



Hochschule für
Technik und Wirtschaft
Dresden
University of Applied Sciences

**Dokumentation Belegarbeit Rechnernetze/Kommunikationssysteme
"Dateitransfer"**

Felix Müller

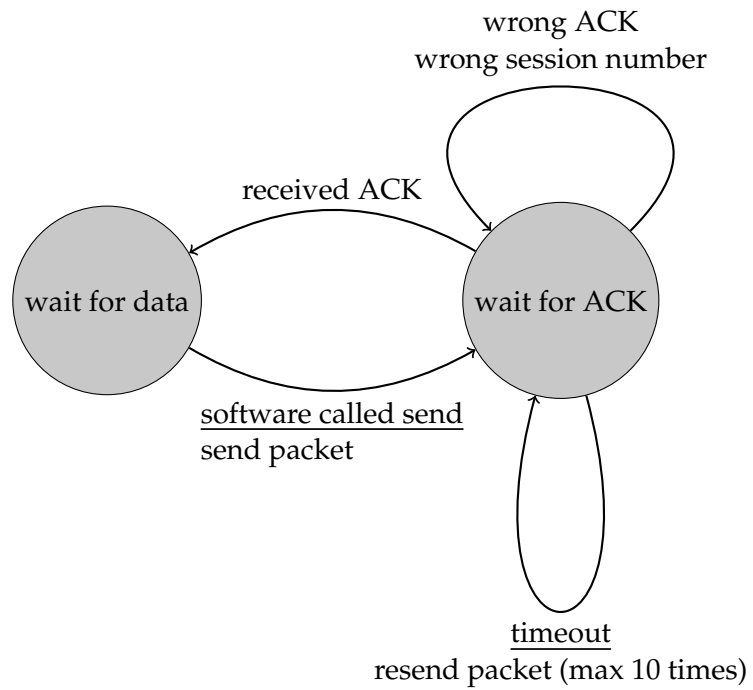
20. Juli 2021

Inhaltsverzeichnis

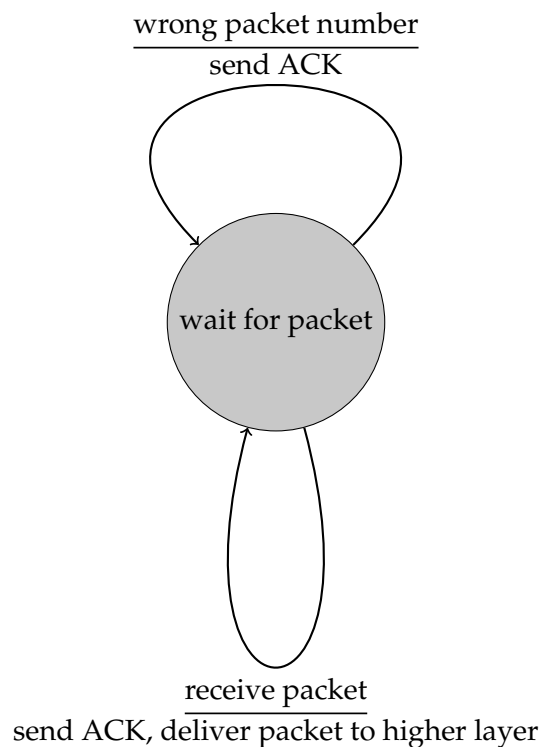
1 Zustandsdiagramme Stop And Wait Protokoll	3
1.1 Zustandsdiagramm Stop And Wait Protokoll (Sender)	3
1.2 Zustandsdiagramm Stop And Wait Protokoll (Empfänger)	4
2 theoretischer und realer Durchsatz (Berechnung, Vergleich und Bewertung)	4
2.1 Berechnung des theoretischen Durchsatzes	4
2.2 Bestimmung des realen Durchsatzes	5
2.2.1 Systemspezifikationen und Randbedingungen	5
2.2.2 Messverfahren	6
2.3 Bewertung der Durchsätze	6
3 Probleme, Limitierungen, Verbesserungen	7

1 Zustandsdiagramme Stop And Wait Protokoll

1.1 Zustandsdiagramm Stop And Wait Protokoll (Sender)



1.2 Zustandsdiagramm Stop And Wait Protokoll (Empfänger)



2 theoretischer und realer Durchsatz (Berechnung, Vergleich und Bewertung)

2.1 Berechnung des theoretischen Durchsatzes

Die Datenrate (brutto) wird mit $r_b = 10 \frac{Gb}{s}$ angenommen. Die Paketlänge beträgt für die konkrete Implementierung $L = 1403 B$, wovon $3 B$ Headerdaten sind. Für Hin- und Rückkanal soll ein Delay von $T_a = 10 ms$ und eine Paketverlustwahrscheinlichkeit von $P_{de} = P_{ru} = 0,1$ angenommen werden.

