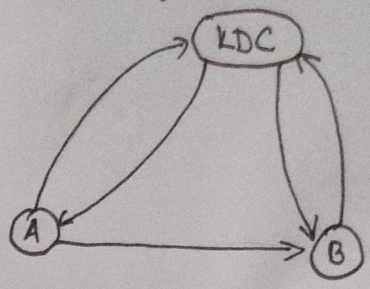
- ověřování
  - přímé
    - symet.
    - asymet.
  - nepřímé (ověřovací servery) < sym / asym + generování relačního klíče

Needham-Schroeder

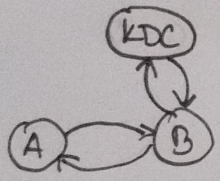
a) symetrické



asymetrické



Otway Reese



CERTIFIKÁTY

- certifikát obsahuje veřejný klíč
- info o vlastníkovi, (vydavateli)
- identifikační údaj (obdoba PK)
- platnost
- účel

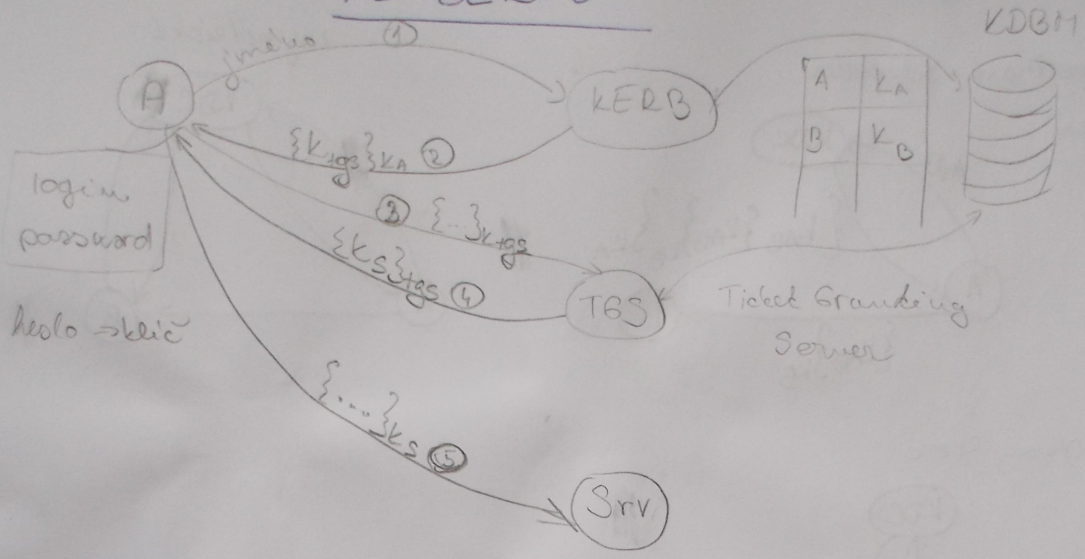
} zabezpečení pomocí cryptogr. kontrolní součet ⇒ zašifrováno (podepsáno) tajným klíčem toho, kdo to vystavil

- veřejný klíč CA je součástí dalšího certifikátu

PGP certifikáty (Pretty Good Privacy)

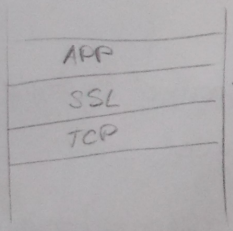
- šifrování a podepisování zpráv e-mailových

KERBEROS



- symetrické šifrování

PROTOKOL SSL



skládá se 4 podprotokolů:

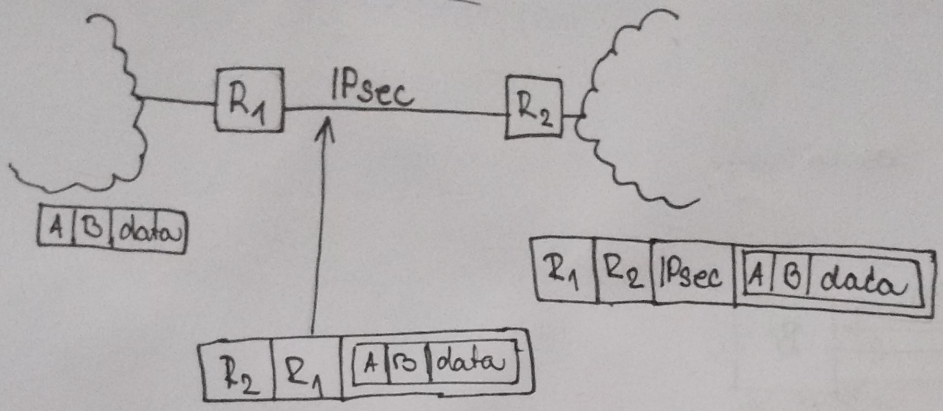
- record protocol (komunikace)
- handshake protocol (výměna klíčů)
- alert
- Change Cipher Specification Protocol (pro změnu šifrovacího klíče za běhu)

- data se šifrují symetricky pomocí relačního klíče
- k výměně relačního klíče se používá asym. šifrování

SSH

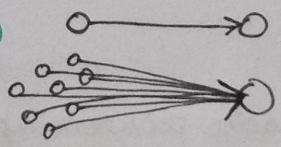
- ssh, scp, ssh tunel
- protokol na aplikační úrovni

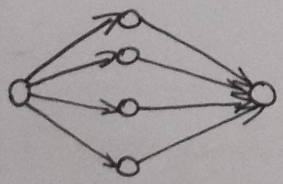
IPsec



ÚTOKY

DoS (Denial of Service)
DDoS (Distribuate DoS)
RDoS (Reflexní DoS)





Prostředky k útoku:

ICMP, TCP, UDP, DNS

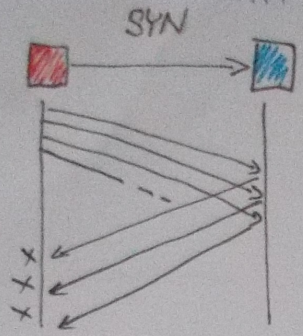
Detekce útoku

Snort (Open Source Intrusion Detection System)

