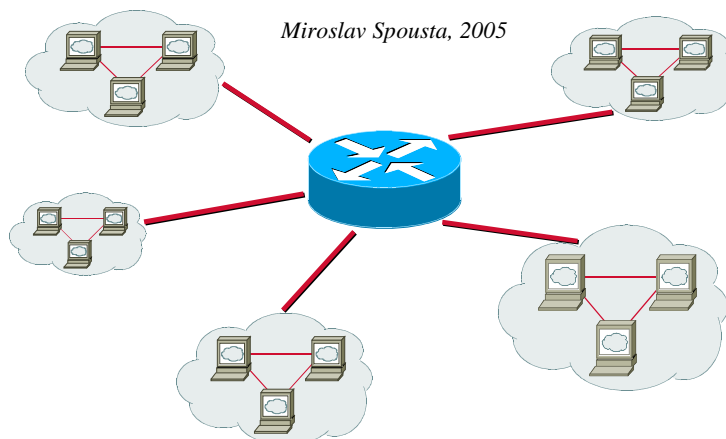


# Poítařová síť II

## 14. Transportní vrstva: TCP a UDP



1

## Transportní vrstva

- přítomná v ISO/OSI i TCP/IP
- zodpovídá za rozšíření vlastností, které požadují vyšší vrstvy (aplikace ní)
  - spojitost
  - spolehlivost
- další možná poskytování
  - řízení toku
  - bufferované vysílání a příjem
  - rozlišení mezi více adresáty v rámci uzlu
  - plně duplexní spojení
  - transparentní přenos libovolných dat
- pozorování: ne všechny aplikace vyžadují všechny tyto vlastnosti

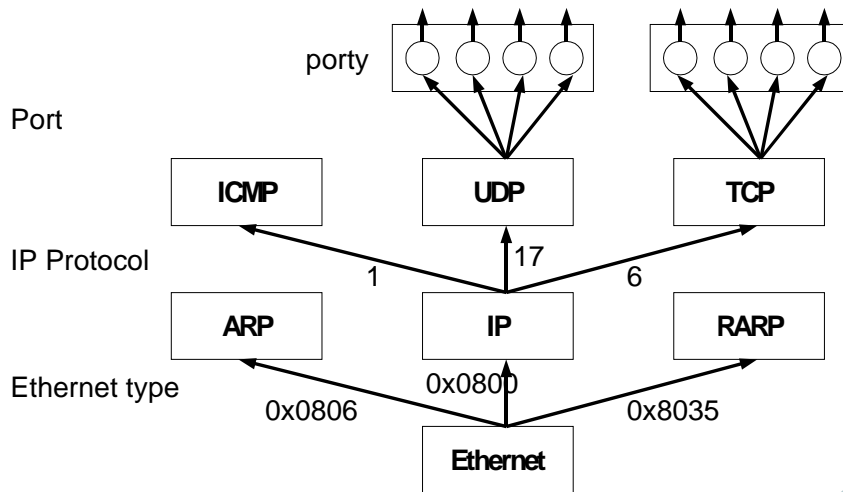
2

## TCP a UDP

- aplikace mohou využívat dva způsoby komunikace
- UDP (User Datagram Protocol)
  - má obdobné vlastnosti jako protokol IP
  - poskytuje navíc možnost adresovat více procesů v rámci jednoho uzlu (pomocí portů)
  - poskytuje kontrolní součet celému datagramu (IP nemá)
- TCP (Transmission Control Protocol)
  - podstatně rozšiřuje vlastnosti IP
  - spolehlivý (ochrání před výpadkem datagramů jejich doposláním)
  - spojovaný – před komunikací je nutné navázat spojení, také se spojení ukončuje
  - libovolně velká data pro přenos
  - řízení toku dat (ochrana proti zahlcení příjemce i sítě)
  - stejně jako UDP nabízí jemnější adresaci než IP protokol pomocí portů

3

# TCP/IP



4

# Porty

- porty (doslova p ístav) jsou reprezentovány celým íslem (1..65535)
  - je to jakási relativní adresa v rámci uzlu
- porty existují „od začátku“, nevznikají, ani nezanikají
- aplikace se p ípojují k port m, pokud chtějí komunikovat pomocí transportních protokol
  - mohou využívat více port pro r znou komunikaci
  - ale k jednomu portu patří nejvíce jedna aplikace
- n které porty jsou tzv. dobře známé (well-known)

