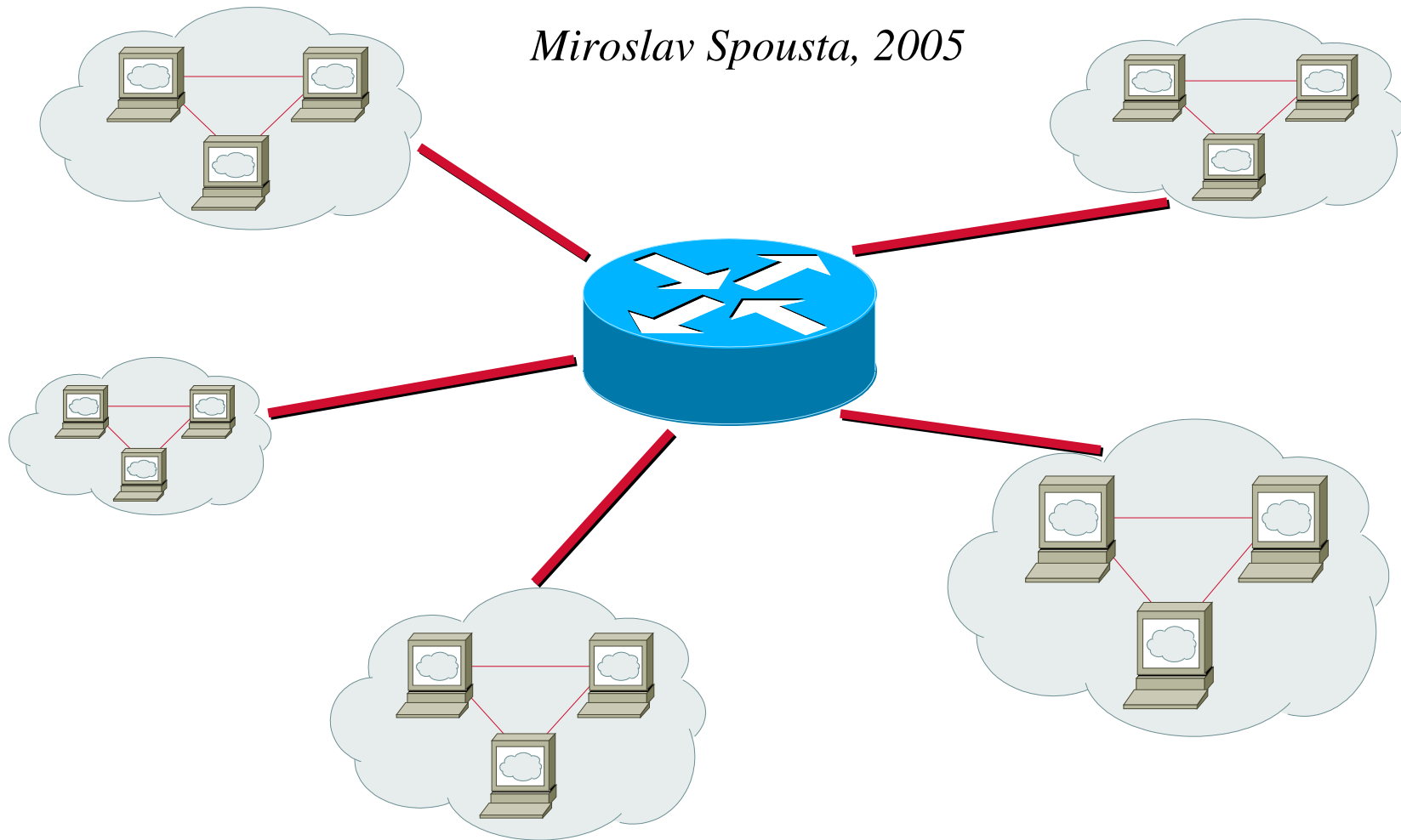


Počítačové sítě II

14. Transportní vrstva: TCP a UDP

Miroslav Spousta, 2005



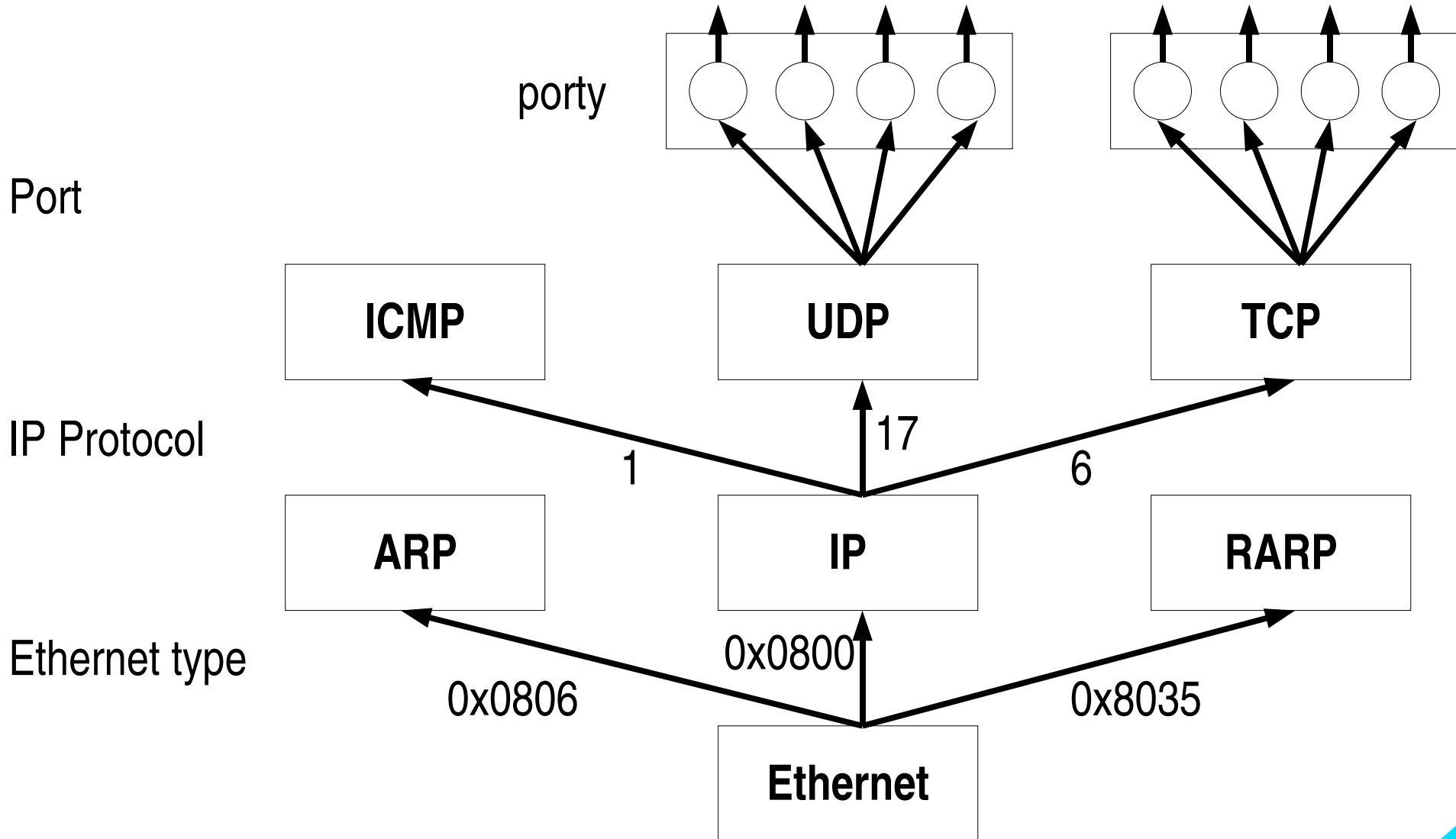
Transportní vrstva

- přítomná v ISO/OSI i TCP/IP
- zodpovědná za rozšíření vlastností, které požadují vyšší vrstvy (aplikační)
 - spojovanost
 - spolehlivost
- další možná přizpůsobení
 - 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

