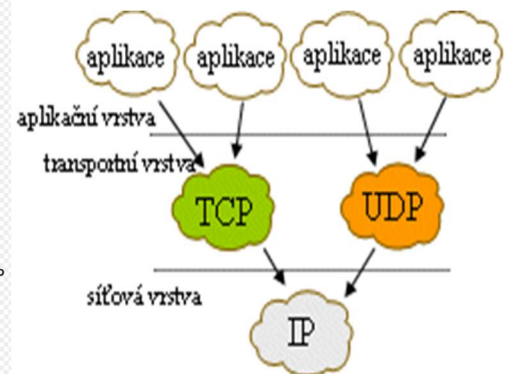
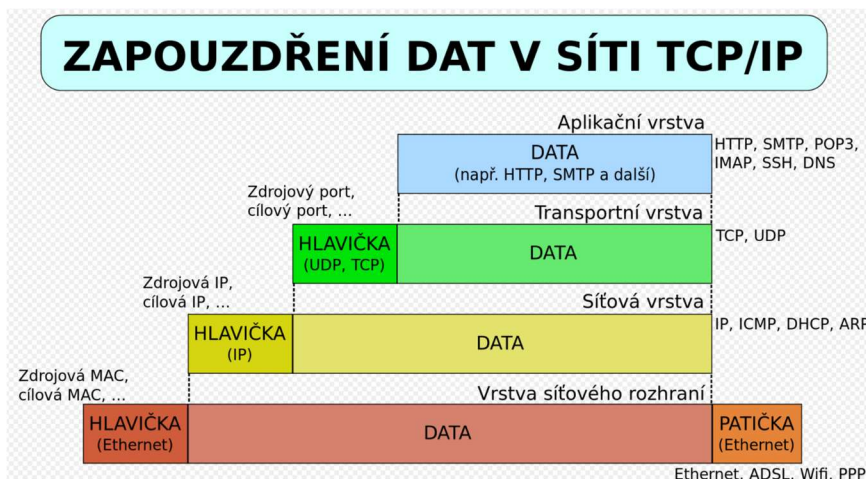


# Maturitní témata 2020/21

## Otázka č. 12 – Počítačové sítě a programování

### Transportní vrstva:

- transport layer, čtvrtá vrstva podle modelu ISO/OSI, umožňuje adresovat přímo aplikace (např. v protokolech TCP/IP pomocí čísel portů), poskytuje transparentní spolehlivý přenos dat s požadovanou kvalitou, vyrovnává různé vlastnosti a kvalitu přenosových sítí, provádí převod transportních adres na síťové ale nestará se o směrování
- zajišťuje „přizpůsobení“ mezi možnostmi nižších vrstev a požadavky vyšších vrstev, end to end komunikaci (jeden z hlavních principů při návrhu TCP), zajišťuje multiplexing (proces a schopnost přenosu více analogových nebo digitálních signálů pomocí jednoho kanálu)/demultiplexing (opak multiplexingu)
- jedním z hlavních úkolů této vrstvy je: rozlišovat příjemce a odesílatele v uzlu (multiplex, demultiplex), řídit datový tok



### TCP:

- Transmission Control Protocol je nejpoužívanějším protokolem transportní vrstvy v sadě protokolů TCP/IP používaných v síti Internet, použitím TCP mohou aplikace na počítačích propojených do sítě vytvořit mezi sebou spojení přes které mohou obousměrně přenášet data
- protokol garantuje spolehlivé doručování a doručování ve správném pořadí, TCP také umožňuje rozlišovat a rozdělovat data pro více aplikací (např. webový server, emailový server) běžících na stejném PC
- TCP využívá mnoho populárních aplikačních protokolů a aplikací na internetu včetně WWW, e-mailu a SSH

