

Moritz Hahn
Sarah Alten-
krüger

1	2	Σ

Übungsblatt Nr. 8 (Abgabetermin 17.06.2024)

Aufgabe 8.1

Selective-Repeat (SR) Protokoll für zuverlässigen Datentransfer

1.

Das Sendefenster ist $[k; k + N - 1]$, und der Sender hat bereits N Pakete verschickt, aber noch keine Bestätigungen erhalten. Das Empfangsfenster des Empfängers deckt normalerweise N aufeinanderfolgende Sequenznummern ab.

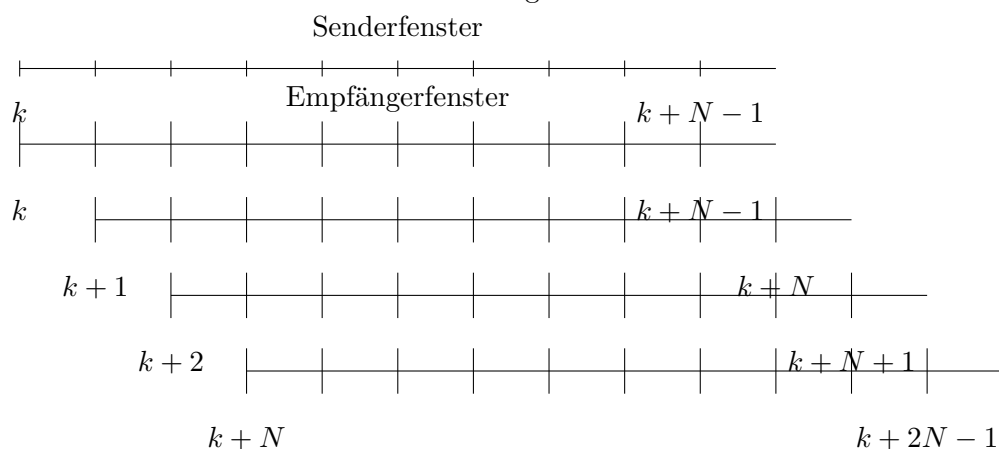
Da die Reihenfolge der Pakete nicht verändert wird und der Empfänger nur Pakete akzeptiert, die innerhalb seines Empfangsfensters liegen, können wir folgende mögliche Empfangsfenster in Betracht ziehen:

- Wenn der Empfänger keine Pakete korrekt empfangen hat, beginnt sein Empfangsfenster bei k . Somit ist das Empfangsfenster $[k; k + N - 1]$.
- Wenn der Empfänger alle N Pakete korrekt empfangen hat, verschiebt sich sein Empfangsfenster um N Positionen weiter und wird $[k + N; k + 2N - 1]$.

Zwischen diesen Extremen gibt es jedoch viele mögliche Szenarien, da der Empfänger einige, aber nicht alle Pakete erhalten haben könnte. Dies führt dazu, dass das Empfangsfenster innerhalb der folgenden Grenzen liegen kann:

- Minimal: $[k; k + N - 1]$
- Maximal: $[k + N; k + 2N - 1]$

Hier ist eine Skizze zur Veranschaulichung:



2.

Wenn wir annehmen, dass Sequenznummern zyklisch sind, bedeutet das, dass die Sequenznummern nach einer bestimmten Anzahl wieder von vorne beginnen. Angenommen, die maximale Sequenznummer ist $M - 1$, dann wird nach $M - 1$ die nächste Sequenznummer wieder 0 sein.

Wenn alle N Pakete verloren gegangen sind und der Sender nach einer Zeitüberschreitung alle Pakete erneut senden muss, müssen die Sequenznummern in der Lage sein, den gesamten Datenstrom korrekt zu identifizieren und einzusortieren.

Um sicherzustellen, dass der Empfänger die erneut übertragenen Pakete korrekt einordnen kann, müssen die Sequenznummern in einem Bereich liegen, der das Sendefenster des Senders und das Empfangsfenster des Empfängers vollständig abdeckt.

