KOMUNIKAČNÍ TECHNOLOGIE (BPC-KOM)

Ústav telekomunikací
Fakulta elektrotechniky a komunikačních technologií
VUT v Brně

doc. Ing. Jan Jeřábek, Ph.D.
jerabekj@feec.vutbr.cz

TRANSPORTNÍ VRSTVA PŘENOSOVÝCH SYSTÉMŮ

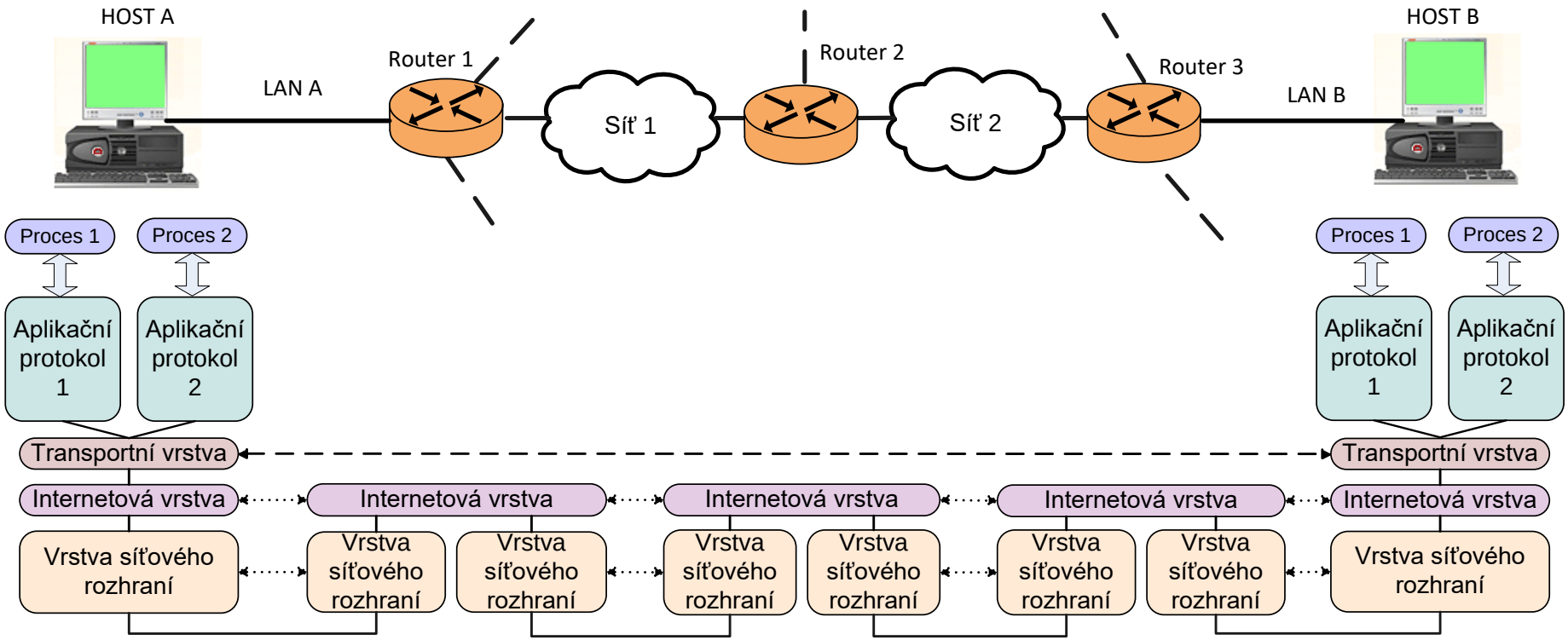
Plán přednášky

- Služby transportní vrstvy
  - Komunikace procesů
  - Adresování na transportní vrstvě
  - Zapouzdřování dat
  - Multiplexování
  - Řízení přenosu
  - Charakter poskytovaných služeb
  - NAT + PAT
- UDP
  - Datagram
  - Služby
  - Využití
- TCP
  - Služby, vlastnosti
  - Segment
  - Práce se spojením a průběh komunikace, velikost okna
  - Využití
- QUIC

SLUŽBY TRANSPORTNÍ VRSTVY

Komunikace procesů

- transportní vrstva nad síťovou
- koncový charakter (komunikace procesů)
  - rozlišení procesů
  - možno více komunikačních okruhů mezi dvěma uzly



Adresování na transportní vrstvě

- Organizace komunikace klient-server
  - klientský proces, serverovský proces
- Čtyři adresy
  - lokální host + lokální proces (= socket 1)
  - vzdálený host + vzdálený proces (= socket 2)
- Transportní adresy (UDP i TCP)
  - čísla portů (16 bit číslo)
  - zdrojový + cílový port

