

11. - Komunikace v síti – Využití UDP/TCP protokolu

TCP

Většinou pro komunikaci v síti používám TCP. Abychom si tyto pakety mohli posílat bezstrátově, používáme TCP. Všechny packety a data se ověřují přes handshake.

Začíná tím, že před spojením naváže spojení. Poté zjistí, jestli někdo poslouchá. Pošlou se mu Acknowledge, ten odpoví zpátky. Poté se pošle request a nakonec se zašle

TCP Umí vytvořit posloupnost dat. Data se odesílají v nějakém pořadí. Dokáže ztracené packety poslat znovu. Zkontrolujeme, jestli nějaké chybí a poté je zašleme znovu. TCP nemá Broadcasting, teda rozesílání na více adres.

TCP je mnohem pomalejší, ale poskytuje všechna data (někdy žádná)

TCP se nejvíce využívá třeba v emailech, hledání na stránkách – nejširší využití se používá na webu, základ HTTP protokolu je TCP protokol.

TCP vždy posílá na jednu stranu data, strana pošle zpátky acknowledge.

Mezi výhody TCP patří to, že pracuje bez řešení operačního systému, dá se optimalizovat rychlost.

Jeho nevýhody jsou to, že je pomalejší a když neprijdou nějaká data, tak neprijdou vůbec žádná.

U TCP je možnost, že někdy přijdou data lehce přeházená, ale vždy se nakonec složí.

UDP

UDP protokol posílá data rozházeně, některé mohou třeba chybět. U UDP se mohou nějaké ztratit a není zaručené jejich doručení.

UDP broadcasting má – to znamená, že může rozeslat packety na více zařízení.

UDP je velmi rychlé, ale je riziko ztracených packetů.

UDP je skvělé pro vysílání live a nebo při hraní her, kde záleží na rychlosti. Posílá velice rychle a nečeká na potvrzení. Hlasové volání, skype, hovory a podobné.

Neexistuje systém na potvrzení doručení dat.

V Pythonu

V Pythonu používáme knihovnu Socket. Vytvoříme instanci, tudíž Berkeleyeho socket a necháme ho poslouchat. Jakmile se uživatel připojí, pošlemu mu nějakou zprávu a poté ho necháme zadávat nějaké zprávy a tiskneme si je do konzole (Ano, můžeme je ukládat jinak normálně do souboru nebo je zpracovat). Jakmile uživatel zadá „konec“, tak se connection ukončí.