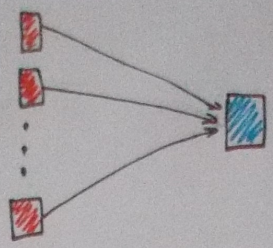
Postup

1) analýza – skenování portů
2) útok

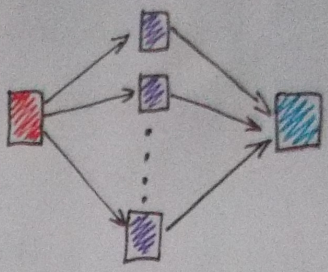
DoS — DoS flood
— DoS MitM



DDoS

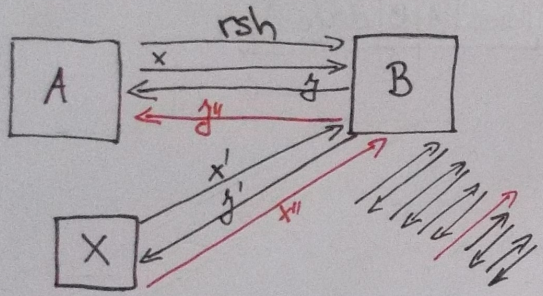


RDoS

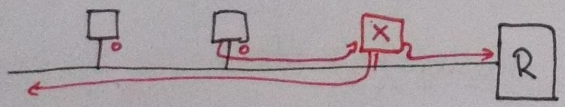


Únos relace (session hijack)

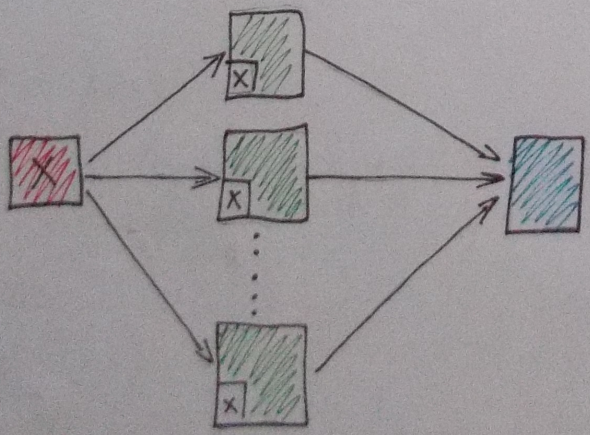
TCP



DoS MitM: ARP cache poisoning



DDoS



1) útočníkův sw trvale na portu
=> odhalení => ICMP

2) odposlouchávání komunikačního kanálu zpráv (příkazů) => šifrování

MULTIMEDIÁLNÍ PŘENOSY

8.4.2015

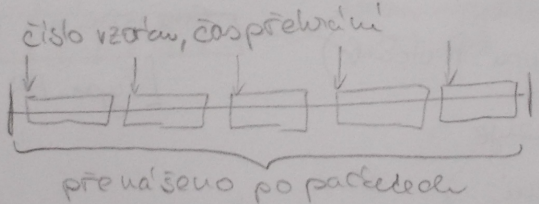
stream (oběžný) – už při stahování lze přehrávat
\ bez stahovat obraz a zvuk z různých serverů

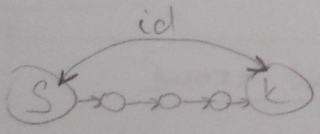
implementace: řídící spojení – TCP
– RTSP (Real Time Streaming Protocol)
datové spojení – UDP
– RTP/RTCP (Real Time Protocol / Real Time Control Protocol)

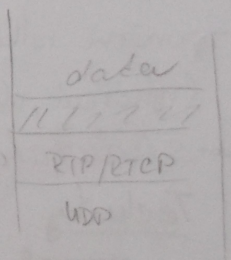
prof. JAN – síťový odborník, streamování přednášky

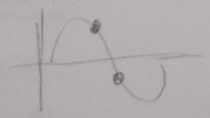
RTSP
- postaven na bázi HTTP
- vlastní informace přenášeny pomocí SDP souboru

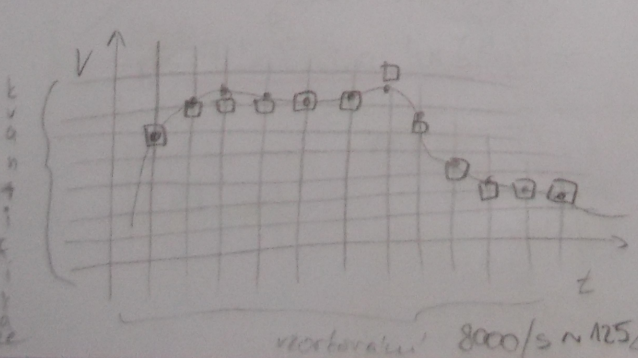
RTP/RTCP
- nesen UDP

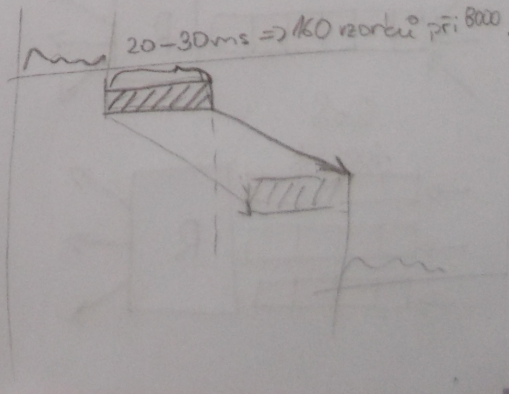












Quality of Services v IP sítích

15.4.2015

- best effort (podle nejlepšího co může) ← zpoždění, rozptyl, chybovost

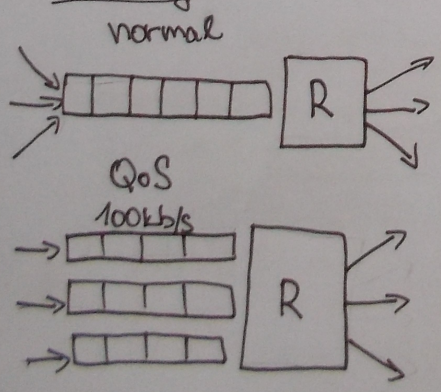
→ multimediální služby
- přenos real-time ⇒ kap. kom. kanálu
⇒ definované zpoždění
⇒ omezený rozptyl
⇒ vliv na chybovost
} splňovali sítě ATM (AT&T)