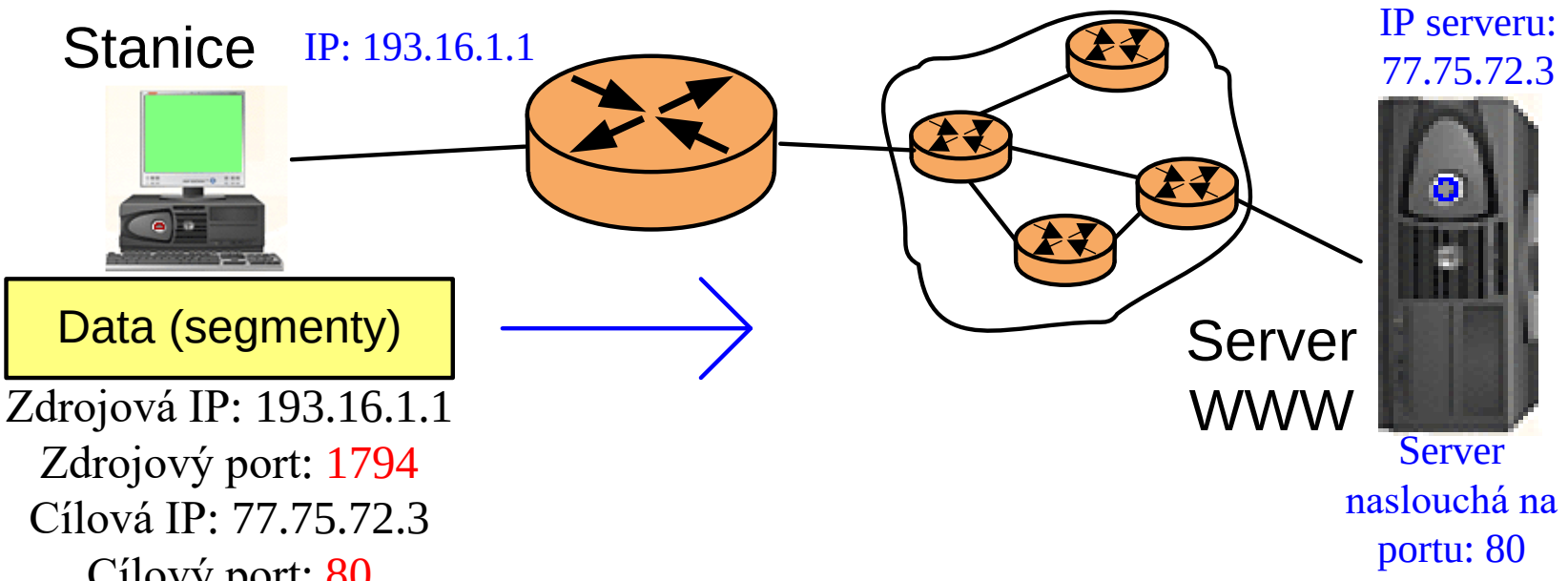
Základní dělení portů

Rozsah čísel portů	Označení portů	Využití
0 – 1023	Známé (*well-known*)	Vyhrazeno pro dobře známé aplikace, číslo portu zpravidla na straně serveru
1024 – 49151	Registrované (*registered*)	Pro méně používané aplikace nebo pro porty na straně klienta při komunikaci; jejich použití je registrováno u organizace IANA
49152 – 65535	Soukromé a dynamické (*private and dynamic*)	Dynamicky přiřazované čísla portů na straně klientské aplikace

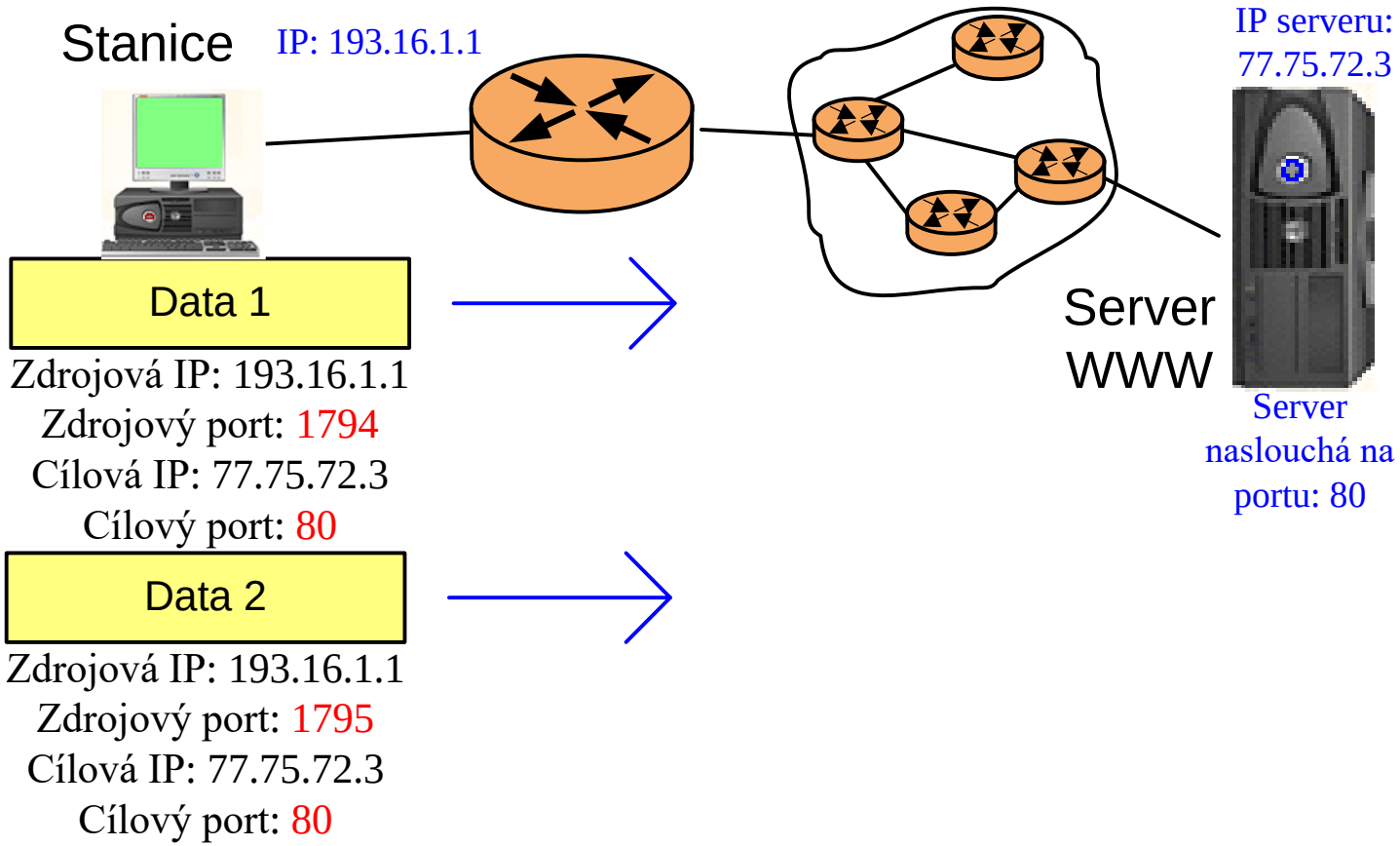
Významné well-known porty

Číslo portu	Transportní protokol	Aplikační protokol
20	tcp	ftp – data
21	tcp	ftp – řízení
23	tcp	telnet
25	tcp/udp	smtp
53	tcp/udp	dns
67	udp	dhcp server
68	udp	dhcp klient
80	tcp/udp	http
443	tcp/udp	https

Ukázka komunikace webový prohlížeč – webový server (jedno spojení)