TCP a UDP

- aplikace mohou využívat dva způsoby komunikace
- UDP (User Datagram Protocol)
 - má obdobné vlastnosti jako protokol IP
 - přidává navíc možnost adresovat více procesů v rámci jednoho uzlu (pomocí *portů*)
 - přidává kontrolní součet k celému datagramu (IP nemá!)
- TCP (Transmission Control Protocol)
 - podstatně rozšiřuje vlastnosti IP
 - spolehlivý (řeší výpadek datagramů jejich přeposláním)
 - 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

TCP/IP



Porty

- porty (doslova přístav) jsou reprezentovány celým číslem (1..65535)
 - je to jakási relativní adresa v rámci uzlu
- porty existují „od začátku“, nevznikají, ani nezanikají
- aplikace se připojují k portům, pokud chtějí komunikovat pomocí transportních protokolů
 - mohou využívat více portů pro různou komunikaci
 - ale k jednomu portu patří nejvíce jedna aplikace
- některé porty jsou tzv. dobře známé (well-known)

Well-known porty

- některé porty, mají definován význam
 - udávají, jaká aplikace (služba) na daném portu pracuje
 - umožňují komunikaci s aplikací na vzdáleném serveru (pokud by byly porty náhodné, nevíme, kam se připojit)
- well-known (dobře známé) porty: mají čísla 0-1023
 - přiděluje IANA, původně 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 čísla 1024 – 49151
 - IANA nepřiděluje, pouze registruje použití (3306 – MySQL, 1433 – MSSQL)
- ostatní porty: 49152 – 65535
 - volně k použití (např. pro klienty)