Die Anzahl der benötigten Sequenznummern M muss daher mindestens $2N$ betragen, um Verwechslungen zu vermeiden. Dies liegt daran, dass das Empfangsfenster sich um bis zu N Positionen verschieben kann und der Empfänger unterscheiden muss, ob es sich um neue oder erneut gesendete Pakete handelt.

$$M \geq 2N$$

3.

Beim Go-Back-N (GBN) Protokoll sendet der Sender alle Pakete im Sendefenster sequenziell und wartet auf die Bestätigung des ältesten unbestätigten Pakets, bevor er weitere Pakete senden kann. Wenn ein Paket verloren geht oder beschädigt wird, muss der Sender alle Pakete ab dem verlorenen oder beschädigten Paket erneut senden.

Um sicherzustellen, dass der Empfänger die Pakete korrekt einordnen kann, muss die Anzahl der Sequenznummern mindestens so groß sein wie die Größe des Sendefensters plus eins. Das liegt daran, dass der Empfänger in der Lage sein muss, zwischen dem neuen Sendefenster und dem alten Sendefenster zu unterscheiden.

Die Anzahl der benötigten Sequenznummern M für das GBN Protokoll beträgt daher $N + 1$.

$$M \geq N + 1$$

4**Selective-Repeat (SR) Protokoll**

Beim SR-Protokoll können Pakete innerhalb des Sendefensters des Senders und des Empfangsfensters des Empfängers in beliebiger Reihenfolge empfangen werden. Wenn das Netzwerk die Paketreihenfolge verändert, muss der Empfänger in der Lage sein, Pakete korrekt anhand ihrer Sequenznummern zuzuordnen.

Da das Empfangsfenster des Empfängers N Positionen umfasst und das Netzwerk die Paketreihenfolge verändern kann, müssen wir sicherstellen, dass keine Verwechslung zwischen neuen und alten Paketen entsteht. Daher bleibt die Anforderung an die Sequenznummernbereiche unverändert:

$$M \geq 2N$$

Diese Anzahl gewährleistet, dass alle Pakete im Sendefenster eindeutig identifizierbar bleiben, selbst wenn die Reihenfolge geändert wird.

Go-Back-N (GBN) Protokoll

Beim GBN-Protokoll sendet der Sender alle Pakete im Sendefenster sequentiell und erwartet eine Bestätigung für jedes Paket. Wenn ein Paket verloren geht oder beschädigt wird, werden alle nachfolgenden Pakete erneut gesendet. Wenn das Netzwerk die Paketreihenfolge verändert, ist es wichtig, dass der Empfänger immer noch die Pakete korrekt in der richtigen Reihenfolge einsortieren kann.

Beim GBN-Protokoll ist das Empfangsfenster effektiv nur ein Paket groß, da der Empfänger nur das nächste erwartete Paket annimmt. Die Sequenznummern müssen jedoch ausreichen, um alle Pakete im Sendefenster zu unterscheiden:

$$M \geq N + 1$$

Diese Anforderung bleibt ebenfalls unverändert, da die Sequenznummern bereits so gewählt sind, dass sie die Pakete im Sendefenster eindeutig identifizieren können, selbst wenn die Reihenfolge verändert wird.

Zusammenfassung

- Für das SR-Protokoll sind weiterhin mindestens $2N$ Sequenznummern erforderlich, um eine korrekte Zuordnung der Pakete zu gewährleisten, auch wenn die Paketreihenfolge verändert wird.
- Für das GBN-Protokoll sind weiterhin mindestens $N + 1$ Sequenznummern erforderlich, um die Pakete korrekt in der richtigen Reihenfolge einzuordnen, selbst wenn die Paketreihenfolge verändert wird.

8.2

1.

TCP Server- und Client-Sockets sind durch die Kombination folgender Attribute eindeutig bestimmt:

- IP-Adresse des Servers
- Portnummer des Servers
- IP-Adresse des Clients
- Portnummer des Clients

2.