$$T_p = \frac{L}{r_b} = \frac{1403 * 8 \text{ b}}{1 * 10^{10} \frac{\text{b}}{\text{s}}} = 0,00000112 \text{ s}$$

$$R = \frac{(1400 + 3) B}{1400 B} \approx 1$$

$$T_w \approx 2 * T_a = 0,02 \text{ s}$$

$$\eta_{sw} = \frac{T_p}{T_p + T_w} (1 - P_{de}) (1 - P_{ru}) R = \underline{0,000453346}$$

Der theoretisch maximale Durchsatz für das implementierte *Stop-And-Wait-Protokoll* beträgt also

$$\eta_{sw_{theoretisch}} = 0,000453346 \hat{=} 0,0453356\%$$

2.2 Bestimmung des realen Durchsatzes

Das Programm verfügt über die Möglichkeit auf den Hin- und Rückkanal Verzögerungen und Paketverluste zu simulieren. Die mittlere Verzögerung und die mittlere Paketverlustwahrscheinlichkeit können dem *Shell-Skript filetransfer* als Parameter übergeben werden. Zur Bestimmung der realen Datenraten wurden Messungen durchgeführt.

2.2.1 Systemspezifikationen und Randbedingungen

- Rechnerart: Laptop
- Hauptspeicher:

Data Width	64 bits
Size	8192 MB
Type	DDR4
Type Detail	Synchronous
Configured Clock Speed	2400 MT/s
Anzahl	2

- Prozessor:

Type	Central Processor
Family	Core i7
Version	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
Current Speed	2000 MHz
Core Count	4 (4 Enabled)

- Betriebssystem:
Ubuntu Linux
Kernelversion: 5.0.0-37-generic
Ubuntuversion: 18.04.1-Ubuntu
- Randbedingungen:
Das Programm war der einzige Benutzerprozess. Die Dateilänge ist 84.856 B

2.2.2 Messverfahren

Zur Ermittlung der realen Nettodatenrate wurde im Client-Programm direkt vor dem Versenden des ersten Pakets (t_1) und direkt nach dem Empfang und der Prüfung des letzten Bestätigungspakets die Zeit (t_2) gestoppt (*java: System.currentTimeMillis()*). Die reale Nettodatenrate ergibt sich folgendermaßen

$$r_n = \frac{\text{Dateilänge}}{t_2 - t_1}.$$

107.0	109.0	117.0	118.0
115.0	114.0	106.0	103.0
115.0	109.0	113.0	112.0
115.0	110.0	118.0	112.0
119.0	125.0	106.0	113.0

Tabelle 1: Messwerte in $\frac{KB}{s}$

Für diese Messwerte beträgt die mittlere reale Nettodatenrate

$$\bar{r}_n = 112,8 \frac{KB}{s}.$$

Als Bruttodatenrate wird wieder $r_b = 10 \frac{Gb}{s}$ angenommen. Der mittlere reale Durchsatz ergibt sich wie folgt

$$\eta_{\text{sw,real}} = \frac{r_n}{r_b} = \frac{112,8 \frac{KB}{s}}{10 \frac{Gb}{s}} = 0,00009024 \hat{=} 0,009024\%.$$

2.3 Bewertung der Durchsätze

Der Mittelwert der durch das Programm gemessenen Durchsätze beträgt nur etwa 2% des theoretisch möglichen Durchsatzes. Dies hat verschiedene Ursachen:

- Daemons und andere Hintergrundprozesse, die ebenfalls den Netzwerkadapter des Computers nutzen, beeinträchtigen den tatsächlichen Durchsatz, da mehrere (zeitlich parallele) Zugriffsanfragen Scheduling und Ressourcenzuteilung nötig machen.

- Auch wenn das Programm bei der Messung der Datenraten als einziger Benutzerprozess ausgeführt wurde, wird es keine der benötigten Ressourcen (hauptsächlich CPU und Netzwerkadapter) zu 100% der Laufzeit vom Betriebssystem zugeteilt bekommen. Durch Benutzung des Bashbefehls **time** lässt sich die reine Rechenzeit eines Programms feststellen. Unter den gegebenen Parameter der Messung lassen sich reine Rechenzeiten zwischen 0,02 s und 0,05 s feststellen, während der Mittelwert der realen Laufzeiten ca. 0,9 s beträgt. Die Übertragungszeit wird also durch Unterbrechung der Prozesse von Server und Client durch das Betriebssystem verlängert.
- Die Dauer für Berechnungen, Vergleiche und lesende beziehungsweise schreibende Speicherzugriffe findet in der Formel zur Berechnung des theoretischen Durchsatzes keine Beachtung.

3 Probleme, Limitierungen, Verbesserungen

- Der Server kann zu einem bestimmten Zeitpunkt nur genau eine Datei von genau einem Client empfangen. Wünschenswert wäre eher, dass der Server zu einem Zeitpunkt über mehrere Verbindungen kommunizieren kann. Dies ließe sich realisieren, indem für jede neue Verbindung serverseitig ein neuer Thread gestartet wird, der die Datei des Klienten entgegennimmt.
- Die Kommunikation erfolgt unverschlüsselt.
- Durch Implementieren des Go-Back-N- oder Selective-Repeat-Protokolls ließen sich bessere Durchsätze und bessere reale Datenraten erzielen.