IP sítě
- integrated services
- differenciated services

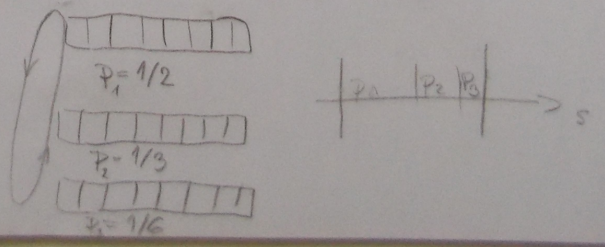
integrované služby
- vytvoření virt. kanálu
- stavové - směrovače
- RSVP (Reservation Protocol)
- poměrně velká režije (vytváření, ukončení, ...)
- škálovatelné!

diff. services (rozlišované služby)
- bezestavové - označení jednotl. paketů (protokol, adresa, ...)
- směrování podle značení (TOS) [Type Of Services]

Techniky



Plánování front
- FIFO (cyklická obsluha)
- Prioritní fronty
- Vážené prioritní fronty

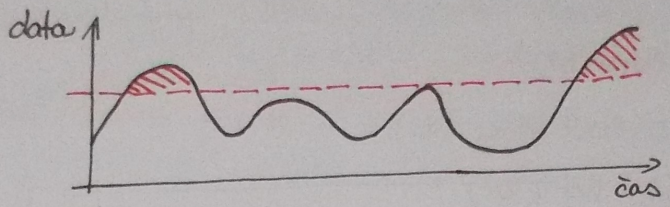


QoS IP

15.4.

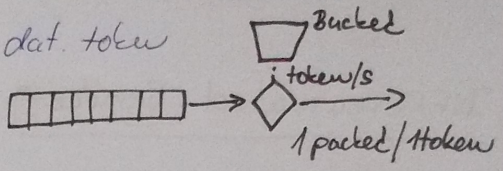
Filtrování požadavků

- uživatel – rezervace kanálu (ind. služby)
  - kontrola akt. stavu

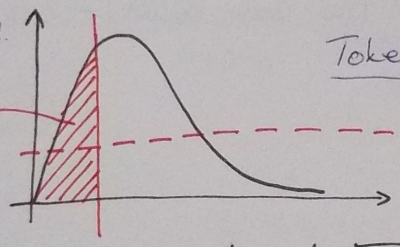


- algoritmus ořezávání dat. toku

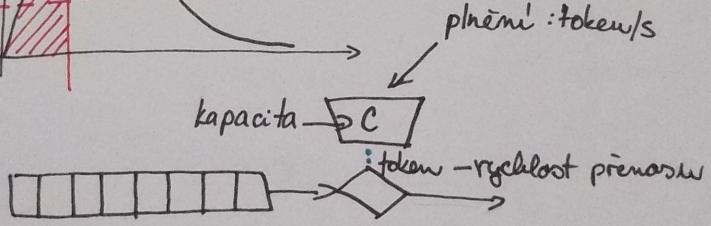
Leakey Bucket :



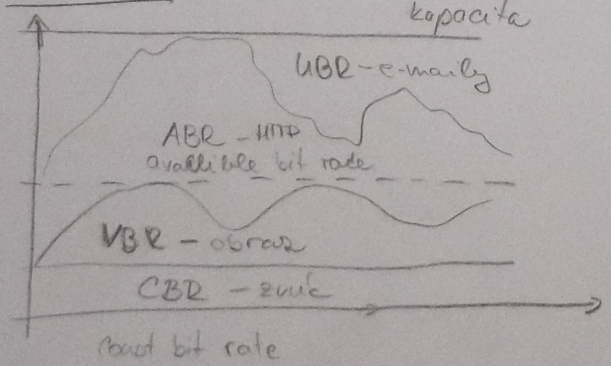
- jedno plnou rychl. dobu ji nevyčerpám kapacitu tokenů ⇒ ⇒ potom je rychlostí plnění tokenů



Token Bucket



Př. ATM



Integrated Services in RSVP

— zprávy
- RESV - prostředek rezervace
- PATH - nabízí parametry

- 3 druhy rezervace (RFC 2205)
  - Fixed Filter (FF)
  - Shared Explicit Filter (SE)
  - Wildcard filter (WF)

Differenciated Services

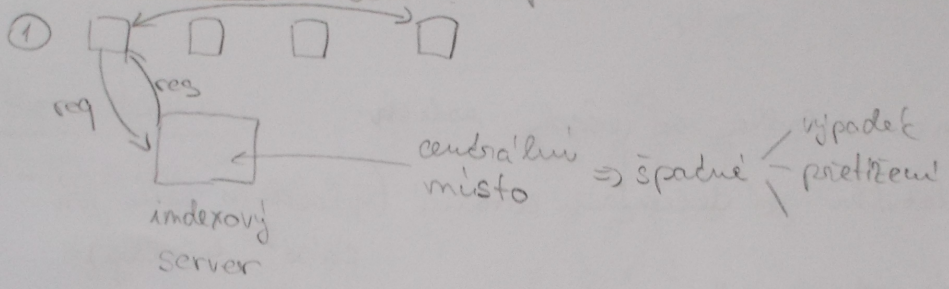
- klasifikace pomocí DSCP (DS Code Point) – 6 bitů

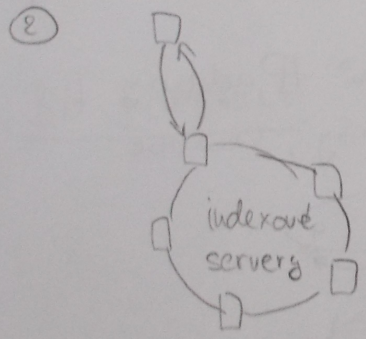
P2P SÍTĚ

uzel — klientská část, serverová část → servent

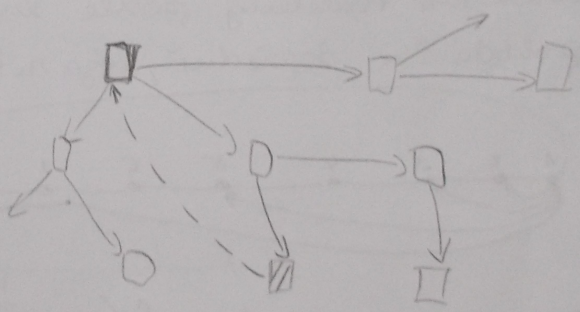
- nestrukturované – uložení – lokálně + repliky
  vyhledávání – záplavovací algoritmus dotazů
- strukturované – uložení – podle algoritmu
  vyhledání – podle algoritmu