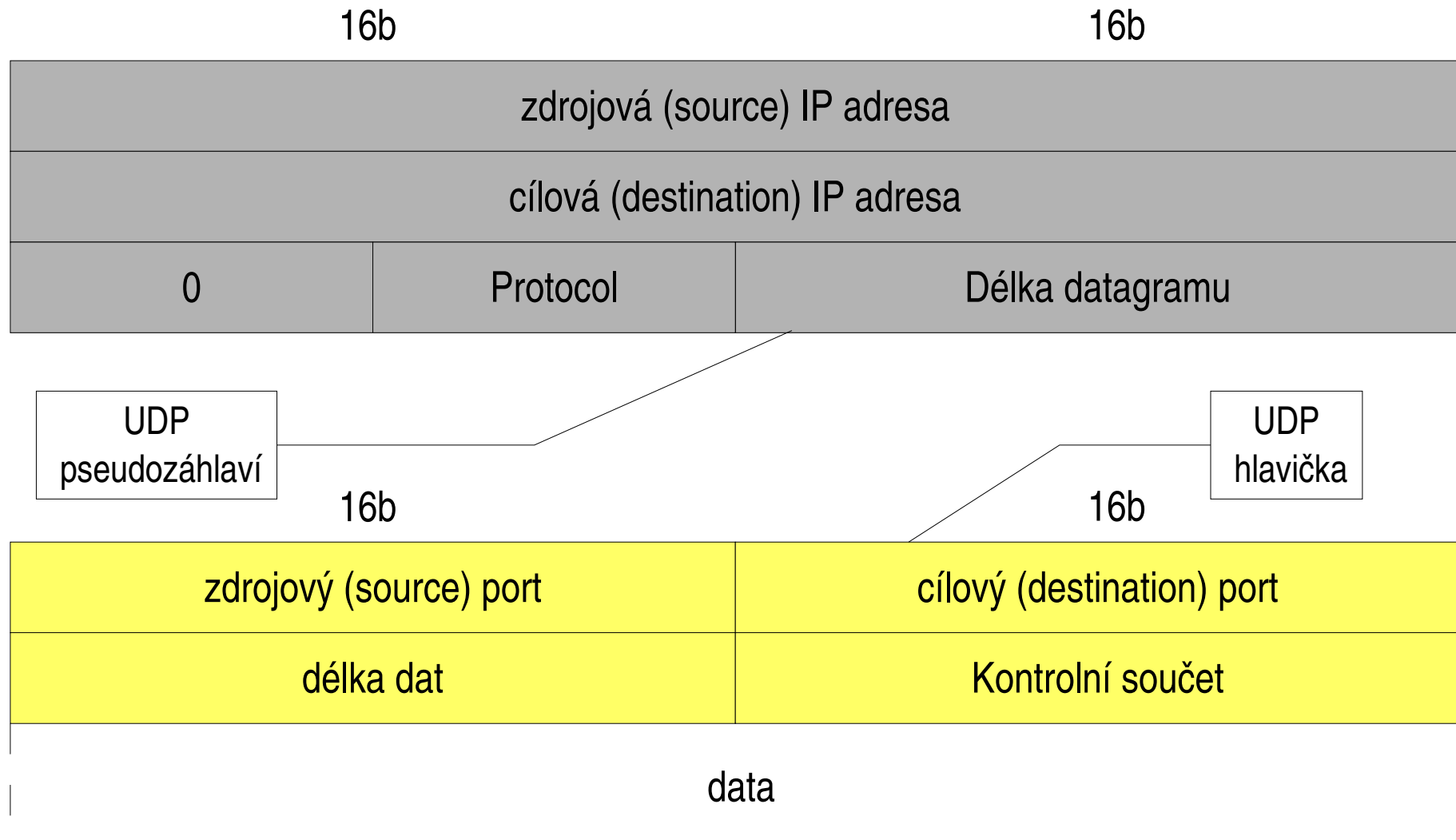
Sockety

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

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 nepřenáší (jsou již obsažena v IP hlavičce), 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

UDP

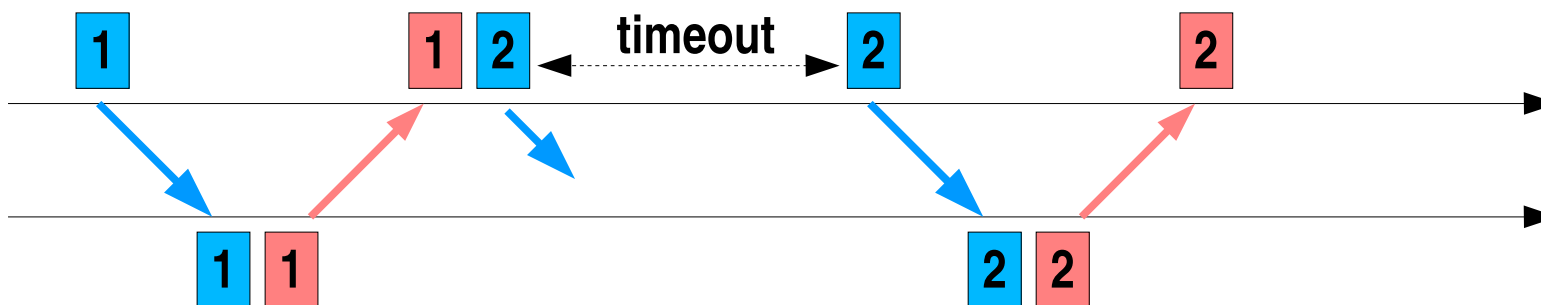


TCP

- výborně řeší 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 komunikace (vždy komunikují právě dva uzly)
 - obousměrný (plně duplexní) přenos dat
 - implementuje řízení toku
 - předchází zahlcení
 - spolehlivost (automatické přeposílání ztracených/poškozených dat)
 - vůči vyšším protokolům se tváří jako bytová roura
 - korektní navázání a ukončení spojení (obě strany souhlasí s navázáním spojení, nedojde ke ztrátě dat při navazování ani při ukončování spojení)

TCP

- spojovaná komunikace:
 - pouze na transportní vrstvě, neumožňuje 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 ztarilo po cestě potvrzení) do určité doby od odeslání, vyšle *segment* dat znovu
 - otázka, jak dlouho čekat?



