

Lösungen Übungsblatt 3: TCP/IP-Architektur und Fehlerbehandlung

Übung

Aufgabe 1 TCP/IP-Architektur – Aufbau und Aufgaben der Schichten bzw. Protokolle

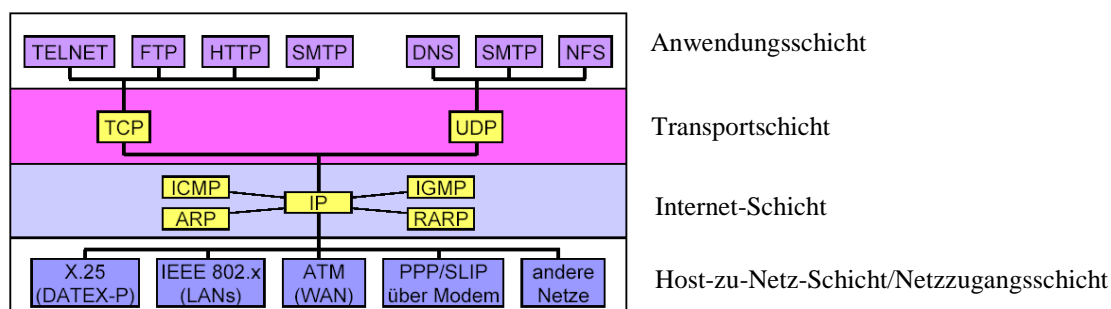
Die TCP/IP-Architektur ist im Gegensatz zum OSI-Referenzmodell keine Kommunikationsarchitektur, sondern eine Protokollarchitektur. So gibt es z. B. keine expliziten Dienstzugangspunkte (SAPs), sondern nur die standardisierten (bekannten) Schnittstellen der einzelnen Protokolle.

Skizzieren und erläutern Sie den Protokollstapel der TCP/IP-Architektur:

- Zeichnen und benennen Sie dazu die einzelnen Schichten und erläutern Sie deren Aufgaben.
- Nennen Sie für jede der Schichten (mehrere) konkrete Protokolle (z.B. TCP, SMTP, ...) und zeichnen Sie diese exemplarisch als Protokollinstanzen in die jeweilige Schicht der Skizze aus a).
- Stellen Sie die Schichten der TCP/IP-Architektur in der Skizze aus a) denen des OSI-Referenzmodells gegenüber. Welche Schichten des OSI-Modells erfüllen jeweils die gleichen Funktionen wie die Schichten von TCP/IP? Gibt es zusätzliche Schichten oder fehlen Schichten? Falls ja, welche?

Lösung:

a + b)



c) Aufgaben und Entsprechungen:

- Der **zentrale Punkt** der TCP/IP-Architektur ist die Internet-Schicht oder **IP-Schicht**. Die Aufgabe dieser Schicht ist die **Vermittlung der IP-Pakete**, also deren korrekte Zustellung, wofür ein geeigneter **Weg gefunden** werden muss. Da in der TCP/IP-Architektur – im Gegensatz zu OSI – von Anfang an auf die Verbindung unterschiedlicher Netztechnologien geachtet wurde, gibt es in dieser Schicht letztlich auch genau ein **Protokoll (IP)**, das **grundsätzlich für die Datenübertragung** verwendet wird. Die anderen eingezeichneten Protokolle sind **Hilfsprotokolle**. Somit entspricht die **IP-Schicht im wesentlichen der Vermittlungsschicht des OSI-Modells**.
 - Anmerkung: Die **Routing-Protokolle** gehören in der TCP/IP-Suite nicht zur Internet-Schicht. Statt dessen werden diese eher der **Anwendungsschicht** zugeordnet, auch weil sie ihrerseits IP und häufig auch TCP oder UDP verwenden. Das ist nicht zwangsläufig ein Bruch, da es sich ja im Prinzip um Steuerfunktionen handelt, die die Routing-Tabellen verändern. Die eigentliche Vermittlung der Pakete findet schließlich trotzdem in der Internet-Schicht statt.*
- Oberhalb der IP-Schicht befindet sich die **Transport-Schicht**. In dieser Schicht werden die allgemein verwendbaren Protokolle definiert, die für die **Ende-zu-Ende-Kommunikation** zwischen Sender- und Empfänger gedacht sind – im konkreten sind dies vor allem TCP (verbindungsorientiert) und UDP (verbindungslos). Mit TCP steht dabei Anwendungen oder Anwendern ein zuverlässiges Protokoll für die Datenübertragung zur Verfügung. Die **Transport-Schicht entspricht weitgehend der Transport-Schicht des OSI-Modells**.
- Über der Transportschicht befindet sich die **Anwendungsschicht**. Sie enthält alle Protokolle oberhalb der Transportschicht. Dies sind letztlich die **Protokolle, die eher für bestimmte Anwendungen** sinnvoll sind (HTTP, SMTP, ...), wobei DNS sicherlich nicht so sonderlich anwendungsspezifisch ist, sondern für sehr viele Anwendungen verwendet wird; möglicherweise hat man das am Anfang nicht so gesehen. Die Protokolle der Anwendungsschicht setzen dann wahlweise auf TCP oder UDP auf. Die Anwendungsschicht der