Nestrukturované data transfer

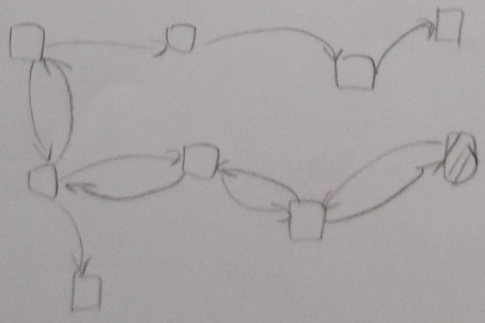




3) Decentralizovaný systém



4) Anonymní : FreeNet



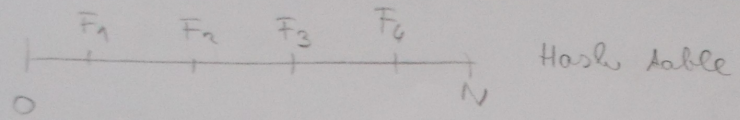
Strukturované

- Vime kde informaci hledat → put (key, addr)
  addr=get (key)

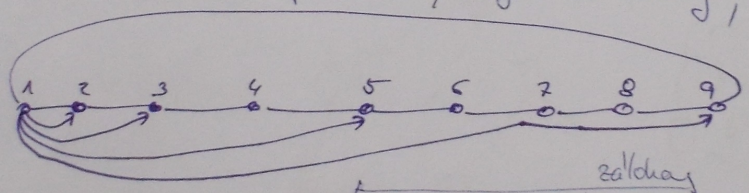
topologie

- kruhové, n-rozměrný prostor, hierarchické (CHORD)

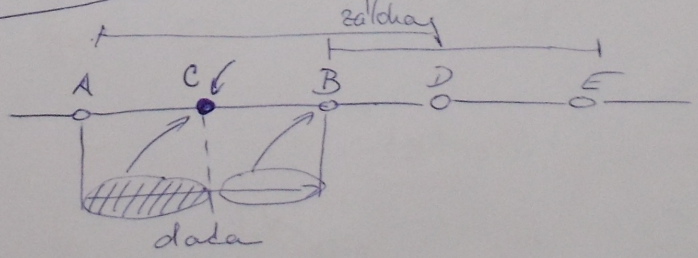
Kruhové (CHORD)



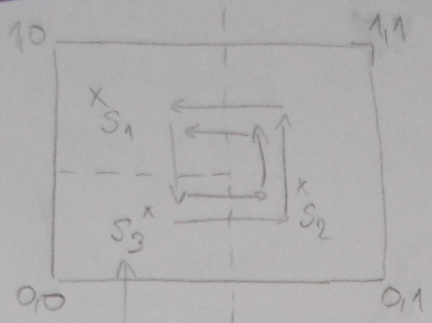
- hashovací tabulka ve všech uzlech
  hash. tabulka je decentralizovaná (v každém uzlu jen část tabulky)
  - zbytek od n do odkazuje IP addresou na server, kde je příslušná hash tabulka
  - hash. tab. rozdělený podle hashů IP adresy
- prohledávání dopředu, logaritmicky, kruhově



Přidání uzlu :



CAN



$x = h_1(file)$

$y = h_2(file)$

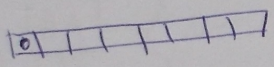
tab. sousedů

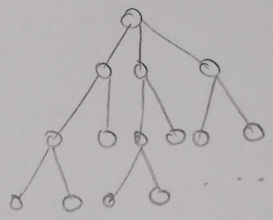
S_1	IPaddr	x	y
S_2	IPaddr	x	y

Hierarchické

- Pastry, Tapestry, ...

$k \leftarrow h( )$





Bit Torrent

PSI

22.4.2015

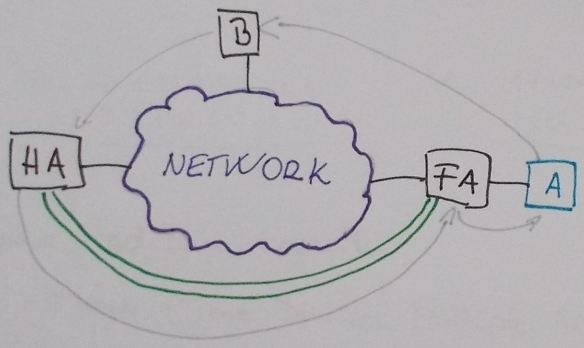
IPv6

IPv4 - 4 složky; málo adres → NAT ⇐ stejně dojdou

IPv6 - 16 složek (128 bitů) ; 2^{96} více adres, $3{,}4 \cdot 10^{38}$

- hierarchické přidělování adres ⇒ min. 65 536 subsítí pro kaž dého
- IPsec, autokonfigurace (Plug&Play)
- MIPv6 - mobilní verze

MIPv4

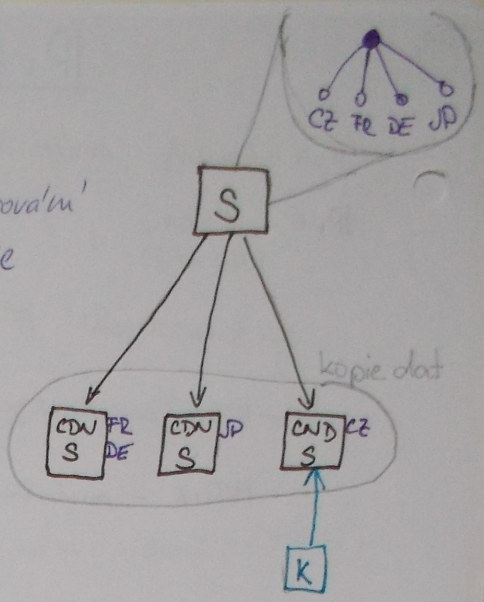


IPsec

- Kontrola
- Šifrování
- Autentizace
- Integrita
- Transportní

CDN (Content Delivery Network)

- více násobné kopie
  - read only
  - dynamické + kopírování zdroje
- výměna dat (synchronizace) mezi CDN servery
- synchronizace mezi serverem a CDN servery
  - pomocí zpráv
    - PUSH (propagace změn)
    - PULL (na vyžádání)

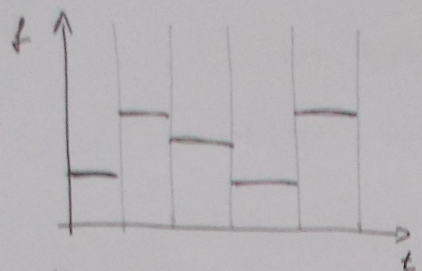


! Pozn.: Přečtení sociální prezentace nutné!