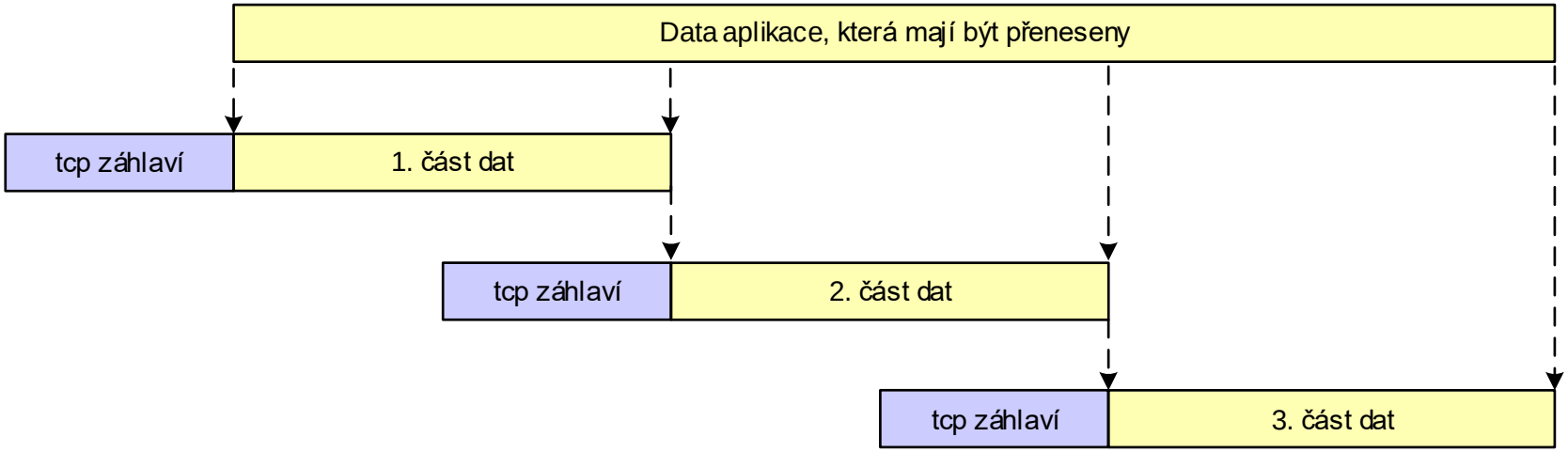


Ukázka komunikace webový prohlížeč – webový server (dvě spojení)



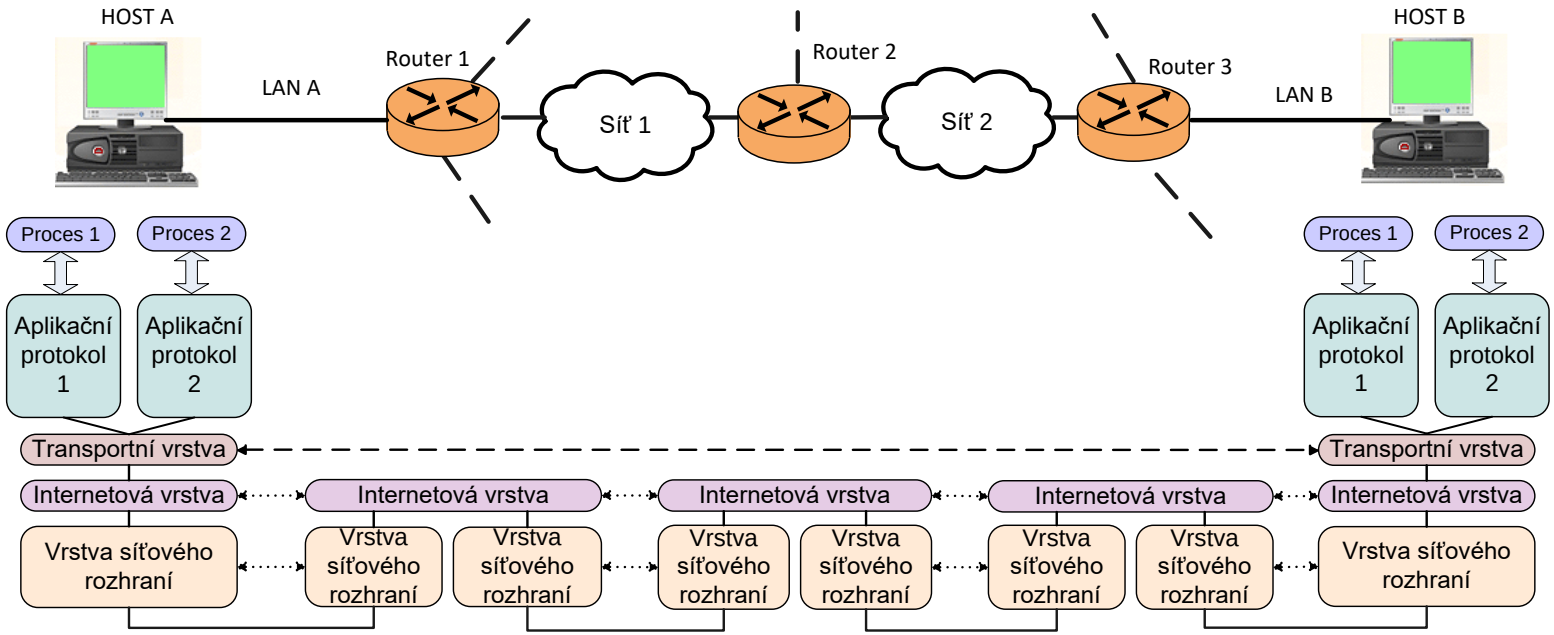
Zapouzdřování dat

- segmentace
  - velké množství dat aplikace nutno rozdělit
  - vzniká segment (TCP) či datagram (UDP)
  - nutné přidání záhlaví (zapouzdření x odpouzdření)
  - nutná informace o socketech, popř. další informace
  - odlišný přístup k číslování jednotek u TCP a UDP, UDP lepší k segmentaci nepoužívat
  - další zapouzdřování na síťové vrstvě, možná i fragmentace



Multiplexování a demultiplexování v transportní vrstvě

- multiplexování
  - střet požadavků z různých zdrojů v jednom bodě
  - opakem demultiplexování
  - řazení požadavků do fronty