TCP/IP-Architektur entspricht im Wesentlichen der Anwendungsschicht des OSI-Modells. Wenn Funktionen der Schichten 5 und 6 nötig sind, sind diese natürlich auch in der Anwendungsschicht der TCP/IP-Architektur enthalten.

Unterhalb der IP-Schicht befindet sich die **Host-zu-Netz-Schicht**. Diese Schicht ist im Grunde nicht genau definiert. Es geht hier lediglich darum, dass ein Host über ein bestimmtes Protokoll zum Netz eine Verbindung aufbauen muss, um die IP-Pakete versenden zu können. Letztlich wird mit dieser Schicht also erreicht, dass IP tatsächlich auf *jedem Netz* aufbauen kann, wobei sich dann natürlich die entsprechenden Implementierungen dieser Schicht mehr oder weniger deutlich unterscheiden. Die Host-zu-Netz-Schicht entspricht in etwa den Schichten 1 und 2 des OSI-Modells. Es gab auch Beispiele, wo sie mehr als zwei Schichten umfasste.

- d) Offensichtlich ist es so, dass ab der Transport-Schicht die Schichten nur in den Endsystemen vorkommen. Die darunter liegenden Schichten müssen auch auf den Zwischensystemen implementiert sein.

Aufgabe 2 Grundlagen zur Fehlerbehandlung

a)

- PDU-Fehler: Signalstörung auf dem Kommunikationsmedium, starkes Rauschen
- PDU-Verlust: Ausfall/Überlastung von Zwischensystemen, Verwerfen der PDU (untere Schichten!) aufgrund von erkannten Bitfehlern
- PDU-Dopplung: Erneutes Senden aufgrund von zu hoher Latenz oder Verlust der Bestätigung

b)

- Forward Error Correction (FEC)
 - u. U. ist es möglich, aufgetretene Fehler in angekommenen Daten zu erkennen und zu beheben
 - dafür ist Redundanz in der Übertragung erforderlich
 - d. h. es müssen immer – ohne dass klar ist, ob ein Fehler auftritt oder nicht – zusätzliche Daten geschickt werden, mit denen möglicherweise Fehler behoben werden können
 - theoretisch ist FEC sowohl für Paketbeschädigungen als auch Paketverluste möglich
 - in letzterem Fall müssen die anderen Pakete Infos über das fehlende Paket beinhalten
 - im ersteren Fall ist es alternativ möglich, dass das Paket über sich selbst redundante Informationen enthält
 - **Nachteile:**
 - es werden mehr Daten gesendet als eigentlich nötig
 - **Vorteil:**
 - u. U. Zeitvorteil, gerade bei Echtzeitströmen sehr vorteilhaft !!!
 - auch möglich, wenn keine Rückkopplung vom Empfänger zum Sender möglich

c)

Rückkopplung:

