**12**

**Protokoly transportní vrstvy – TCP, UDP, port, sliding window, navázání a ukončení TCP spojení**

**Port**

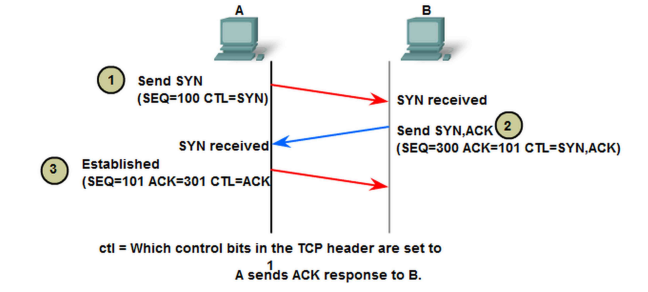
* rozhraní mezi aplikační a transportní vrstvou
* celé kladné číslo, využívané protokoly transportní vrstvy k rozlišení cílového aplikačního procesu
* Dělení:
  + Well-known
    - slouží službám a aplikacím, využívají je protokoly s těmito aplikacemi svázanými
    - přiděluje IANA
    - např.: http
  + Registrované porty
    - slouží na straně uživatele
    - registruje IANA
  + Privátní
    - slouží pro navázání spojení
    - přiřazovány dynamicky
    - klientské procesy je volí náhodně
  + Socket
    - uspořádaná dvojice IP adresa user-end systému + číslo portu
    - jednoznačně určuje proces v rámci sítě
    - tvar: ip.ip.ip.ip.port
* Synchronní přístup k portům
  + výpočet v aplikaci se pozastaví po dobu přístupu k portu
  + data, která dorazí, než je proces připraven je přijmout, se uloží do vyrovnávací paměti

**TCP protokol**

* zapouzdřuje data z aplikační vrstvy do TCP segmentu
* zajišťuje že všechna data dorazí v pořádku a ve správném pořadí
* navazuje spojení, vytvoří mezi komunikujícími procesy virtuální full-duplex
* spojení tvoří dvojice soketů
* k datům se přidává TCP hlavička (velikost 20B)
* hlavička
  + Sequence Number (SN) – nastaven flag SYN – počáteční číslo sekvence, jinak pořadové číslo prvního okteteu
  + Acknowledgement Number (ACKN) – potvrzovací číslo, vysílá se během celého spojení, představuje SN dalšího segmentu
  + Header Lenght – určuje délku hlavičky v Bytech
  + Flags – řízení relace při zacházení se segmenty
    - URG – Urgent pointer field significant
    - ACK – Acknowledgement field significant (potvrzení)
    - PSH – Push function
    - RST – Reset the connection
    - SYN – Synchronize sequence numbers
    - FIN – No more data from sender (žádost o ukončení)
* Window size
  + počet oktetů, které mohou najednou projít před očekávaným potvrzením
  + pří klient-server spojení specifikuje velikost okna klient pro server, podobně nastaví velikost okna i server pro klienta
  + během komunikace se může měnit
  + hodnota nemusí být symetrická, každá strana může mít jinou velikost
  + závisí na velikosti paměti příjemce i odesílatele
* TCP Checksum – kontrolní součet hlavičky, pseudohlavičky a dat

**Navazování TCP spojení**

* three-way handshake
  + klient A pošle segment s počáteční hodnotou sekvence sloužící jako žádost o zahájení komunikace
  + klient B vrátí segment obsahující potvrzovací hodnotu rovnou přijaté sekvenční hodnotě + 1 a vlastní synchronizační hodnotu
  + klient A potvrdí hodnotou sekvence o jednu vyšší a dokončí proces navazování komunikace



* potvrzování
  + pozitivní – potvrzuje se příjem všech oktetů předcházejících číslu potvrzení
  + kumulativní – byty se potvrzují postupně v toku dat
* potvrzování doručných oktetů provádí přijímací stanice podle pořadových čísel bytů
* vysílací stanice si po odeslání segmentu nastaví časovač, pokud vyprší před přijetím potvrzení, pošle segment znovu
* duplicitní segmenty cílová stanice zahodí

**Ukončování TCP spojení**

* klient A pošle segment se žádostí o ukončení
* klient B obdržení žádosti a ukončení komunikace potvrdí a pošle žádost o ukončení
* klient A ukončení opět potvrdí

**Seřazování segmentů**

* segmenty se řadí podle pořadového čísla v hlavičce
* při navazování se nastaví počáteční pořadové číslo segmentu, toto číslo se zvyšuje podle počtu bajtů, které byly předány, lze identifikovat chybějící segmenty
* segmenty se podle pořadového čísla seřadí a předají aplikační vrstvě
* nesouvisle doručené segmenty se podrží a zpracují se až po doručení všech segmentů

**Proces řízení toku (flow-control)**

* TCP poskytuje mechanismy pro upravování toku a zefektivňování průtoku dat
* Sliding Window
  + při zpoždění při obdržení potvrzení odesílatel nevysílá nic pro danou relaci
  + když je síť nebo zdroje příjemce vytíženy, zpoždění se zvýší, přenosová rychlost se sníží a dojde k dynamickému snížení velikosti okna
  + pokud příjemce nestačí přijímat data, zažádá o znovu zaslání segmentu se sníženou velikostí okna
  + po určité době přenosu bez datových ztrát začne příjemce opět zvětšovat velikost okna

**UDP protokol**

* nenavazuje spojení a nekontroluje doručení segmentů (datagramů)
* je nenáročný, ale nespolehlivý, kontrolu doručení zajišťuje aplikační vrstva
* nelze řídit transport
* je efektivní – vysoká rychlost, nízká náročnost na zdroje
* užití:
  + VoIP (Voice over Internet Protocol)
  + video streaming
  + DNS
  + online hry
  + atd.
* příklady transportní služby:
  + RIP
  + NTP
  + SNMP
  + DNS
* pokud datagramy dorazí v různém pořadí, neřadí se a datagramy doručené mimo pořadí jsou ztracené (navždy-vždy-dy-dy)

**Kontrolní součet**

* při přenosu přes IP protokol se kontrolní součet posílá z kombinace:
  + pseudo IP hlavičky
  + segmentu (TCP) nebo datagramu (UDP)