NA Kapitel 3

Arne Mejlholm

15th June 2004

Hermed følger noter i punktform om kapitel 3.

- **Transportlagets ansvarsområde:** Transportlaget sørger for at der for applikationslaget ser ud til at være en fast forbindelse mellem to processer på forskellige end-systems. Dette lag eksisterer (oftest!) kun på end-hosts.

- **Services for netværkslaget:** Dette bruger IP, som er en best-effort delivery service, den giver altså ingen garanti for delivery, nogle gange osse kaldet unreliable service. Vi husker også på at en end-host har en IP addresse.

- **(De)Multiplexing:** Når transportlaget laver en besked om til et segment med source og destination socket(socket indeholder ip og socket nr.) kaldes det multiplexing og den omvendte procedure kaldes demultiplexing.

- **Reliable Data Transfer:** Går i sin enkelthed ud på at bygge en pålidelig data kanal. Dette bliver mærkbart mere besværligt når protokollen bygger på et unreliable underlag, som IP er.

Dette sidste emne kræver yderligere forklaring, hvilket vi kigger på i næste sektion.

1 Reliable Data Transfer

Der er flere faktorer der skal tages i betragtning: Hvad hvis en pakke bliver korrupt eller bliver væk? Først og fremmest får vi brug for en checksum hvorudfra der kan detecteres om en pakke er defekt. Desuden skal modtageren have mulighed for at svare om den fik pakken i god eller dårlig stand. Disse svar kaldes ACK og NAK. Muligheden for at et ACK/NAK kan blive væk, ordnes ved at indføre en timer mekanisme på senderen, der sender pakken igen, hvis den ikke modtager ACK/NAK. For at håndtere retranmission, skal der også indføres

pakke nr. (binær nr. er tilstrækkeligt når der ikke er pipeline). Hermed kan der detekteres pakker der er kommet dobbelt og der kan holdes styr på hvilke ACK/NAK der er blevet sendt til en given transmisison.

For at se principperne i brug, se fig. 3.16 og modtager 3.14 samt sender 3.15.

2 Pipelined Reliable Data Transfer

For at udnytte forbindelsen ordentlig og undgå at senderen står og venter på et ACK for hver pakke, introducerer vi pipelined rdt.

Se fig. 3.19 side 217.

Der indføres en kø og to pointere, base og nextseqnum. Hermed kan køen opdeles i fire dele:

- [0, base-1] De pakker der er modtaget ACK på.
- [base, nextseqnum − 1] De pakker der er sendt, men ikke modtaget ACK på.
- [nextseqnum, base + N − 1] Pladser i køen der ikke er sendt, men tilgængelige.
- [base + N, ...] Pladser der ikke er tilgængelige.

Denne maskine (med variablerne) sender en strøm af pakker og venter så på at der kommer ACK på dem en efter en(angivet ved base pointeren). Når der kommer et korrekt ACK, flyttes pointeren og der sendes en ny pakke af sted. Hvis der sker en timeout, så sendes alt fra base og frem igen.

Selected repeat kan findes på side 222.

3 TCP

En detaljeret beskrivelse af TCP findes fra side 228 og frem, men dette bliver kun betragtet som baggrundsviden, da der ikke er nogen konkrete spørgsmål om TCP.

4 Congestion Control

Congestion Control, dvs. kontol over netværkets mætning, kun udføres to steder, på de enkelte end-hosts og på netværkslaget/de enkelte links. Da IP protokollen ikke giver nogen form for understøttelse for den sidstnævnte, må vores TCP protokol selv ordne det på end-host systemet.

TCP gør det ved at regulere på den rate, hvormed den sender pakker ind i netværket. Dette gøres ved at regulere på en grænse for hvor stort mellemrum der er mellem antallet af sendte bytes og antallet af bytes der er blevet ACK'ed på.

4.1 Additive-Increase, Multiplicative-Decrease

Sende raten på TCP reguleres på to måder: når der opfanges et miss, så halveres senderaten(CongWin)(dog højest ned til en minimumsgrænse), og når der observeres at alt kører fint, så prøver den lige så stille at forhøje CongWin, da antagelsen er at der er så er ubrugt båndbredde. Dette kaldes congestion aviodance.

4.2 Slow Start

TCP starter langsomt ud under dens intiliserings fase, men herunder lader den CongWin fordobles hver gang der er ACK problemer. Herefter probes der som beskrevet i sidste sektion.

4.3 Reaction to Timeout Events

Der skelnes mellem om en pakke er timed-out eller der er kommet trippel ACK. Når der observeres en time-out, så sættes CongWin til 1 MSS (Minimum værdi) og der laves en slow start(som vokser eksponentielt indtil der observeres trippel ACK, fast recovery) indtil TRESHOLD nåes. TRESHOLD er det halve af den værdi hvormed der sidst blev konstanteret en timeout. Herefter fortsætter TCP med liniær vækst.

5 Fairness

Enkelte, dvs. ikke paralelle, TCP forbindelser siges at være fair, da de har en tendens til at fordele linien ligeligt, i modsætning til UDP som bare "pumper" ind i netværket. Dog vil der med adskillige parallelle TCP forbindelser være en skævridning overfor en anden applikation, da denne kun får 1/N'te del af forbindelsen, hvor N er antallet af forbindelser.