Computernetzwerke

6., aktualisierte Auflage

Andrew S. Tanenbaum
Nick Feamster
David J. Wetherall

Pearson

EXTRAS ONLINE

Sublayer bezeichnet), und der MAC-Schicht (*Media Access Control*), auf die wir in *Kapitel 4* näher eingehen werden. Über der Bitübertragungsschicht muss DOCSIS eine Reihe von Aufgaben für die Vermittlungsschicht übernehmen, darunter die Bandbreitenzuweisung für Upstream- und Downstream-Übertragungen (Flusskontrolle), Rahmenbildung und Fehlerkorrektur (manchmal wird Fehlerkorrektur auch als Konstrukt der Bitübertragungsschicht betrachtet). Wir haben diese Konzepte weiter oben in diesem Kapitel bereits beschrieben, wollen in diesem Abschnitt aber untersuchen, wie DOCSIS jedes dieser Probleme angeht.

Ein DOCSIS-Rahmen enthält verschiedene Informationen, darunter Indikatoren der Dienstqualität und Unterstützung für die Fragmentierung oder Verkettung von Rahmen. Jede unidirektionale Folge von Rahmen wird als **Dienstfluss** (*service flow*) bezeichnet. Die primären Dienstflüsse ermöglichen es dem CMTS im Kabelunternehmen, Verwaltungsnachrichten an jedes Kabelmodem zu übermitteln. Jeder Dienstfluss hat eine eindeutige Kennung und ist oft mit einer bestimmten Dienstklasse verbunden, z.B. Best Effort, Polling (d.h., ein Kabelmodem fordert explizit Bandbreite an) und Grant Service (d.h., ein Kabelmodem überträgt Daten-Bursts zu einer garantierten Zeit). Ein primärer Dienstfluss ist der Standarddienstfluss, der alle Rahmen überträgt, die keinem anderen Dienstfluss zugewiesen sind. In den vielen Breitbanddienstkonfigurationen gibt es standardmäßig nur einen Upstream- und einen Downstream-Dienstfluss zwischen dem CM und dem CMTS, der den gesamten Benutzerverkehr sowie alle Verwaltungsnachrichten umfasst. DOCSIS-Netze wurden ursprünglich unter der Annahme entworfen, dass die meisten Daten in Downstream-Richtung übertragen werden. Bestimmte Anwendungen, wie z.B. Videokonferenzen, laufen diesen Trends zuwider, auch wenn kürzlich angekündigte Cloud-Gaming-Dienste (z.B. Stadia, GeForce Now, xCloud) eventuell zu einer noch stärkeren Downstream-Nutzung führen, da diese Anwendungen auf kontinuierliche Streaming-Raten von 30–35 Mbit/s abzielen.

Sobald ein Kabelmodem eingeschaltet ist, stellt es eine Verbindung zum CMTS her, wodurch es sich in der Regel mit dem restlichen Netz verbindet. Nach der Registrierung beim CMTS erhält es von ihm die zu verwendenden Upstream- und Downstream-Kommunikationskanäle sowie die Verschlüsselungsschlüssel. Die Upstream- und Downstream-Netzbetreiber stellen zwei gemeinsame Kanäle für alle Kabelmodems bereit. In Downstream-Richtung empfangen alle an das CMTS angeschlossenen Kabelmodems jedes übertragene Paket. In Upstream-Richtung senden viele Kabelmodems und das CMTS ist der einzige Empfänger. Zwischen dem CMTS und den einzelnen Kabelmodems können mehrere physikalische Pfade bestehen.

Vor DOCSIS 3.1 wurden Pakete in Downstream-Richtung in 188-Byte-MPEG-Rahmen mit jeweils einem 4-Byte-Header und einer 184-Byte-Nutzlast unterteilt (die sogenannte MPEG-Übertragungskonvergenzschicht). Zusätzlich zu den eigentlichen Daten sendet das CMTS in regelmäßigen Abständen Verwaltungsinformationen an die Kabelmodem, die Informationen über das Ranging, die Kanalzuweisung und andere Aufgaben im Zusammenhang mit der Kanalzuordnung enthalten, die von der MAC-Schicht durchgeführt werden (auf die wir in *Kapitel 4* näher eingehen werden). DOC-

SIS 3.1 unterstützt diese Konvergenzschicht zwar immer noch aus Gründen der Abwärtskompatibilität, stützt sich aber nicht darauf für die Downstream-Kommunikation.