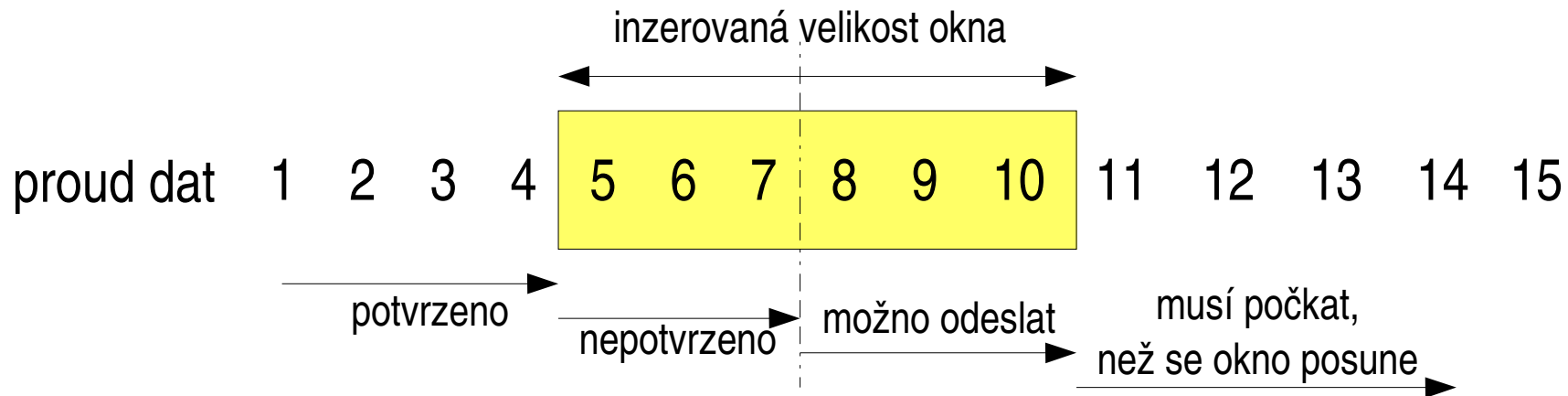
Adaptivní opakování

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

Řízení toku dat

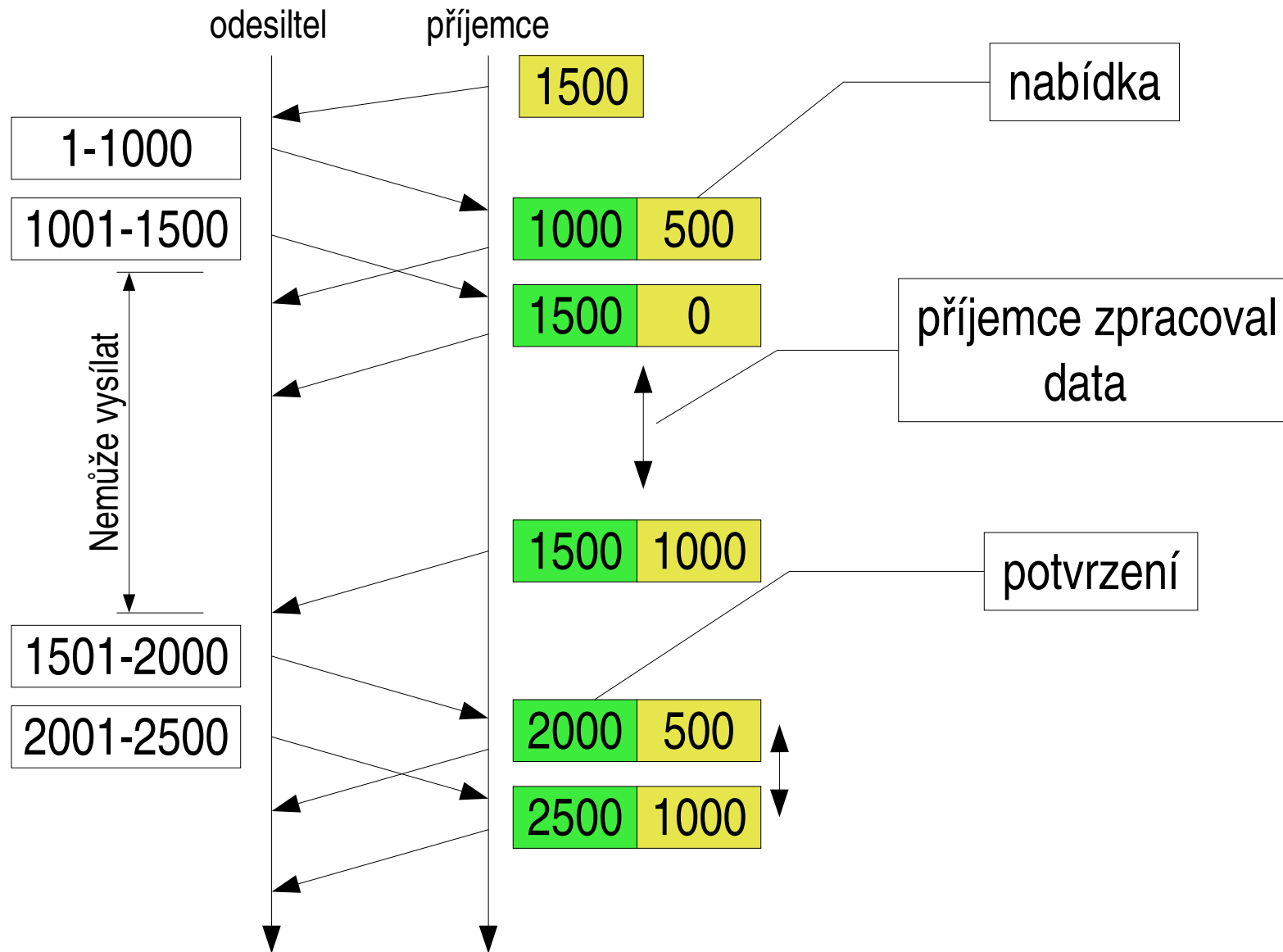
- TCP se na data dívá jako na posloupnost bytů
 - postupně je bufferuje, až má dostatečně velký úsek (nebo po vypršení timeoutu), předá data ke zpracování nižší vrstvě (vytvoří tzv. segment)
- aby nedošlo k zahlcení příjemce, používá se řízení toku dat pomocí tzv. metody *klouzavého okna* (sliding window)
 - protějšek inzeruje v každém poslaném segmentu dat, kolik má místa v bufferech
 - pokud inzeruje 0, měl by uzel přestat vysílat
- je potřeba zabránit, aby příjemce inzeroval příliš malé okno (bajty)
 - příliš by vzrostla rezie
 - potvrzení se posílá až když je zpracováno určité množství dat (zpoždování potvrzení, max. 500ms)
- stejně tak není žádoucí, aby odesílatel posílal malé segmenty
 - opět kvůli reži

Sliding window



- v okně jsou nepotvrzená data
- přijetím potvrzení se okno posune doprava
- pokud nemáme potvrzená data a vyčerpali jsme okno, musíme počkat
 - na potvrzení, případně (pokud vyprší časovač) na retransmisi

Řízení toku



TCP

16b

16b

zdrojový (source) port		cílový (destination) port	
pořadové číslo (seq)			
číslo potvrzení (ack)			
délka záhlaví	0	flags (6 bitů)	šířka okna
kontrolní součet		ukazatel spěšných dat	
optional data (+padding)			
data			