5

# Well-known porty

- n které porty, mají definován význam
  - udávají, jaká aplikace (služba) na daném portu pracuje
  - umožní komunikaci s aplikací na vzdáleném serveru (pokud by byly porty náhodné, nevíme, kam se p ípojit)
- well-known (dobře známé) porty: mají íslo 0-1023
  - p ídluje IANA, p ívodn jako RFC, dnes <http://www.iana.org/assignments/port-numbers>
  - 21 – FTP, 22 – SSH (secure shell), 23 – telnet, 25 – SMTP (pošta), 53 – DNS
  - 80 – HTTP (WWW), 110 – POP3 (vybírání pošty), 143 – IMAP
  - 443 – HTTPS (šifrovaný web), ...
- registrované porty: mají íslo 1024 – 49151
  - IANA nep ídluje, pouze registruje pou ítí (3306 – MySQL, 1433 – MSSQL)
- ostatní porty: 49152 – 65535
  - voln k použití (nap í. pro klienty)

6

# Sockety

- *spojení* je identifikováno pomocí dvou socket
  - každý na jedné straně komunikujícího páru uzlů
- socket je dvojice IP adresa – číslo portu
  - nap. 192.168.33.1:3306
- na jednom portu tedy může probíhat nekonečně mnoho spojení
  - rozeznávají se podle adresy/portu protistrany
  - nap. web server: spojení pracuje na portu 80, ale liší se socketem protistrany (klienta)
- sockety poprvé vznikly v UNIXu (kde je všechno soubor) – pracuje se s nimi jako se soubory

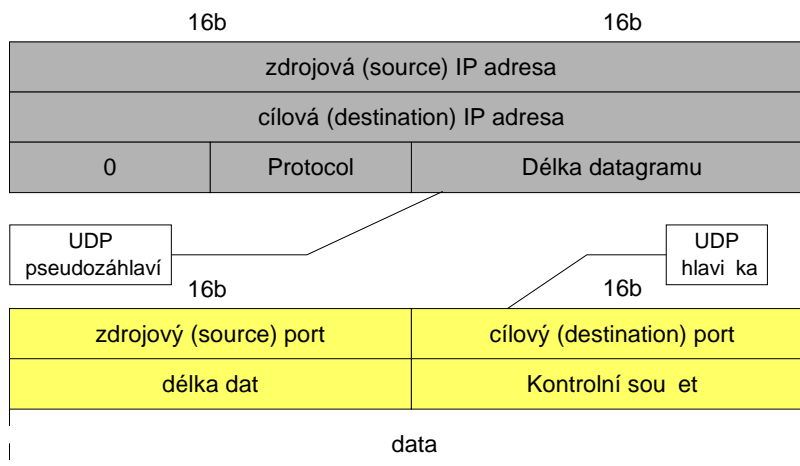
7

# UDP

- velmi jednoduchá nadstavba nad IP
- navíc poskytuje (de)multiplexování v rámci jednoho uzlu
  - neboli porty
- má malou režii (8 byte)
  - výhodné pro aplikace, které nepotřebují vlastnosti TCP
  - ale hodí se jim jednoduchost (a rychlost), bezestavovost
- může být použit pro broadcast/multicast
- také zahrnuje kontrolní součet přes celý datagram (0 – nepoužitý)
  - dokonce i přes pseudozáhlaví – data, která se nepřenášejí (jsou již obsažena v IP hlavici), ale kontrolní součet se počítá, jako by byla přítomna
  - pokud uzel přijme UDP datagram se špatným kontrolním součtem – zahodí se
- velikost datagramu musí být taková, aby se vešel do IP datagramu

8

# UDP



9

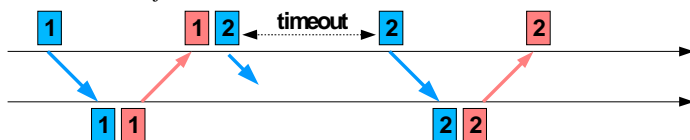
# TCP

- výbornější problém spolehlivé komunikace – nad IP
- oproti ostatním protokolům (IP, UDP) velmi složitý
- vlastnosti:
  - spojovaný charakter (3 fáze: navázání spojení, přenos dat, ukončení spojení)
  - jedná se o dvoubodovou komunikaci (vždy komunikují právě dva uzly)
  - obousměrný (plně duplexní) přenos dat
  - implementuje řízení toku
  - předchází zahlcení
  - spolehlivost (automatické přeposílání ztracených/poškozených dat)
  - v rámci vyšších protokolů se tváří jako bytová roura
  - korektní navázání a ukončení spojení (obě strany souhlasí s navázáním spojení, nedojde ke ztrátě dat při navázování ani při ukončování spojení)

10

# TCP

- spojovaná komunikace:
  - pouze na transportní vrstvě, nemožné využívat nižší vrstvy
  - neodpovídá úplně virtuálním okruhům
  - týká se pouze koncových uzlů, nikoli transportní (IP) infrastruktury
- zajištění spolehlivosti: kontinuální potvrzování a retransmise
  - generují se pozitivní potvrzení (že příjemce data obdržel)
  - odesílatel čeká na příjem potvrzení od příjemce
  - pokud ho nedostane (příjemce nedostal data, nebo se ztratilo po cestě potvrzení) do určité doby od odeslání, vyšle *segment* dat znovu
  - otázka, jak dlouho čeká?