Řízení přenosu v transportní vrstvě

- standardní součástí především
  - **řízení toku dat** (*flow control*)
    - způsob organizace komunikace mezi koncovými body, realizaci front a vyrovnávacích pamětí
    - posuvné okno
  - **řízení chybových stavů** (*error control*)
    - číslování přenášených jednotek či dat a potvrzování jejich úspěšného přenosu
    - posuvné okno a další mechanizmy
  - **předcházení zahlcení** (*congestion control*)
    - posuvné okno
    - nastavení dalších parametrů, např. pravidel pro opakovaný přenos či potvrzování přenosů

Řízení přenosu v transportní vrstvě

- techniky se vyskytují i na síťové či spojové vrstvě
- techniky řízení spojové vrstvy
  - dílčí segmenty trasy
  - ne vždy k dispozici
  - ne koncový charakter řízení
- techniky řízení síťové vrstvy
  - částečně koncový charakter
  - omezené prostředky (ICMP či ICMPv6)
- techniky řízení transportní vrstvy
  - koncový charakter
  - klíčové pro komunikaci

Charakter poskytovaných služeb

- **bez spojení** (*connectionless*)
  - aplikace potřebuje pouze rozdělit data do bloků přiměřené velikosti
  - vyžadováno pouze sekvenční odesílání jednotek
  - může dojít ke změně pořadí či ztrátám
  - není možné implementovat mechanizmy řízení toku, řízení chybových stavů či předcházet zahlcení
  - existují však aplikace, kterým tento způsob postačuje
  - výhoda – malá režie komunikace
  - typickým zástupcem protokol UDP

Charakter poskytovaných služeb

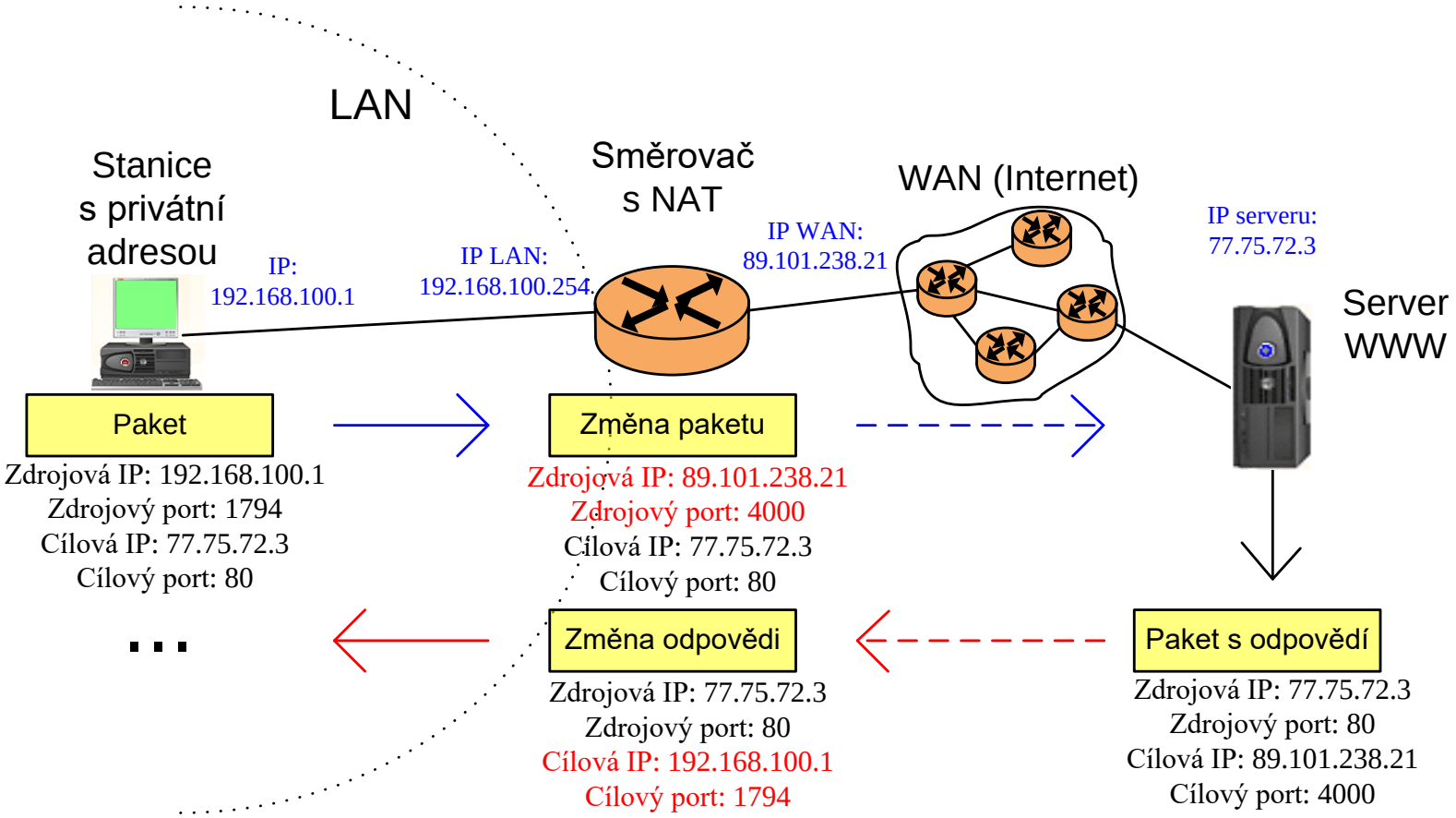
- **se spojením** (*connection-oriented*)
  - koncové strany komunikace před vlastním přenosem navazují spojení
  - přenos dat pouze po navázání spojení
  - potvrzování úspěšnosti přenosu či opakovaný přenos v případě chyb, úprava rychlosti
  - po provedení přenosu je spojení ukončeno
  - služba se spojením na transportní × síťové vrstvě
    - transportní vrstva se nezabývá fyzickými trasami paketů v síti
    - služba se spojením na síťové vrstvě vyžaduje spolupráci směrovačů k vytvoření (virtuální) přenosové trasy
    - na transportní vrstvě se zabýváme pouze koncovým charakterem komunikace
    - Pozn.: nad síťovou vrstvou bez spojení může vzniknout transportní služba se spojením
  - typickým příkladem je TCP