Die DOCSIS-Sicherungsschicht organisiert die Übertragung anhand von **Modulationsprofilen**. Ein Modulationsprofil ist eine Liste von Modulationsanweisungen (d.h. Bit-Loadings), die den OFDM-Unterträgern entsprechen. In Downstream-Richtung kann das CMTS unterschiedliche Profile für verschiedene Kabelmodems verwenden, aber in der Regel wird eine Gruppe von Kabelmodems, die die gleiche oder eine ähnliche Leistung aufweisen, in demselben Profil zusammengefasst. Auf der Grundlage der Dienstflussidentifikation und der QoS-Parameter gruppiert die Sicherungsschicht (in DOCSIS 3.1), die jetzt als **Konvergenzschicht** bezeichnet wird, Pakete mit demselben Profil in denselben Sendepuffer. In der Regel gibt es einen Sendepuffer pro Profil, der jeweils flach ist, um hohe Latenzzeiten zu vermeiden. Der Codeword Builder ordnet dann jeden DOCSIS-Rahmen den entsprechenden FEC-Codewörtern zu, wobei er Pakete aus verschiedenen Profilpuffern nur an jeder Codewort-Grenze abruft. Bei der FEC-Codierung wird der DOCSIS-Rahmen als Bitstrom und nicht als Bytefolge betrachtet. DOCSIS stützt sich auf ein LDPC-Codewort. In Downstream-Richtung hat ein vollständiges Codewort bis zu 2.027 Bytes, von denen bis zu 1.799 Bytes Daten und 225 Bytes Parität sind. Innerhalb jedes Bytes eines DOCSIS-Rahmens wird das niederwertigste Bit zuerst übertragen; wenn ein Wert von mehr als einem Byte übertragen wird, werden die Bytes vom höherwertigen zum niederwertigen geordnet, eine Ordnung, die manchmal auch als **Netzordnung** bezeichnet wird. Das CMTS wendet auch Byte Stuffing an: Wenn kein DOCSIS-Rahmen in Downstream-Richtung verfügbar ist, fügt das CMTS mit Null-Bit gefüllte Unterträger in OFDM-Symbole ein oder fügt einfach Folgen von Einsen in Codewörter ein, wie in ▶Abbildung 3.28 gezeigt.

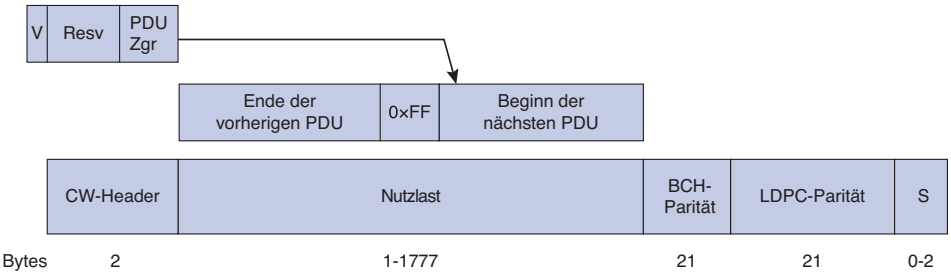

Abbildung 3.28: DOCSIS-Rahmen an Codewort-Abbildung.

Seit Version 3.0 unterstützt DOCSIS eine Technologie namens **Kanalbündelung** (*channel bonding*), die es einem einzelnen Teilnehmer ermöglicht, mehrere Upstream- und Downstream-Kanäle gleichzeitig zu nutzen. Diese Technologie ist eine Form der **Verbindungsaggregation** (*link aggregation*), bei der mehrere physische Verbindungen oder Ports zu einer einzigen logischen Verbindung kombiniert werden können. Mit DOCSIS 3.0 können bis zu 32 Downstream-Kanäle und 8 Upstream-Kanäle gebündelt werden, wobei jeder Kanal 6–8 MHz breit sein kann. Die Kanalbündelung in DOCSIS 3.1 unterscheidet sich von der in DOCSIS 3.0 nur insofern, als DOCSIS 3.1 breitere