### Fungování TCP:

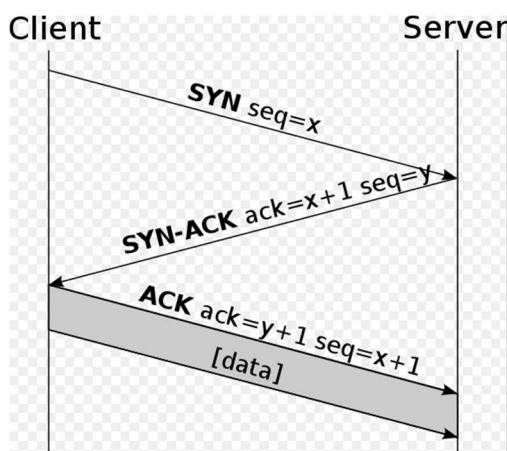
- protokol TCP dostává od aplikace proud bajtů k doručení do sítě, TCP si proud rozdělí do segmentů jejichž velikost závisí na parametru MTU, tyto segmenty potom pošle

- každý paket dostane své pořadové číslo, díky kterému se ověří že se pakety neztratily a dorazily ve správném pořadí, TCP modul který pakety přijal posílá zpět potvrzení o přijetí, pokud odesílatel nedostane potvrzení včas nebo vůbec posílá pakety znovu

#### Navázání spojení:

- Klient odešle na server datagram s nastaveným příznakem SYN (synchronization) a náhodně vygenerovaným pořadovým číslem ( $x$ ), potvrzovací číslo = 0
- Server odešle klientovi datagram s nastavenými příznaky SYN a ACK (acknowledge), potvrzovací číslo =  $x+1$ , pořadové číslo je náhodně vygenerované ( $y$ )
- Klient odešle datagram s nastaveným příznakem ACK, pořadové číslo =  $x+1$ , číslo odpovědi =  $y+1$

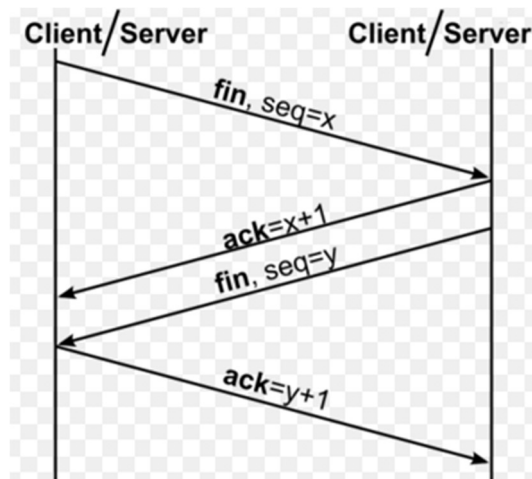
- obě strany si pamatují pořadové číslo vlastní i protistrany, používají se totiž i pro další komunikaci a určují pořadí paketů



#### Ukončení spojení:

- pobíhá podobně jako jeho navázání, používá se k tomu příznaků FIN a ACK:

- Strana, která nechce posílat další data odešle datagram s nastaveným příznakem FIN
- Protistrana odpoví datagramem s nastaveným příznakem ACK s potvrzovacím číslem o jedničku větším, než bylo pořadové číslo v datagramu s příznakem FIN
- Druhá strana která ukončuje posílání dat odešle datagram s nastaveným příznakem FIN
- Protistrana odpoví datagramem s nastaveným příznakem ACK



## UDP:

- User Datagram Protocol jeden ze sady protokolů internetu, nedává záruky na datagramy (základní jednotka která je přepravovaná v počítačové síti), někdy je označován jako nespolehlivý ale je jenom bez záruky doručení což je hlavní rozdíl oproti protokolu TCP
  - protokol UDP je vhodný pro nasazení které vyžaduje jednoduchost, malá režie nebo pro aplikace pracující systémem otázka-odpověď (např. DNS, sdílení souborů LAN), jeho bezstavovost je užitečná pro servery které obsluhují mnoho klientů nebo pro nasazení kde se počítá se ztrátami datagramů a není vhodné aby se ztrácel čas novým odesláním (starých) nedoručených zpráv
  - pokud data doručí můžou být ve špatném pořadí, UDP také neposkytuje jakýkoli druh kontroly zahlcení pokud v síti dojde k přetížení pakety by mohly být ztraceny a samozřejmě nebude mít na starosti jejich opětovné odeslání jako je tomu u TCP
- proto UDP nemá řízení přetížení, řízení toku nebo řízení chyb → velmi nespolehlivý protokol