Network and Port Address Translation

- NAT
  - primárně technikou síťové vrstvy
  - překlad IP adres (veřejná × privátní)
  - dobře použitelné při překladu 1:1
- Více stanic na síti, nutnost odlišení
  - **překlad n:n**
    - počet privátních adres odpovídá počtu veřejných
    - v praxi málo časté (vysoký počet veřejných adres)
  - **překlad se záměnou adres transportní úrovně**
    - směrovač zaměňuje všechny vnitřní IP adresy na nižší počet veřejných (typicky jednu)
    - aby byl schopen rozlišit provozy jednotlivých stanic, zasahuje i do transportních adres (každá stanice má rezervovány nějaká čísla portů)
    - v praxi běžné, úspora adresního prostoru

Network and Port Address Translation

- Network and Port Address Translation (NPAT) × Network Address Port Translation (NAPT) × NAT



USER DATAGRAM PROTOCOL (UDP)

Úvod do protokolu UDP

- jednoduchý transportní protokol
- nespojovaný a nespolehlivý přenos (*best effort*)
- jednotky UDP = datagramy
- navíc oproti IP vrstvě
  - přenos mezi konkrétními procesy (transportní adresy)
- hlavní vlastnosti
  - jednoduchost, minimální režie a zpoždění
- vhodný pro krátké zprávy (ne kritické důležitosti)
  - systém dotaz – odpověď (pouze dva datagramy) x efektivita u TCP
- záhlaví pouze 8 B

Bity 0-15	16-31
Zdrojový port	**Cílový port**
Celková délka	**Kontrolní součet**
Data aplikace	

Datagram protokolu UDP

- **Zdrojový port** (*source port*)
  - port na straně odesílatele datagramu
  - odesilatel klientem – port vybrán z příslušného rozsahu
  - odesilatel server – číslo portu zpravidla dáno dle typu služby
- **Cílový port** (*destination port*)
  - port na straně příjemce datagramu
  - zpravidla není shodné se zdrojovým
  - vychází především z toho, zda je odesilatel klient či server
- **Celková délka** (*total length*)
  - délka celého datagramu včetně záhlaví, v bajtech
- **Kontrolní součet** (*checksum*)
  - k detekci základních chyb na transportní úrovni
  - ve srovnání s mechanizmy řízení chyb protokolu TCP zanedbatelné
  - počítán z
    - UDP záhlaví
    - datové části
    - části IP záhlaví paketu (tzv. pseudozáhlaví)

Služby protokolu UDP

- **Komunikace proces-proces**
  - pomocí portů
- **Přenos dat bez spojení**
  - každý datagram přenášen jako samostatná jednotka
  - datagramy nejsou číslovány
  - neprobíhá žádné navazování spojení či testování dostupnosti adresáta
- **Žádné řízení toku dat, řízení proti zahlcení či řízení chybových stavů**
  - vysílač UDP datagramů může potenciálně zahltit příjemce či síť
  - v rámci UDP protokolu neexistují mechanizmy na řešení těchto problémů
  - kromě kontrolního součtu žádné mechanizmy řízení chyb, chybových stavů či řízení přenosu jednotek
- **Zapouzdřování a odpouzdřování dat**
  - služba vytváření jednotek transportní úrovně na straně vysílače
  - následně oddělení záhlaví na straně příjemce
- **Frontování, multiplexování a demultiplexování**
  - vstupní a výstupní fronty odděleně pro jednotlivé aplikace, dle portů

Příklady využití protokolu UDP

- jednoduchý a rychlý protokol pro spoustu aplikací dostatečný
  - malé zatížení tras, prvků, aplikací
- Klasicky: systémy dotaz – odpověď
  - Domain Name System (DNS)
    - řazen do aplikační vrstvy
    - krátké dotazy na IP adresy na základně jmenných názvů
    - ztráta či chyba přenosu řešena opakovaným dotazem
    - Umí použít i TCP !
  - Voice over IP (VoIP)
    - malé ztráty méně kritické než velké zpoždění
    - problém pořadí datagramů musí být řešen
- Současné použití výrazně širší (web, HTTP/3)

TRANSMISSION CONTROL PROTOCOL (TCP)