Upstream- und Downstream-Kanäle unterstützt: Diese können mit bis zu 192 MHz im Downstream und 96 MHz im Upstream viel breiter sein im Vergleich zu DOCSIS 3.0 mit seinen 6 oder 8 MHz im Downstream und bis zu 6,4 MHz im Upstream. Andererseits kann ein DOCSIS-3.1-Modem Kanäle verschiedener Typen bündeln (z.B. könnte ein DOCSIS-3.1-Modem einen 192-MHz-OFDM-Kanal und vier 6-MHz-SC-QAM-Kanäle bündeln).

Zusammenfassung

Die **Sicherungsschicht** hat die Aufgabe, den reinen Bitstrom aus der Bitübertragungsschicht für die Weiterverarbeitung durch die Vermittlungsschicht in einen Rahmenstrom zu konvertieren. Die Sicherungsschicht kann diesen Strom mit unterschiedlichen Zuverlässigkeitsebenen anbieten, vom verbindungslosen unbestätigten Dienst bis hin zum zuverlässigen verbindungsorientierten Dienst.

Bei der **Rahmenbildung** gibt es verschiedene Methoden, z. B. Bytezählverfahren, Byte Stuffing und Bit Stuffing. Protokolle der Sicherungsschicht implementieren Fehlerkontrollmethoden, sodass beschädigte Rahmen erkannt und korrigiert oder verlorene Rahmen erneut übertragen werden können. Um einen schnellen Sender davon abzuhalten, einen langsamen Empfänger mit Daten zu überschütten, setzen die Protokolle der Sicherungsschicht Flusskontrollmechanismen ein. Mit dem Schiebefenstermechanismus können Fehler- und Flusskontrolle einfach koordiniert werden. Wenn die Fenstergröße 1 Paket beträgt, dann wird ein Stop-and-Wait-Protokoll verwendet.

Codes zur **Fehlerkorrektur** und -entdeckung fügen den Nachrichten redundante Information hinzu, indem eine Reihe von mathematischen Techniken verwendet wird. Faltungscodes und Reed-Solomon-Codes werden zur Fehlerkorrektur weitverbreitet eingesetzt, wobei Paritätsprüfcodes mit geringer Dichte zunehmend beliebter werden. Zu den Codes zur Fehlererkennung, die in der Praxis eingesetzt werden, gehören zyklische Redundanzprüfung und Prüfsummen. Diese Codes können sowohl in der Sicherungsschicht als auch in der Bitübertragungsschicht und höheren Schichten eingesetzt werden.

Wir haben verschiedene **Protokolle** untersucht, die eine zuverlässige Sicherungsschicht mithilfe von Bestätigungen und Neuübertragungen bzw. – unter realistischeren Annahmen – ARQ (*Automatic Repeat reQuest*) unterstützen. Angefangen bei einer fehlerfreien Umgebung, in der der Empfänger jeden an ihn gesendeten Rahmen verarbeiten kann, haben wir Flusskontrolle eingeführt, gefolgt von Fehlerkontrolle mit Sequenznummern und den Stop-and-Wait-Algorithmus. Dann haben wir den Schiebefensteralgorithmus verwendet, um bidirektionale Kommunikation zu ermöglichen, und haben das Konzept des Huckepacktransports eingeführt. Die letzten beiden Protokolle leiten die Übertragung von mehreren Rahmen, um den Sender davor zu bewahren, auf einer Verbindung mit einer langen Übertragungsverzögerung blockiert zu werden. Der Empfänger kann entweder alle Rahmen bis auf den Rahmen verwerfen, der in der Folge der nächste wäre, oder die Rahmen zwischenspeichern, die außerhalb der Reihenfolge ankommen, und negative Bestätigungen senden, um die Bandbreiteneffizienz zu erhöhen. Die erste Strategie ist ein Go-Back-N-Protokoll und die zweite ein Protokoll mit selektiver Wiederholung.