11

# Adaptivní opakování

- jak nastavit timeout
  - malá hodnota: způsobuje zbytečné retransmise
  - velká hodnota: způsobuje přenos při výpadku
- odesílatel sleduje přenosové zpoždění datagramů, podle toho si nastaví timeout
  - sleduje průměrný round-trip (čas, za který mu půjde potvrzení)
  - timeout se dynamicky mění podle kvality linky (podle váženého průměru a rozptylu)
- timeout je něco málo nad standardní hodnotou
  - pružně reaguje – je-li velký rozptyl, zvyšuje se, není-li, blíží se standardní hodnotě
- potvrzování se vkládá do protisměrných paketů (není samostatné) – nemá velkou režii

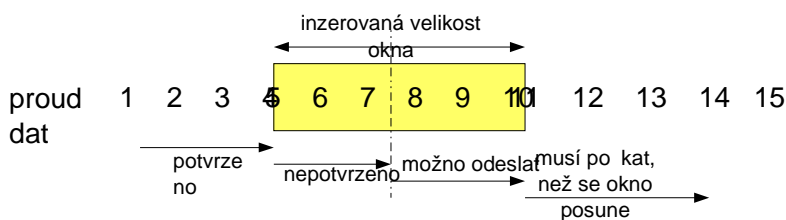
12

## ízení toku dat

- TCP se na data dívá jako na posloupnost byt
  - postupn je bufferuje, až má dostate n velký úsek (nebo po vypršení timeoutu), p edá data ke zpracování nižší vrstvě (vytvo í tzv. segment)
- aby nedošlo k zahlcení p íjemce, používá se ízení toku dat pomocí tzv. metody *klouzavého okna* (sliding window)
  - prot íšek inzeruje v každém poslaném segmentu dat, kolik má místa v bufferech
  - pokud inzeruje 0, m í by uzel p estat vysílat
- je pot eba zabránit, aby p íjemce inzeroval p íliš malé okno (bajty)
  - p íliš by vzrostla režie
  - potvrzení se posílá až když je zpracováno ur íté množství dat (zpoř ování potvrzení, max. 500ms)
- stejn tak není žádoucí, aby odesílatel posílal malé segmenty
  - op t kv íli režii

13

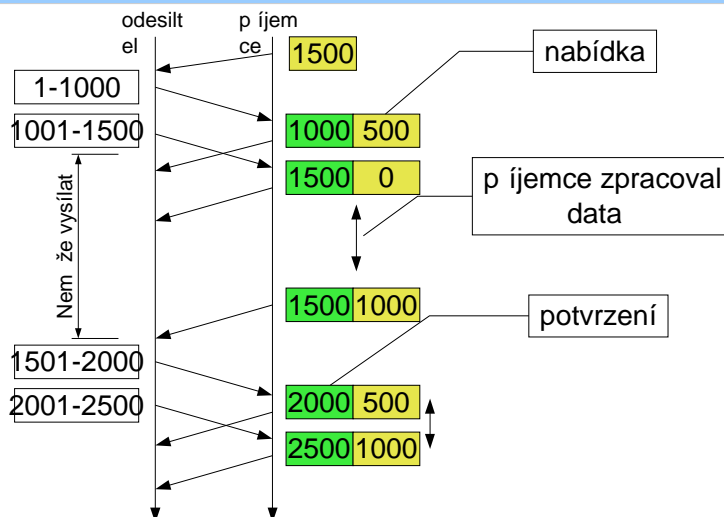
## Sliding window



- v okn jsou nepotvrzená data
- p íjetím potvrzení se okno posune doprava
- pokud nemáme potvrzená data a vy erpali jsme okno, musíme počkat
  - na potvrzení, p ípadn (pokud vyprší asova ) na retransmisi