- ist FEC nicht im Einsatz oder versagt im konkreten Fall, dann müssen Daten irgendwie erneut gesendet werden
- dafür ist irgendwie Rückkopplung vom Empfänger zum Sender nötig
- wie soll die Rückkopplung aussehen? **WIE?**
 - **Fehlermeldung? Fehlerkanal?**
 - i. woher weiß Empfänger, dass ein Paket fehlt?
 - ii. in unserem Fall: evt. durch fehlende Sequenznummer, falls Pakete durchnummeriert (sie werden durchnummeriert ...)
 - iii. was aber wäre, wenn letzte Paket verloren geht?
 - Fehlermeldung nicht in jedem Fall möglich (wird aber tw. zusätzlich eingesetzt; im folgenden nicht diskutiert)
 - Bestätigung des Empfangs **WAS wird bestätigt**
 - iv. Empfänger bestätigt die korrekt empfangenen Pakete

v. aber: Bestätigung kann verloren gehen

→ Timer beim Sender

- was passiert nun bei ausbleibender Bestätigung? **Korrektur**
 - Wiederholen des Sendens

d)

Selective Repeat

- die naheliegendste Variante ist vermutlich, nur das beschädigte (oder verlorengegangene) Paket neu zu übertragen
- Konsequenz:
 - es werden nur beschädigte Pakete erneut übertragen
 - auf der **Senderseite** müssen natürlich **alle Pakete bis zu ihrer Bestätigung zwischengespeichert** werden
→ um sie ggf. erneut übertragen zu können
 - **der Empfänger** muss **Pakete nach den verlorengegangenen Paketen zwischenspeichern**
→ es muss überhaupt erst einmal eine Pufferverwaltung auf der Empfängerseite geben!
→ andererseits können auf der Sender-Seite die erfolgreich empfangenen Pakete bereits entfernt werden

Go-Back-N

- alle außer das als nächstes zu erwartende Paket werden verworfen
 - **andere Pakete werden also auch nicht bestätigt, selbst wenn sie erfolgreich empfangen wurden**
- Konsequenzen:
 - Pakete werden auch bei erfolgreicher Übertragung mehrfach übertragen
 - **Sender muss wie beim Selective Repeat zunächst alle Pakete speichern**
 - auf **Empfängerseite keine Pufferverwaltung** nötig – und keine Umsortierung!

e)

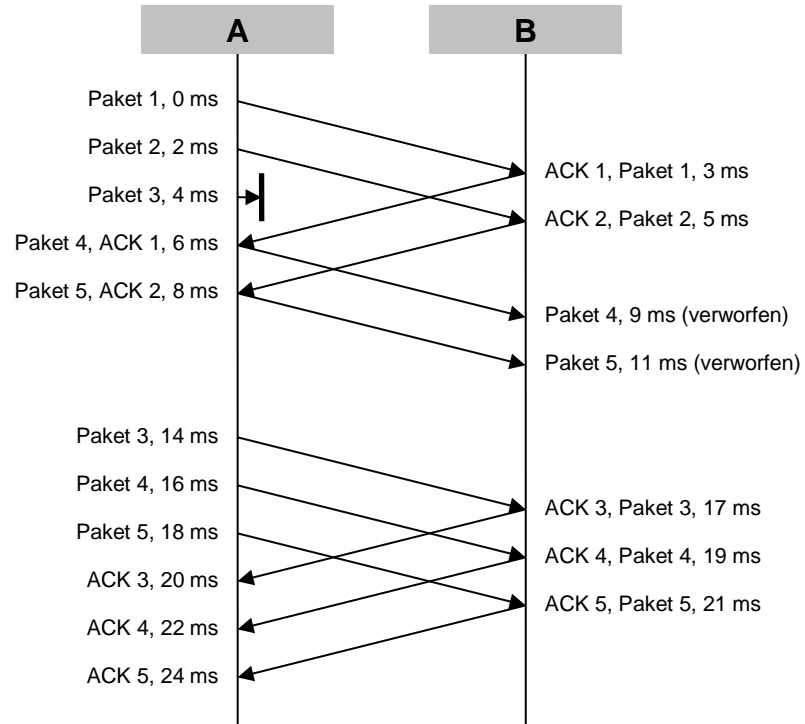
- Beim **Go-Back-N-Verfahren** ist es eigentlich sinnlos, mit individuellen Bestätigungen zu arbeiten. Schließlich wird der Empfänger ein Paket x nur dann bestätigen, wenn er Paket $(x-1)$ empfangen hat. Demzufolge kann der Sender ankommende Bestätigungen immer als **kumulative Bestätigungen** auffassen. Die einzige Ausnahme könnte sein, wenn der Sender nicht wüsste, ob der Empfänger Go-Back-N oder Selective-Repeat einsetzt.
- Beim **Selective-Repeat-Verfahren** sind prinzipiell **beide Verfahren möglich**. Individuelle Bestätigungen sind sicherlich zunächst nahe liegend, **bei geringen Fehlerraten werden jedoch häufig auch kumulative Bestätigungen verwendet**. Um für diesen Fall zu vermeiden, dass der Sender nach Ablauf des Timers doch wieder alle Pakete schickt, wird dann meist mit **negativen Bestätigungen** gearbeitet (implizit (bei TCP) oder explizit).