Das **Internet** verwendet vorwiegend PPP als Schicht-2-Protokoll auf Punkt-zu-Punkt-Leitungen. PPP stellt einen verbindungslosen unbestätigten Dienst zur Verfügung, benutzt Flagbytes zur Rahmenbegrenzung und ein CRC zur Fehlererkennung. PPP wird zur Paketübertragung für eine Vielzahl von Verbindungen benutzt, dazu gehören SONET-Verbindungen in WANs und ADSL-Verbindungen für Privathaushalte. DOCSIS wird verwendet, wenn der Internetdienst über das bestehende Kabel-TV-Netz angeboten wird.

Übungsaufgaben

EXTRAS ONLINE
Lösungshinweise

1 Ethernet verwendet eine Präambel mit einer Bytezählung, um die Rahmen zu trennen. Was geschieht, wenn ein Benutzer versucht, Daten zu senden, die diese Präambel enthalten?

2 In einem Schicht-2-Protokoll wird die folgende Zeichencodierung verwendet:

A.: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000

Zeigen Sie, welche Bitfolge (binär) für den folgenden vier Zeichen langen Rahmen übertragen wird: A B ESC FLAG, wenn die folgenden Rahmenbildungsmethoden verwendet werden:

a. Bytezählung

b. Flagbytes mit Byte Stuffing

c. Start- und Endflagbytes mit Bit Stuffing?

3 Das folgende Datenfragment erscheint inmitten eines Datenflusses, für den der im Text beschriebene Byte-Stuffing-Algorithmus verwendet wird: A B ESC C ESC FLAG FLAG D. Welches Ergebnis erhält man nach dem Auffüllen?

4 Welches ist der maximale Overhead beim Byte-Stuffing-Algorithmus?

5 Sie empfangen das folgende Datenfragment: A ESC FLAG A B A FLAG FLAG C B ESC FLAG ESC ESC ESC FLAG FLAG. Sie wissen, dass das Protokoll Byte Stuffing verwendet. Zeigen Sie den Inhalt von jedem Rahmen nach dem „Entstopfen".

6 Sie empfangen das folgende Datenfragment: 0110 0111 1100 1111 0111 1101. Sie wissen, dass das Protokoll Bit Stuffing verwendet. Zeigen Sie den Inhalt von jedem Rahmen nach dem „Entstopfen".

7 Einer Ihrer Kommilitonen, Dagobert, hat darauf hingewiesen, dass es Verschwendung ist, jeden Rahmen mit einem Flagbyte zu beenden und dann den nächsten wieder mit einem zweiten Flagbyte zu beginnen. Dies könnte man auch mit einem Flagbyte erledigen und sich das zweite Flagbyte sparen. Stimmen Sie dem zu?

8 Die Bitfolge 0111101111101111110 muss über die Sicherungsschicht übertragen werden. Welche Bitfolge wird nach dem Bit Stuffing tatsächlich übertragen?

9 Ein Paket der oberen Schicht wird in 10 Rahmen aufgeteilt, die jeweils eine 80%ige Chance haben, unbeschädigt anzukommen. Wie oft muss die Nachricht im Durchschnitt gesendet werden, um das ganze Paket zu übertragen, wenn das Protokoll der Sicherungsschicht keine Fehlerkontrolle bietet?

10 Können Sie sich Situationen vorstellen, in denen Protokolle mit offenen Schleifen (z. B. ein Hamming-Code) besser geeignet sind als die in diesem Kapitel beschriebenen Protokolle mit Bestätigungsfunktion?

11 Um eine höhere Zuverlässigkeit bereitzustellen, als ein einzelnes Paritätsbit bieten kann, verwendet ein Codierschema zur Fehlererkennung ein Paritätsbit zum Überprüfen aller Bits mit ungerader und ein zweites Paritätsbit für alle Bits mit gerader Nummer. Wie lautet der Hamming-Abstand dieses Codes?

12 16-Bit-Nachrichten werden unter Verwendung eines Hamming-Codes übertragen. Wie viele Prüfbits sind erforderlich um sicherzustellen, dass der Empfänger Einzelbitfehler entdecken und korrigieren kann? Geben Sie das übertragene Bitmuster für die Nachricht 1101001100110101 an. Gehen Sie davon aus, dass im Hamming-Code die gerade Parität verwendet wird.