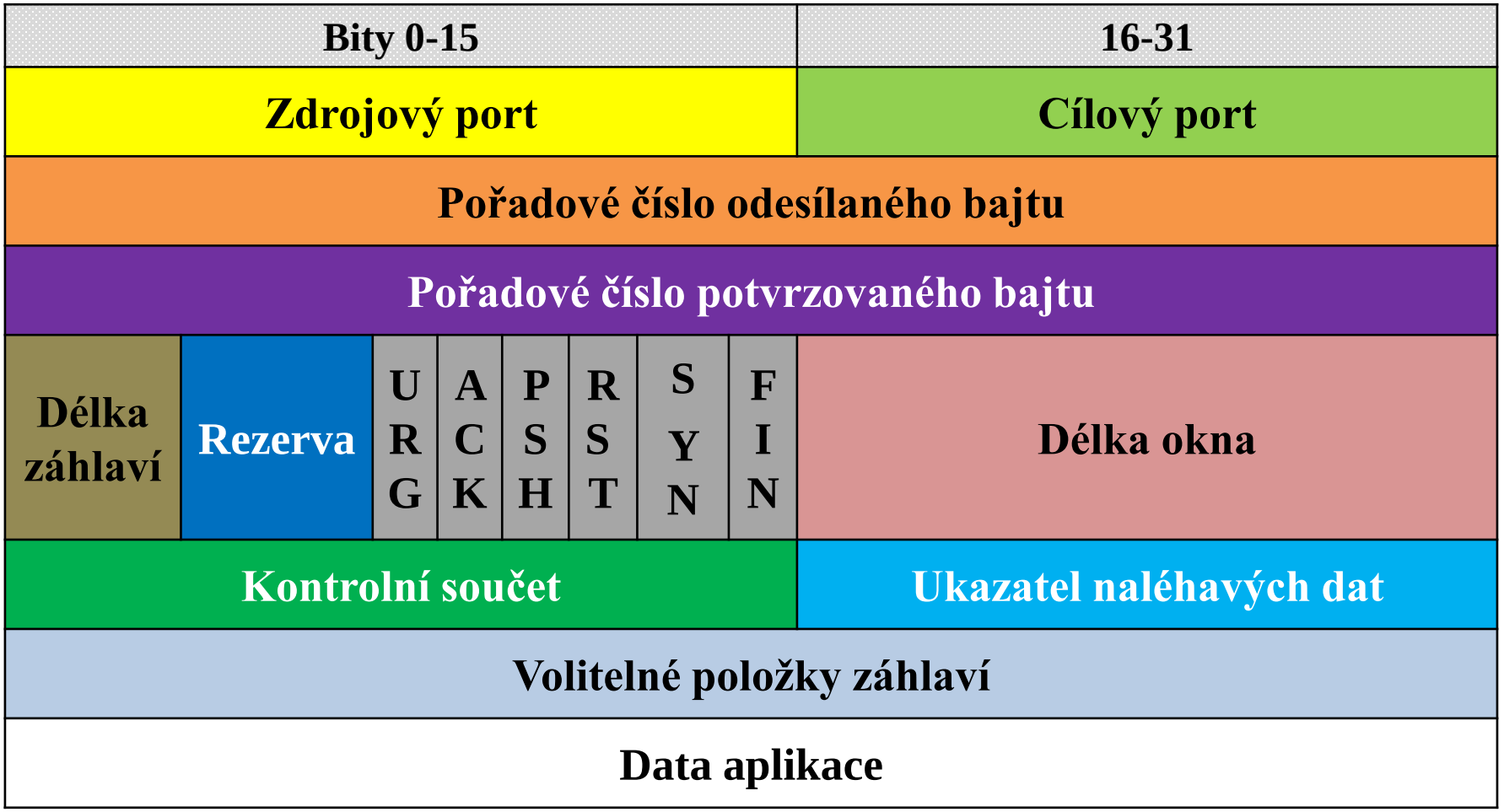
Služby protokolu TCP

- **Komunikace proces-proces**
  - stejně jako UDP
- **Přenos proudu dat**
  - odlišný koncept od UDP
  - TCP vytváří dojem propojení komunikujících procesů okruhem, kterým je možné přenášet proud bajtů
  - pro přenos síťovou vrstvou vytváří segmenty
  - přenášené bajty jsou určitým způsobem číslovány (správné seskládání dat)
- **Plně duplexní přenos dat**
  - strany komunikují oběma směry zároveň
- **Multiplexování a demultiplexování**
  - stejně jako UDP protokol
- **Spojově orientovaná služba**
  - navázání spojení před přenosem dat, ukončení spojení po přenosu
  - Zjištění dostupnosti a ochoty komunikovat u druhé strany, nastavení spojení
- **Spolehlivý přenos dat**
  - TCP používá potvrzovací mechanizmy, které umožňují ověřit, že došlo k úspěšnému přenosu

Vlastnosti protokolu TCP

- Vlastnosti umožňující poskytování služeb, odlišné od UDP
  - **Číslovací systém**
    - založen na číslování odesílaných a potvrzovaných bajtů
    - nejsou číslovány segmenty jako celky
    - komunikace je obousměrná, celkem čtvero číslování
      - odeslané bajty jedné strany
      - odeslané bajty druhé strany
      - bajty potvrzované jednou stranou
      - bajty potvrzované druhou stranou
  - **Řízení toku dat**
    - především práce s velikostí okna
  - **Řízení chybových stavů**
    - mechanizmy sledování chyb a řízení způsobů reakce na tyto chyby
  - **Řízení stavů zahlcení**
    - pružná reakce na zahlcení na straně příjemce či v síti
    - podstatou možnost regulovat množství a rychlost odesílaných dat

Segment protokolu TCP



Segment protokolu TCP

- **Zdrojový port** (*source port*)
  - port na straně odesílatele segmentu, obdobně jako u UDP
- **Cílový port** (*destination port*)
  - port na straně příjemce segmentu, opět obdobně jako u UDP
- **Pořadové číslo odesílaného bajtu** (*sequence number* – **SEQ**)
  - pole číslování odesílaných bajtů
  - pole obsahuje pořadové číslo prvního z odesílaných bajtů v daném segmentu
- **Pořadové číslo potvrzovaného bajtu** (*acknowledgment number* – **ACK**)
  - komunikace probíhá obousměrně, potvrzení dříve přijatých dat od protistrany
  - hodnota dalšího očekávaného bajtu dle číslování bajtů protistrany
- **Délka záhlaví** (*header length*)
  - délka celého záhlaví
  - musí být uvedeno kvůli poli Volitelné položky záhlaví (proměnná délka)

Segment protokolu TCP

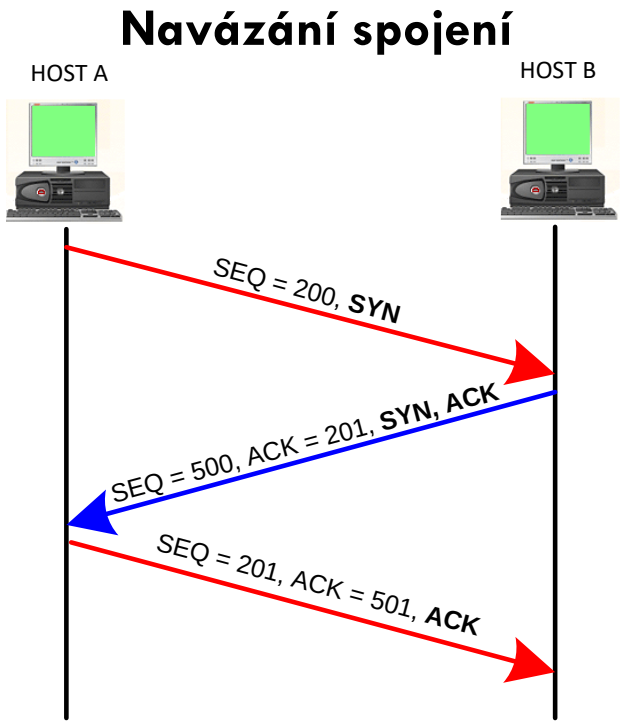
- **Příznakové bity** (*flags*)
  - mohou být různě kombinovány k dosažení funkcí řízení toku, navázání či ukončení spojení
  - Význam jejich nastavení na „1“ je
    - **URG** (*urgent*) – segment nese naléhavá data
    - **ACK** (*acknowledgment*) – indikuje, že hodnota uvedená v poli potvrzovaného bajtu je platná
    - **PSH** (*push function*) – signalizuje, že data mají být ihned po přijetí předána aplikaci a nemá se čekat na přijetí dalších segmentů
    - **RST** (*reset the connection*) – pro řešení situace s duplikáty navazovacích segmentů, k odmítnutí spojení
    - **SYN** (*synchronize sequence numbers*) – odesílatel začíná novou sekvenci číslování bajtů, využíváno při navazování spojení
    - **FIN** (*terminate the connection*) – odesílatel ukončil přenos dat, využíváno při uzavírání spojení