Popis TCP segmentu

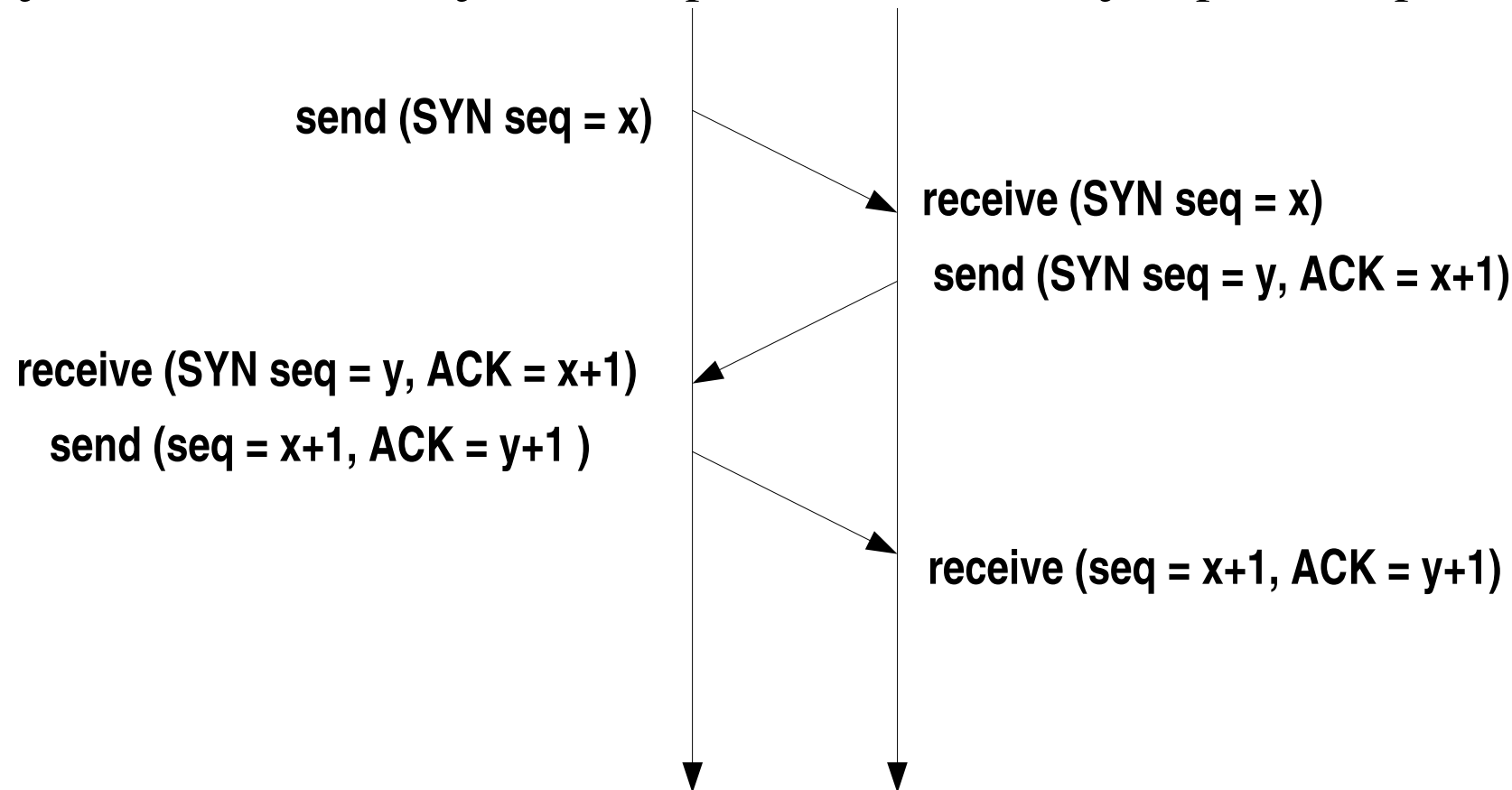
- porty: 0 – 65535 (jsou v prvních 64 bitech (kvůli ICMP))
- pořadové číslo: sequence number – označuje, kolik byte bylo odesláno
 - na počátku spojení je to náhodné číslo
 - kvůli bezpečnosti (aby se nedalo odhadnout)
- číslo potvrzení – udává, že všechny předcházející byte jsou potvrzeny
 - je to číslo, které protistrana očekává jako následující
- délka záhlaví – v násobcích 32 bitů (4B)
- šířka okna: počet byte, které je možné ještě vyslat bez příchodu 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

Příznaky TCP segmentu

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

Navázání spojení

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



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!)