f)

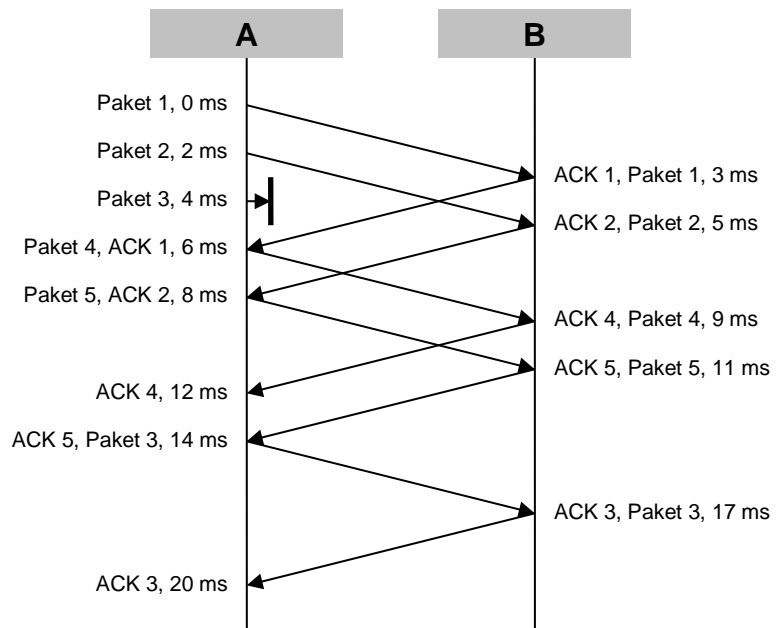
- Geht eine Bestätigung verloren, dann muss der **Sender annehmen, dass das Paket verlorengegangen** ist. Dafür werden **Timer** eingesetzt, um **das Ausbleiben der Bestätigung** zu erkennen. Im Fall von Go-Back-N könnte der Sender natürlich durch das Erhalten von Bestätigungen für darauf folgende Pakete darauf schließen, dass doch nur die Bestätigung verloren gegangen ist. (Bei Go-Back-N können Bestätigungen also immer kumulativ sein!)
- Letztlich könnte es aber sowohl bei Selective Repeat als auch Go-Back-N passieren, dass für sämtliche Sequenznummern eines Sliding Windows die Bestätigungen verloren gegangen sind. Dann würde der Sender zwar diese Pakete wiederholen, aber der Empfänger würde sie vielleicht einfach verwerfen. Wenn er das wirklich tun würde, dann würde der Sender vielleicht nie wieder eine Bestätigung für ein scheinbar fehlendes Paket bekommen würde. Um dies zu vermeiden, sollten **Empfänger Bestätigungen ggf. wiederholen, bei kumulativen Bestätigungen am besten mit der höchsten Nummer**, bis zu der alle Pakete empfangen wurden.