Segment protokolu TCP

- **Délka okna** (*window size*)
  - vyjadřuje maximální počet bajtů, které může vysílač odeslat, aniž by čekal na potvrzení od přijímače
  - může se podle potřeby měnit
- **Kontrolní součet** (*TCP checksum*)
  - obdobné jako UDP kontrolní součet
- **Ukazatel naléhavých dat** (*urgent pointer*)
  - pole vyplněno jen když je příznakový bit URG nastaven na „1“
- **Volitelné položky záhlaví** (*options*)
  - pole nemusí být přítomno vůbec, nad rámec tohoto kurzu

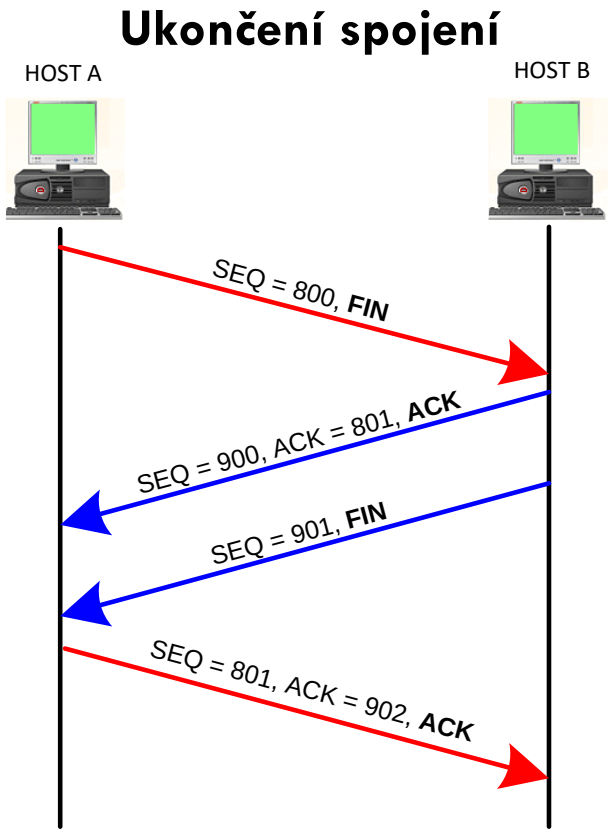
Navazování a ukončování spojení u protokolu TCP

- TCP vytváří virtuální okruh mezi procesy



Three-way handshake

[SYN] > [SYN, ACK] > [ACK]



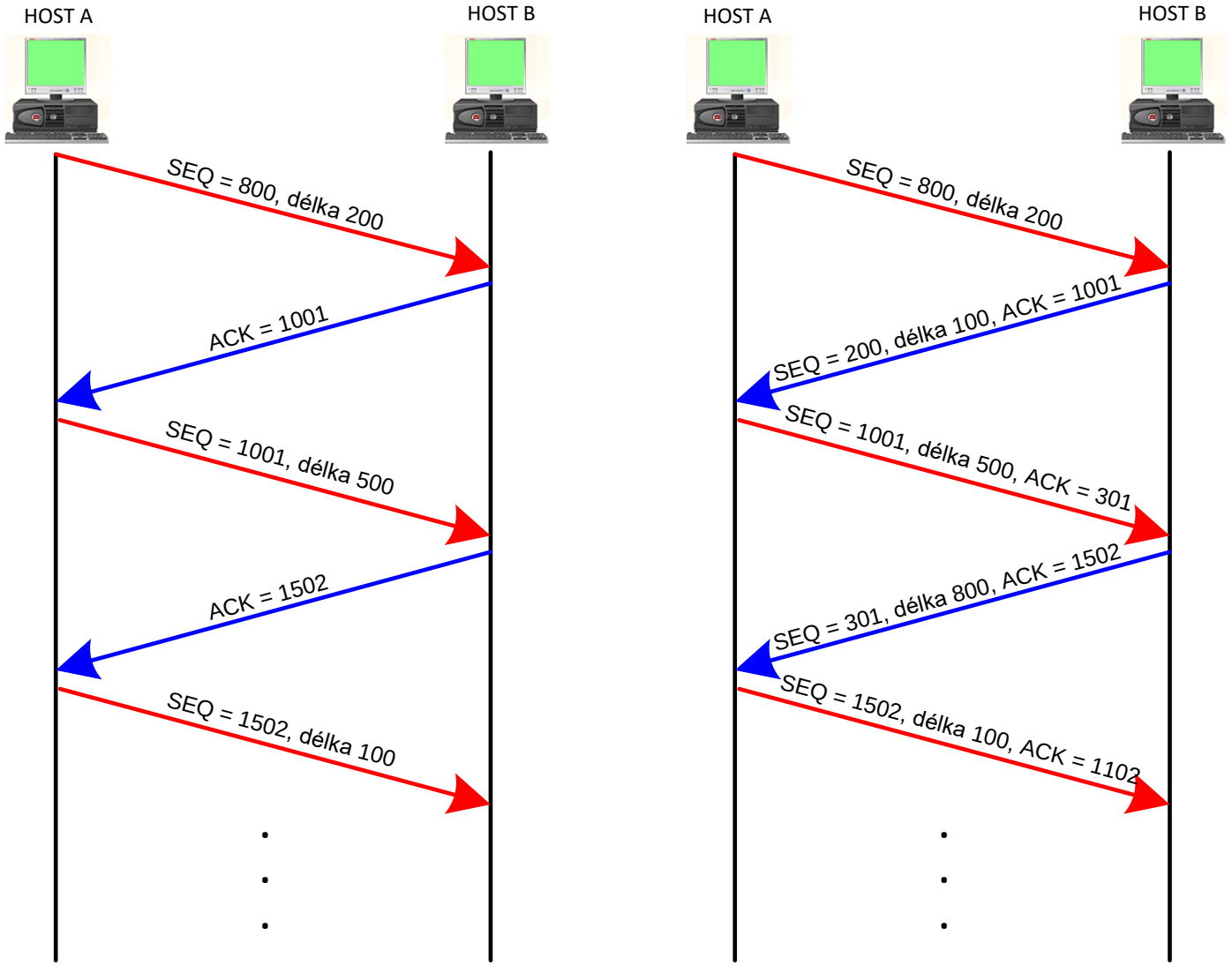
Four-way handshake

[FIN] > [ACK], [FIN] > [ACK]