BEZDRÁTOVÉ POČÍTAČOVÉ SÍTĚ

- systémy prac. v rozprostřeném pásmu

a) přeskok. mezi frekvencemi (FHSS)



b) modulace jednotl. bitů sekvencí změn signálu (DSSS)

0 = 0 0 1 1 0 1 0 1 } přenos 0 ... (-1)
1 = 1 1 0 0 1 0 1 0 } 1 ... (+1)

0 ... -1 -1 +1 +1 -1 +1 -1 +1
1 1 0 0 1 0 1 0
−4

1 ... +1 +1 -1 +1 -1 +1 -1 -1
1 1 0 0 1 0 1 0
+4

Metody sdílení kanálu

- TDMA (Time Division Multiple Access)
- FDMA (Frequency ...)
- CDMA (Code ...)
  - synchronní → ortogonální vektory ($a \cdot b = 0$)

$$A: \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \end{bmatrix}$$

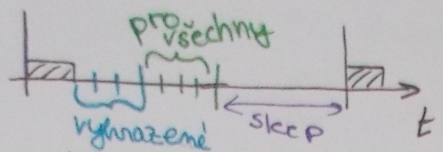
A chce přenést: 10 {1 -1 +1 -1} {-1 +1 -1 +1}
B — " — : 01 {-1 -1 +1 +1} {+1 -1 +1 -1}

{0 -2 2 0} {0 2 -2 0} - součet signálů

A 1 0 1 0 1 0 1 0
0 0 2 0 0 0 -2 0

B 1 1 0 0 1 1 0 0
0 -2 0 0 0 2 0 0

Beecon rámec



vyhrazené
konkrétním
stanicím

PE 1 11.2.15

Literatura: IBM redbooks : TCP/IP Tutorial and Technical Overview

Bee; „IPC Unix"

Šmrha, Rudolf: Internetworking pomocí TCP/IP

Comer: Internetworking with TCP/IP

knihovny: Pcap (JPCap)

TCP/IP — NIS (synchronizuje UID, GID ze vzdáleného souborového systému), NFS (Network File System), XDR (eXtended Data Representation), (Network Inform. system)

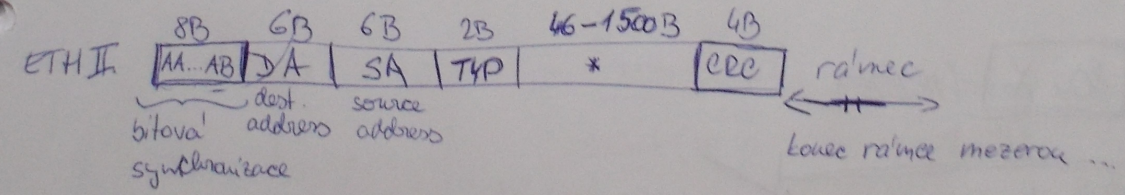
Aplikační (FTP, RPC)

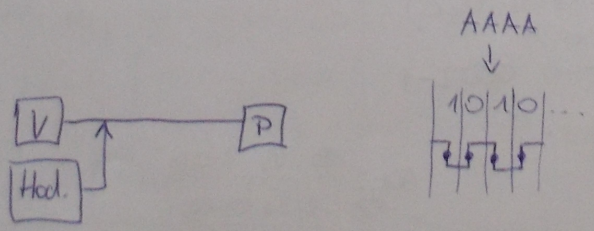
4 TCP | UDP (Transportní)

3 IP (Síťová)

1-2 Přenosová

Ethernet 4 – kódování: Inverzní Manchester

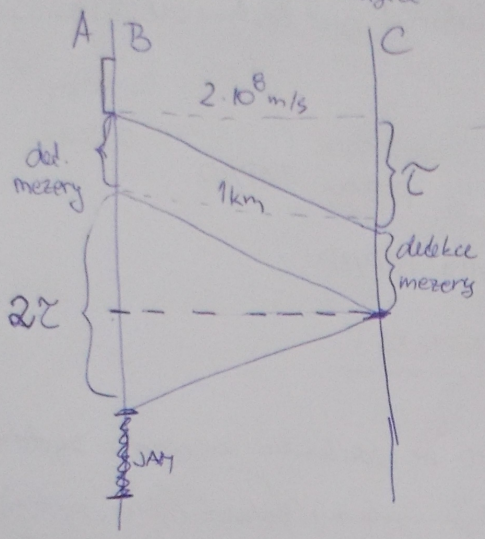




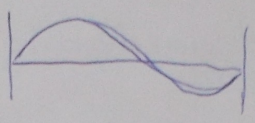
– kódování na straně vysílače kvůli fázi (zpoždění)

* Proč 46B?

CSMA naléhající



koaxiální kab. má koeficient zkrácení $K = 0,7$



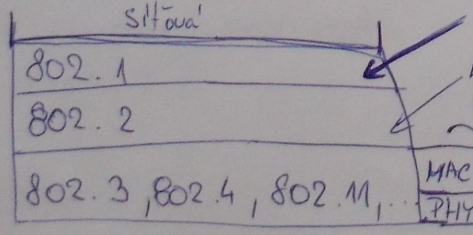
- ve vakuu ~~vlnová délka~~ má vlna 1Hz délku 300 000 km
- ve hmot. prostředí je délka k-krát kratší

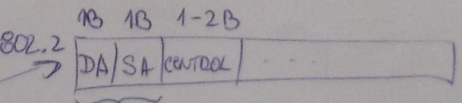
cílové adresy:

individuální (unicast) 0xxx...
multicast (skupinové) 1xxx...
broadcast (samé 1)