Ein TCP Server-Socket stellt die folgende Funktionalität zur Verfügung:

- **bind**: Verbindet den Socket mit einer bestimmten IP-Adresse und Portnummer.
- **listen**: Setzt den Socket in den Zustand, in dem er Verbindungsanfragen von Clients akzeptiert.
- **accept**: Wartet auf eingehende Verbindungsanfragen und erstellt neue Sockets für die Kommunikation mit den jeweiligen Clients.

3.

Pro IP-Adresse können theoretisch **65.536 TCP Server-Sockets** (Listen-Sockets) unterstützt werden, da es insgesamt 65.536 Portnummern gibt (von 0 bis 65.535). In der Praxis sind einige Ports reserviert oder von anderen Anwendungen belegt, daher ist die tatsächliche Anzahl etwas geringer. Ein Listen-Socket belegt eine Kombination aus IP-Adresse und Portnummer, und da Portnummern die begrenzende Ressource sind, ergibt sich diese Zahl.

4.

Der Mechanismus, der gewährleistet, dass an Host B keine Daten verloren gehen, ist das **TCP Receive Window (RWin)**. Das Receive Window begrenzt die Menge an Daten, die der Sender ohne Bestätigung vom Empfänger senden kann. Wenn Host B die Daten nur mit 50 Mb/s lesen kann, signalisiert es dies dem Sender durch ein entsprechendes RWin, das die Sende- und Empfangsrate synchronisiert und Datenverluste verhindert.

5.

Das TCP Receive Window (RWin) wird durch ein 16-bit Feld im TCP-Header angegeben, was ein maximales RWin von $2^{16} = 65.536$ Bytes bedeutet. Bei einer Round-Trip-Time (RTT) von 20 ms (0,02 Sekunden) lässt sich die maximale Übertragungsrate wie folgt berechnen:

$$\text{Maximale Übertragungsrate} = \frac{\text{Maximales RWin}}{\text{RTT}} = \frac{65.536 \text{ Bytes}}{0,02 \text{ Sekunden}} = 3.276.800 \text{ Bytes/Sekunde} \approx 3,28 \text{ MB/s}$$

Um die Rate in Bits pro Sekunde (bps) zu erhalten:

$$3,28 \text{ MB/s} \times 8 = 26.214.400 \text{ bps} = 26,21 \text{ Mbps}$$

6.

Die **TCP Window Scale Option** wurde eingeführt, um das maximale RWin über die 65.536 Bytes hinaus zu erweitern. Diese Option erlaubt eine linksschiebbare Skalierung des RWin-Werts um bis zu 14 Bit, was eine Vergrößerung des Empfangsfensters auf maximal $2^{16} \times 2^{14} = 2^{30} = 1.073.741.824$ Bytes ermöglicht. Die Skalierung erfolgt während des TCP Handshakes, wobei beide Endpunkte sich auf einen gemeinsamen Skalierungsfaktor einigen.

7.

Angenommen, der Empfänger signalisiert ein maximales RWin unter Verwendung der TCP Window Scale Option. Wenn der Skalierungsfaktor 14 ist, dann beträgt das maximale RWin:

$$\text{Maximales RWin} = 65.536 \text{ Bytes} \times 2^{14} = 1.073.741.824 \text{ Bytes}$$

Die maximale Übertragungsrate bei einer RTT von 20 ms ist:

$$\text{Maximale Übertragungsrate} = \frac{1.073.741.824 \text{ Bytes}}{0,02 \text{ Sekunden}} = 53.687.091.200 \text{ Bytes/Sekunde} \approx 53,69 \text{ GB/s}$$

Um die Rate in Bits pro Sekunde (bps) zu erhalten:

$$53,69 \text{ GB/s} \times 8 = 429.496.729.600 \text{ bps} = 429,50 \text{ Gbps}$$