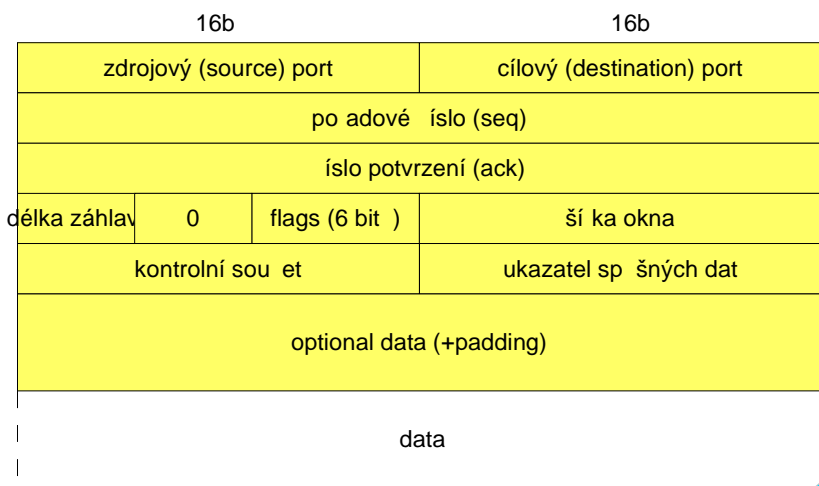
14

## ízení toku



15

# TCP



16

## Popis TCP segmentu

- porty: 0 – 65535 (jsou v prvních 64 bitech (kvalitně ICMP))
- po adové číslo: sequence number – označuje, kolik byte bylo odesláno
  - na počátku spojení je to náhodné číslo
  - kvalitně bezpečnosti (aby se nedalo odhadnout)
- číslo potvrzení – udává, že všechny předcházející byte jsou potvrzeny
  - je to číslo, které protistrana očekává jako následující
- délka záhlaví – v násobcích 32 bit (4B)
- šířka okna: počet byte, které je možné ještě vyslat bez předchozího potvrzení
  - odpovídá velikosti vyrovnávací paměti protistrany
- kontrolní součet – zabezpečení přes celý segment (včetně pseudozáhlaví)
- ukazatel spěšných dat – poslední byte urgentních dat
- volitelné možnosti – např. MSS

17

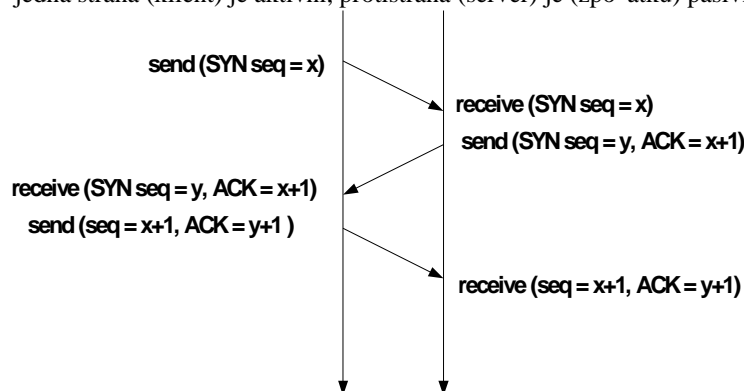
## Priznaky TCP segmentu

- bitové pole (6b)
- URG – spěšná (urgentní data) v segmentu
- ACK – udává, že pole potvrzení je platné
- PSH – požaduje okamžité doručení segmentu vyšší vrstvy (pro interaktivní aplikace, ...)
- RST – požaduje okamžité ukončení spojení
- SYN – žádost o navázání spojení (segment neobsahuje data)
- FIN – žádost o přerušení spojení

18

## Navázání spojení

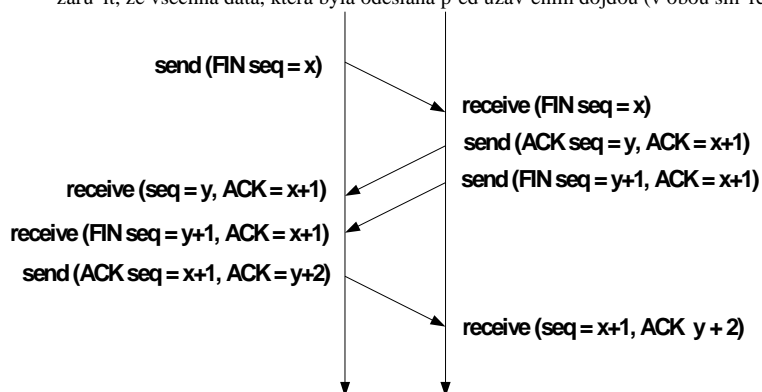
- musí fungovat spolehlivě i v případě ztráty některého IP datagramu
- jedna strana (klient) je aktivní, protistrana (server) je (zpočátku) pasivní



19

## Ukončení spojení

- stejně jako navázání musí fungovat spolehlivě
  - i v případě ztráty některého IP datagramu
  - zaručit, že všechna data, která byla odeslána před uzavřením dojdou (v obou směrech!)



20