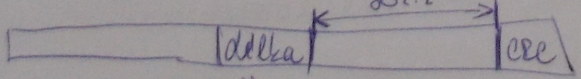
zdroj. adresa: pouze individuální

802.X (IEEE)





802.3 (rámec typu Ethernet)



⇓

proto musí být u EthII. typ min. 1500

802.2 — (nečíslované) rámce nespoj. služby

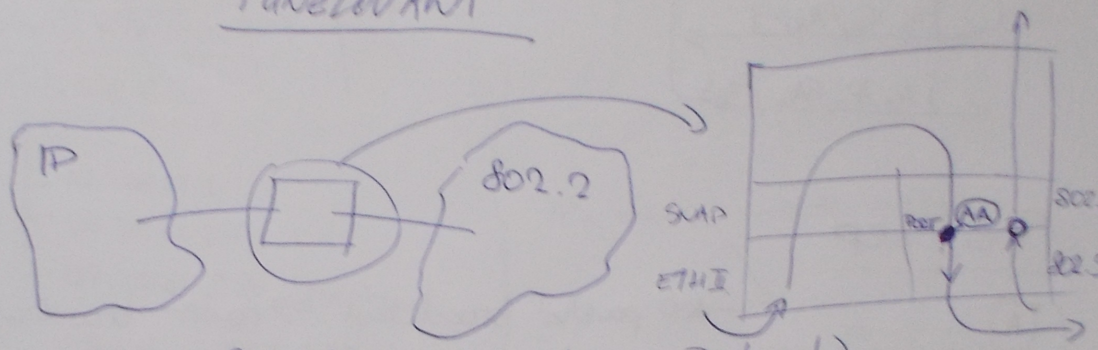
— rámce spoj. služby – navazání, ukončení, přenos, reset, ...

— STOP & WAIT (potvrzovaná nespojovaná služba)

} určuje řídící byte (CONTROL)

podobné HDLC

TUNELOVÁNÍ



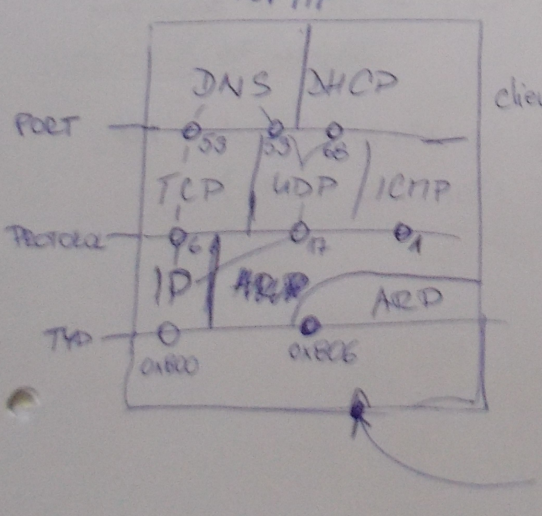
SNAP (Sub Network Access Protocol)

ETH II. |DA|SA|TYP| |CRC|

802.3 |DA|SA|LEN|802.2 DSAP SSAP CTRL | DATA |CRC|

0xAA 0xAA — 3B TYP + INFORMACE — výrobce

TCP/IP



BSD Sockety

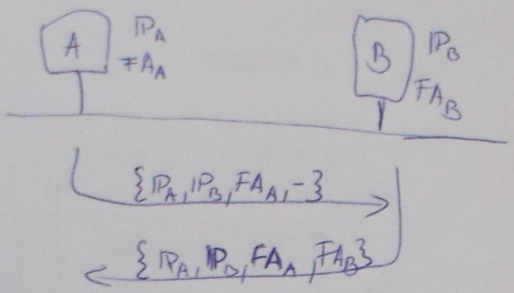
client { socket(AF_INET, SOCK_STREAM, IPPROTO_TCP) — TYP, PROTOKOL
connect(... IP addr., PORT ...) — PORT

TYP rozlišuje IPv4 a IPv6

socket(
listen(
bind(
accept(

ARP (Address Re Protocol)

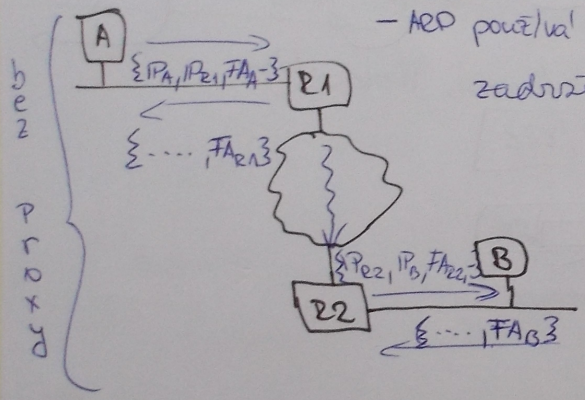
- převádí IP na fyzickou adresu (RARP : FA → IP)
  nepoužívá, nahrazen DHCP



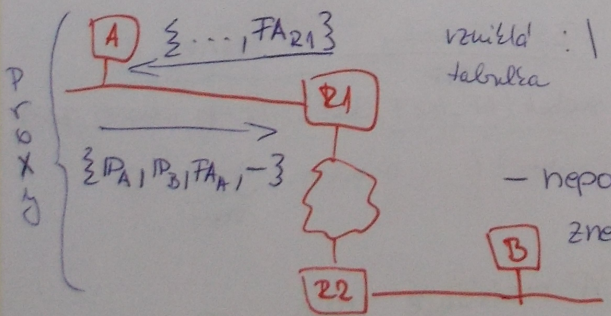
ARP tabulka

IP	FA	TIME

Proxy ARP



- ARP používá broadcast ⇒ router broadcast zadrží (takže nepotřebujeme znát FA_B)



vzniklá tabulka: $| IP_B | FA_{R1} |$ – více síť. adres má stejnou FA

- nepoužívá se z bezp. důvodu ~~MITM~~ zneužití (Man in the middle)

Směrování

pasivní směrovač

cíl	maska	router	rozhraní	typ
R1	?	—	eth0	U
B	B	R1	eth0	UR

U – up
R – router

147.228.67.0 | 255.255.255.0
0.0.0.0 | 0.0.0.0

IP adresa sítě (subsítě)