Aufgabe 3 ARQ-Verfahren – Selective Repeat vs. Go-Back-N

a) Go-Back-N:

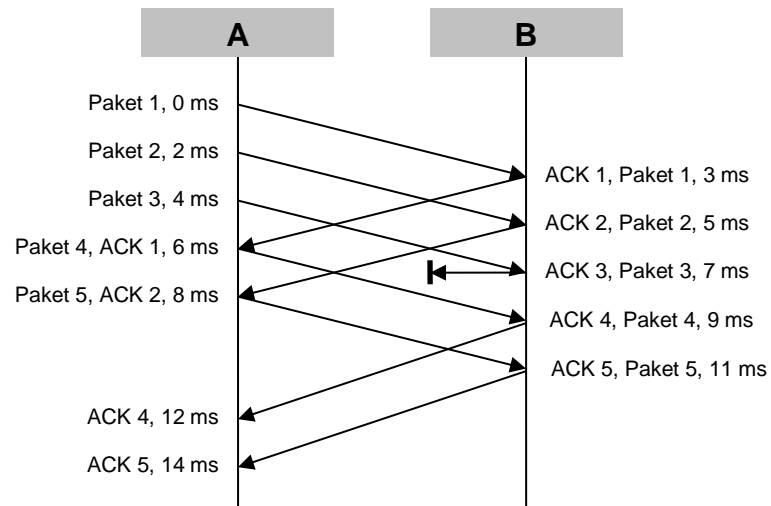


Selective-Repeat:



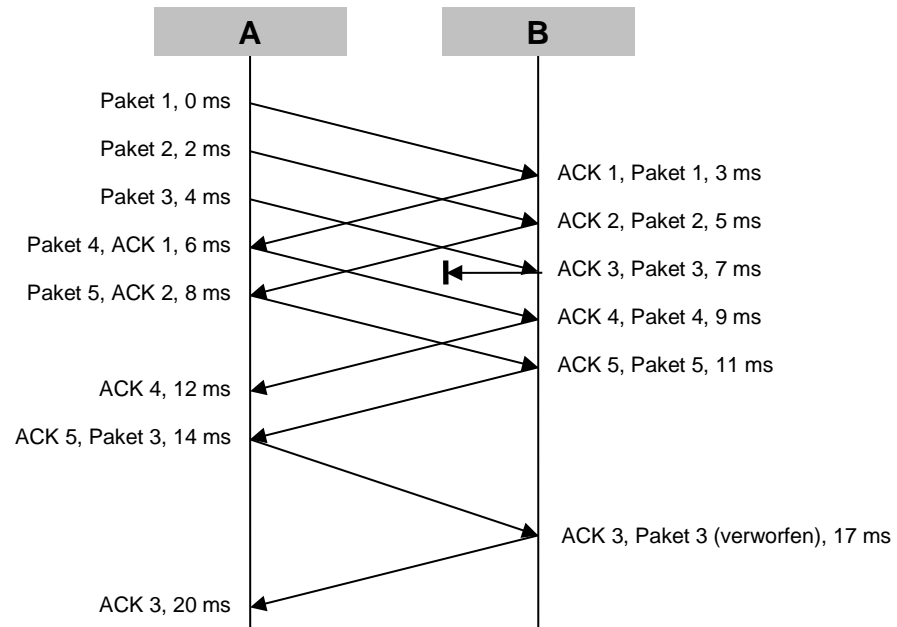
Aus der Aufgabenstellung ergibt sich eindeutig, dass es sich beim Selective-Repeat-Verfahren um individuelle Bestätigungen handelt.

b) Go-Back-N:

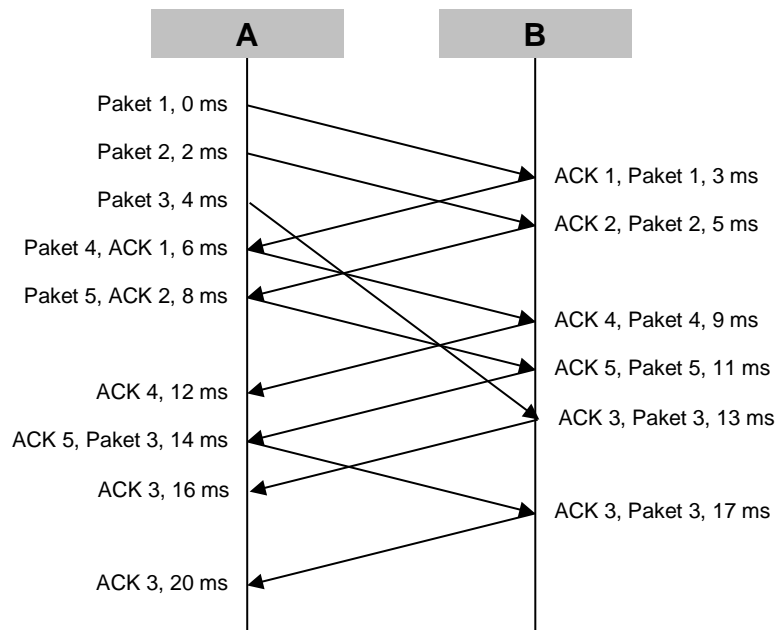


Die Lösung ist eigentlich eindeutig, da beim Go-Back-N-Verfahren Bestätigungen immer als kumulativ aufgefasst werden können – und daher auch sollten.

Selective-Repeat:



Aus der Aufgabenstellung ergibt sich eindeutig, dass es sich beim Selective-Repeat-Verfahren um individuelle Bestätigungen handeln muss.



Selbststudium

Aufgabe 4: OSI-Referenzmodell

Dazu bitte das Vorlesungskapitel 2.4 anschauen.

Aufgabe 5: Kommunikation mit Pluto-Sonde

FEC

- Zunächst einmal ist festzustellen, dass die RTT für diesen Fall sehr groß ist (mehrere Stunden). Dies hat zur Folge, dass relativ viele Daten zwischengespeichert werden müssen, bevor eine Rückkopplung eintreffen kann. Dies spricht zunächst vielleicht für die FEC und gegen ARQ-Verfahren. Für FEC spricht zusätzlich, dass es natürlich relativ schwierig ist, mit geringer Leistung durch die Atmosphäre zuverlässig zu senden (ähnliche Probleme wie beim WLAN). Insofern kann FEC durchaus als empfehlenswert angesehen werden.

ARQ

- Neben Verfahren der FEC gibt es auch die ARQ-Verfahren, wobei zwischen Go-Back-N und Selective Repeat unterschieden wird. Die RTT ist zwar relativ hoch, spricht aber nur gegen ARQ-Verfahren, wenn die Daten nicht ohnehin zwischengespeichert werden müssen. Ist dies nicht der Fall, dann ist der Nutzen von ARQ-Verfahren auf jeden Fall zweifelhaft.
- Allerdings fallen die Bilder nur in der Nähe von Himmelskörpern an – und die Bilder sollen in guter Qualität zur Erde gesendet werden. Damit die Sonde dennoch mit begrenzter Übertragungskapazität auskommen kann, werden deshalb häufig die Bild-Daten zwischengespeichert, um sie nach dem Passieren eines Himmelskörpers kontinuierlich mit relativ geringer Datenrate an die Erde senden zu können. Auf diese Weise kann Gewicht für die Kommunikationstechnik gespart werden.
- Wenn die Daten aber sowieso zwischengespeichert werden, können auch ARQ-Verfahren zum Einsatz kommen. Sinnvoll ist dabei aber nur Selective-Repeat, denn es ist wenig sinnvoll, auf der Erde bereits angekommene Pakete zu verwerfen. (Go-Back-N hat nur den Vorteil, den Empfänger einfacher zu halten; dies ist hier aber wahrlich nicht notwendig.) Gegen ein zu umfangreiches FEC spricht beim Einsatz von Selective-Repeat dann natürlich die benötigte zusätzliche Übertragungskapazität.

Fazit: Go-Back-N wäre definitiv nicht zu empfehlen, Selective Repeat und FEC könnten sinnvoll sein.