Průběh komunikace u protokolu TCP

Vlevo jednosměrná komunikace
Vpravo obousměrná komunikace

Pro jednoduchost pouze režim stop-and-wait, při r přenosu neefektivní



Velikost okna u protokolu TCP a návaznost na řízení provozu

- TCP poskytuje mechanizmus pro řízení toku dat
  - napomáhá celkové spolehlivosti přenosu
- Záhlaví TCP obsahuje pole ***délka okna***
  - umožňuje, aby přijímač nastavil, kolik mu vysílač může maximálně odeslat bajtů bez čekání na potvrzení
  - povolení k odesílání dalších dat
  - nedochází ke zbytečnému zahlcení a zahazování dat
- Problematika fungování tohoto mechanizmu (***technika posuvného okna***)
  - diskutována již v rámci spojové vrstvy
  - slouží k řízení toku dat, chybových stavů i zahlcení
  - specifika u TCP nad rámec kurzu

Příklady využití protokolu TCP

- robustní protokol, významná režie přenosu, mnoho funkcí pro aplikace
- využití časté např. u
  - HTTP (*HyperText Transfer Protocol*) – přenos webových stránek
  - FTP (*File Transfer Protocol*) – přenos souborů
  - SMTP (*Simple Mail Transfer Protocol*) – přenos elektronické pošty
  - …
- všechny tyto protokoly potřebují spolehlivou službu; TCP není jediná v současnosti používaná možnost (QUIC nad UDP)
- TCP využíván běžně u více aplikačních protokolů než UDP

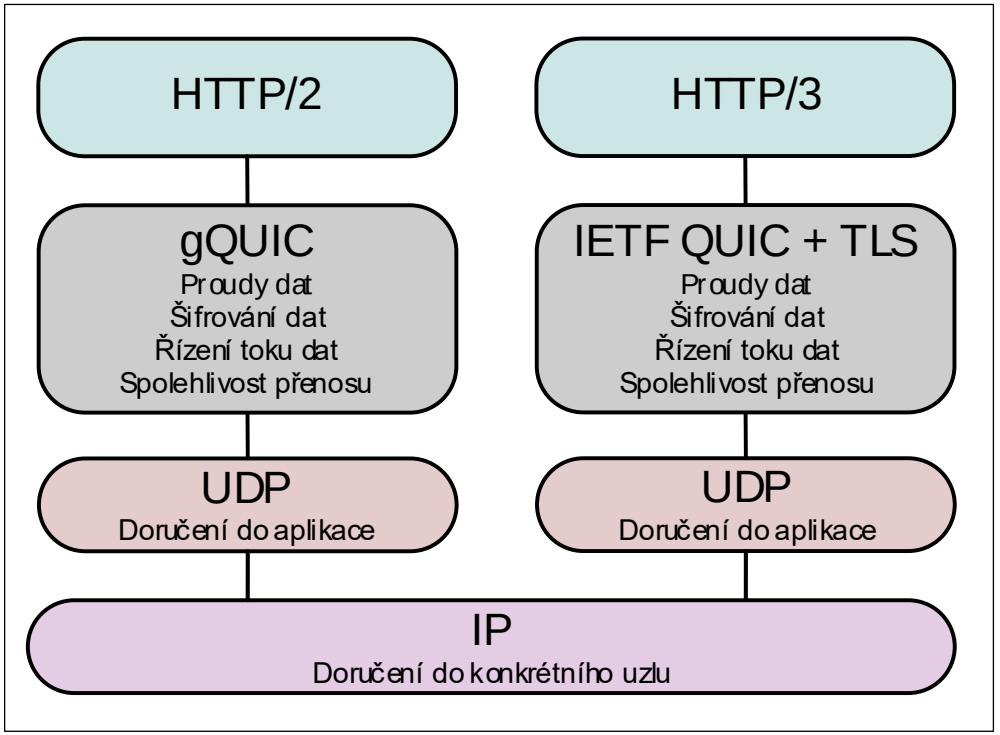
PROTOKOL QUIC

Google QUIC

- QUIC = Quick UDP Internet Connection, gQUIC
- Chrome *2013
- Úzce provázán s HTTP/2
- Alternativa k TCP protokolu
- Běží nad UDP, další transportní vrstva funkčně podobná TCP
- Není součástí OS, ale aplikace – zrychlení vývoje

IETF QUIC

- *2016 snaha o standardizaci QUIC u IETF, stále probíhá
- QUIC, popř. IETF QUIC
- Nekompatibilní s gQUIC, přesto velmi podobné
- Určen pro HTTP/3
- Pevně svázán s TLS – šifrování a autentizace



IETF QUIC – základní vlastnosti

- Podobnost s TCP – spojení
- Proudy dat v rámci spojení (STREAM x FRAME)
- Proudy dat na sobě velmi nezávislé
- Řízení toku dat na úrovni spojení i proudu dat
- Snížení zpoždění před přenosem dat v šifrované podobě
  - TCP: 2 RTT + TLS: 1 RTT
  - QUIC + TLS: 1 RTT či i 0 RTT
- Přechod spojení mezi adresami (L3 i L4) díky Connection ID
- Zkušební implementace
  - Chrome, Firefox, vybrané webové servery
  - https://http3-explained.haxx.se/en/proc-status

Doplnění k transportní vrstvě

- Existují i jiné protokoly transportní úrovně
  - např.
    - SCTP (*Stream Control Transmission Protocol*)
    - RSVP (*Resource Reservation Protocol*)
    - RUDP (*Reliable User Datagram Protocol*)
  - typicky snaha nějakým způsobem kombinovat vlastnosti TCP a UDP
  - spíše okrajové využití
- Zařízení transportní vrstvy
  - žádné síťové zařízení přímo transportní úrovně
  - součást stavových firewallů (sledují stav spojení TCP), nad rámec kurzu