147.228.67.0/24

- 147.228.67.1
- 147.228.67.254
- 147.228.67.255 broadcast

maska: 255.255.255.0

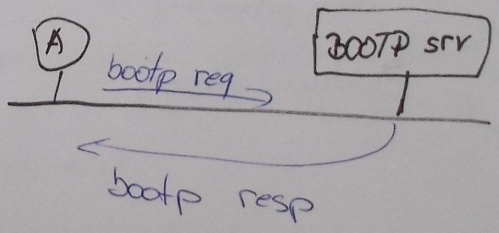
IP x maska → 147.228.67.0

BOOTP

bootování

- získání param. (BOOTP)
- zavedení OS

| FA | TYP | HODNOTA |



- ipadresa
- maska (podsítě)
- adresa routeru (výchozí brána)
- adresa DNS

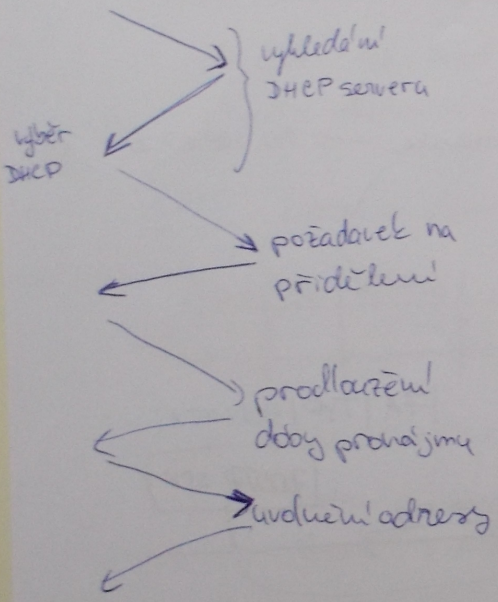
pevná část	IP adresa, adresář, ...		
prom. část	99	... MAGICKÉ ČÍSLO ...	99
	TYP	délka	hodnota
	1-255		

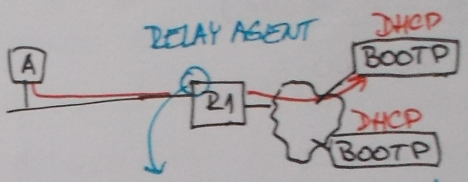
DHCP

- statické
- přidělování na neomezeno
- přidělování na urč. dobu

- stejná struktura packetu jako Bootp (geniální kompatibilita mezi bootp a dhcp)

- komunikace DHCP:





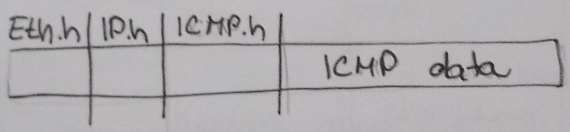
- pošlá pož. všem bootp serverům, kt. má v tabulce
- pošlá požadavek broadcastem, takže i ost. DHCP servery se dozví jaký srv si klient vybral
- klient si přidělenou IP adr. musí ověřit (ARP, ICMP)

PSI-2 18. 2.

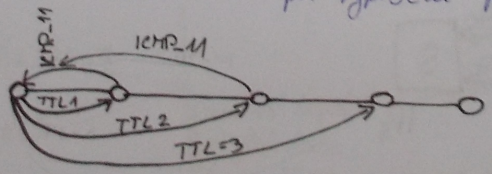
Záhlaví IP protokolu

- identification (16b): všechny fragmenty mají stejné číslo
- protocol (8b): specifikace vyšší úrovně protokolu (TCP/UDP/..)
  - např. 0x800 ⇒ IP, 0x806 ⇒ ARP } RFC ASSIGNED NUMBERS
  - TCP ⇒ 6 ; UDP ⇒ 17, ICMP ⇒ 1 ; ...
- header checksum: kontrolní součet záhlaví
  - Součet 16b bloků → jeho doplněk → takže součet hlavičky + header checksum rovná se nula

ICMP (Internet Control Message Protocol)

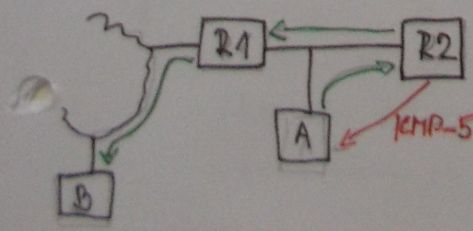


- typy zpráv: Error-reporting messages
  - 3 Destination unreachable: např. při špatně zadaném portu
  - 4 Source quench: router posílá data hostitelskému PC aby spomalil
  - 11 Time exceeded: při vypršení platnosti packetu

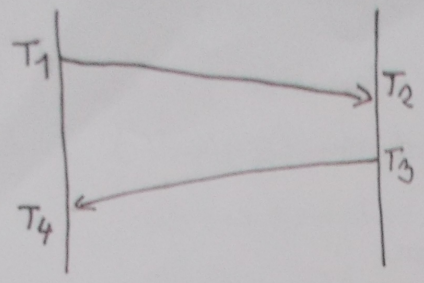


5 Redirection: modifikace směrovací tabulky

adresa, maska, brána, DNS } konfigurace uzlu



ICMP → Timestamp ⇒ Christiansonův algoritmus



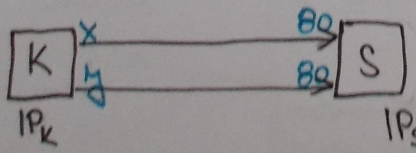
$$\Delta T = \frac{(T_2 - T_1) + (T_4 - T_3)}{2}$$

průměrné zpoždění na lince

- nastane-li chyba v ICMP chybová, tak se již neoznamuje (rovněž nelze oznamovat chybu samotného IP protokolu, pouze protok. vyšší úrovně)
- ICMP používá IP protocol, protože pouze protokol IP umí přeposlat packety přes směrovače

UDP

- porty
  - 0 unused
  - 0 – 1023 : do 511 porty síťových protokolů
    nad porty unixových protokolů
  - 1024 : unused
  - 1024 – 49 151 : uživatelské



{IPS, IPK, PortS, PortK, Protokol}

TCP

- spojovaný protokol

- formát záhlaví:

  sequence number : pořadové číslo první slabiky (v bytech)

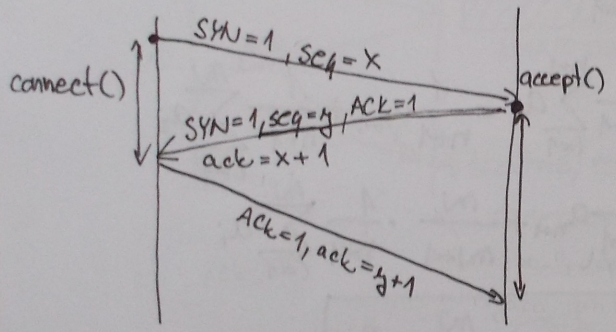
  acknowledgment nr. : očekávané číslo slabiky

  window size : ~~velikost~~ počet nepotvrzených slabik

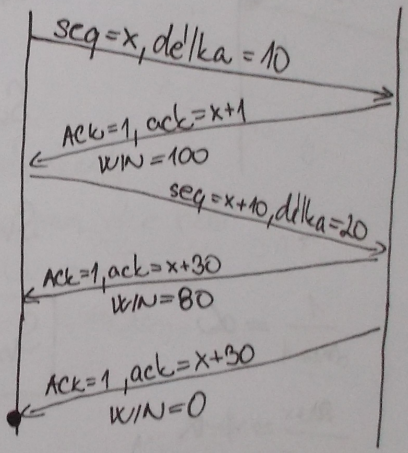
  urgent pointer : pokud je nastaven, zpracují se data předurstně

  příznaky: URG / ACK / PSH / RST / SYN / FIN

NAVÁZÁNÍ SPOJENÍ

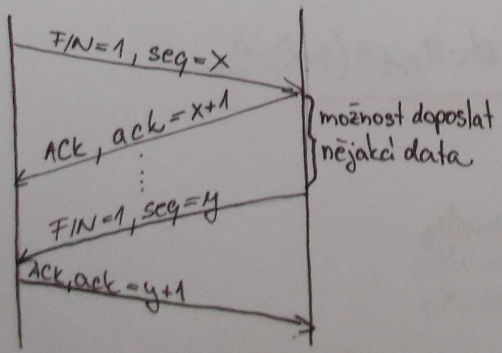


POSÍLÁNÍ DAT



přerušení přenosu

UKONČENÍ SPOJENÍ



PSH: nastaví se pokud už jsem poslal všeco jsem chtěl, tak aby příjemce nečekal na naplnění bufferu
→ oznámení že může zpracovat data z bufferu

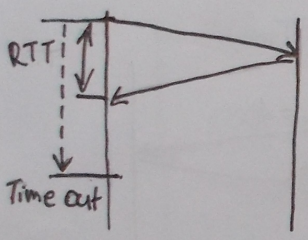
Options:

window scale factor $\Rightarrow$ window_size $\cdot 2^{WSF}$

$\rightarrow$ max = 15, protože kvůli následné při ztrátě packetu nemohl rozpoznat jestli je to nový packet nebo opak, toho původního

SACK-permitted : povolení neseřazeného příjmu

Odhad doby odezvy



$$S_n = \frac{1}{n}\sum_{i=1}^{n} a_i$$

$$S_{n+1} = \frac{1}{n+1}\sum_{i=1}^{n+1} a_i = \frac{1}{n+1}a_{n+1} + \frac{1}{n+1}\sum_{i=1}^{n} a_i$$

$$S_{n+1} = \frac{1}{n+1}a_{n+1} + \frac{n}{n+1}\cdot\frac{1}{n+1}\sum_{i=1}^{n} a_i$$

$$\frac{1}{n+1} = \alpha$$

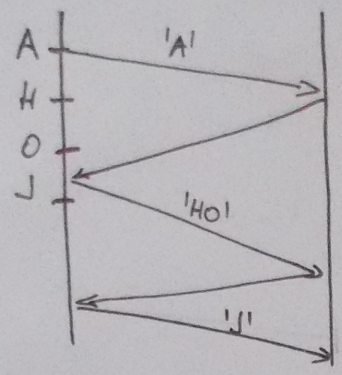
$$\frac{n}{n+1} = 1-\alpha$$

$$\boxed{S_{n+1} = \frac{1}{n+1}a_{n+1} + \frac{n}{n+1}\cdot S_n}$$

$$S_{n+1} = \alpha \cdot a_{n+1} + (1-\alpha)\cdot S_n$$

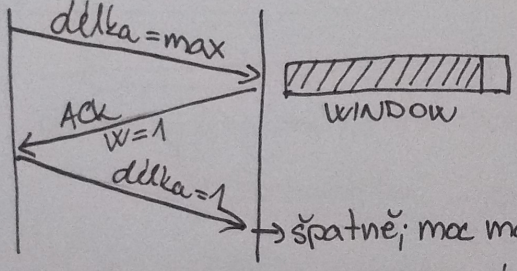
nový způsob: – z prům. hodnoty
– rozptyl
– prům. hodnoty rozptylu

- TCP optimalizuje délku zprávy ⇒ max. délka segmentu



TCP_NODELAY

SWS (Syl: Window Syndrom)

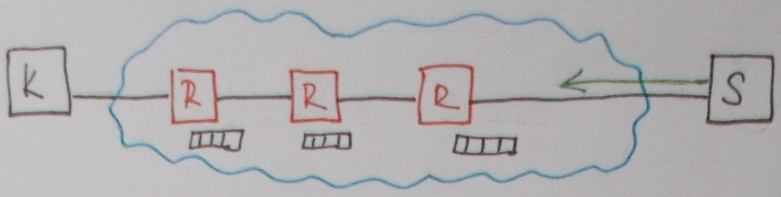


→ špatně; moc malé pakety

⇒ server, pokud má více než polovinu délka zaplněnou pošlá klientovi W=0

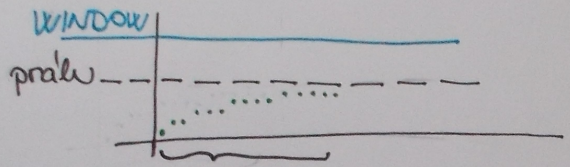
PSI-3

Řízení toku dat v síti



metody / explicitní – směrovač označí bit „hrozí zaplnění"

\ implicitní – vychází ze ztráty dat / RTT + TimeOut
\ duplicitní ACK



zkoušlivý – lineární nárůst ⇒ časově moc dlouhý

exponenciální – pošlu: 1 segment, 2, 4, ...