13 Ein 8-Bit-Byte mit dem Binärwert 10101111 soll mit einem Hamming-Code gerader Parität codiert werden. Wie lautet der Binärwert nach der Codierung?

14 Ein 12-Bit-Hamming-Code mit dem hexadezimalen Wert 0xE4F kommt beim Empfänger an. Wie lautet der Ursprungswert in hexadezimaler Darstellung? Gehen Sie davon aus, dass nicht mehr als ein Bit fehlerhaft ist.

15 Eine Möglichkeit der Erkennung von Fehlern ist, Daten als einen Block mit n Zeilen und k Bit pro Zeile zu übertragen und jeder Zeile und jeder Spalte ein Paritätsbit anzufügen. Das Bit rechts unten ist ein Paritätsbit, das seine Zeile und seine Spalte prüft. Können mit diesem Schema alle Einzelfehler aufgedeckt werden? Doppelte Fehler? Dreifache Fehler? Zeigen Sie, dass dieses Schema einige Vierbitfehler nicht entdecken kann.

16 Angenommen, Daten werden in Blöcken von 1.000 Bits übertragen. Welches ist die maximale Fehlerrate, bei der Fehlererkennungs- und Neuübertragungsmechanismen (1 Paritätsbit pro Block) besser als die Benutzung von Hamming-Code sind? Setzen Sie voraus, dass Bitfehler unabhängig voneinander sind und kein Bitfehler während einer Neuübertragung auftritt.

17 Ein aus n Zeilen und k Spalten bestehender Bitblock benutzt horizontale und vertikale Paritätsbits zur Fehlererkennung. Nehmen Sie an, dass aufgrund von Übertragungsfehlern genau vier Bit invertiert werden. Leiten Sie einen Ausdruck für die Wahrscheinlichkeit ab, dass dieser Fehler nicht entdeckt wird.

18 Welche Ausgabefolge wird bei Anwendung des Faltungscodes von Abbildung 3.7 erzeugt, wenn die Eingabefolge 10101010 (von links nach rechts) ist und der interne Status anfangs aus lauter Nullen besteht

19 Nehmen Sie an, dass die Nachricht 1001 1100 1010 0011 unter Benutzung der Internetprüfsumme (4-Bit-Wort) übertragen wird. Welchen Wert hat die Prüfsumme?

20 Welchen Rest erhält man, wenn man x^7+x^5+1 durch das Generatorpolynom x^3+1 teilt?

21 Der Bitstrom 10011101 wird mit der im Text beschriebenen CRC-Standardmethode übertragen. Das Generatorpolynom ist x^3+1. Geben Sie die tatsächlich übertragene Bitfolge an. Gehen Sie davon aus, dass das dritte Bit von links bei der Übertragung invertiert wird. Zeigen Sie, dass dieser Fehler vom Empfänger entdeckt wird. Geben Sie ein Beispiel von Bitfehlern in der übertragenen Bitfolge, die nicht vom Empfänger entdeckt werden.

22 Eine 1.024-Bit-Nachricht, die aus 992 Datenbits und 32 CRC-Bits besteht, wird gesendet. Der CRC-Wert wird mithilfe des durch IEEE 802 standardisierten Polynoms vom Grad 32 berechnet. Erklären Sie für jede der folgenden Situationen, ob die Fehler während der Nachrichtenübertragung vom Empfänger entdeckt werden.

a. Es gab einen Einzelbitfehler.

b. Es gab zwei isolierte Bitfehler.

c. Es gab 18 isolierte Bitfehler.

d. Es gab 47 isolierte Bitfehler.

e. Es gab einen 24 Bit langen Burstfehler.

f. Es gab einen 35 Bit langen Burstfehler.

23 Bei der Besprechung des ARQ-Protokolls in Abschnitt Simplexprotokolle der Sicherungsschicht wurde ein Szenario entworfen, das damit endete, dass der Empfänger zwei Kopien desselben Rahmens akzeptierte, weil der Bestätigungsrahmen verloren ging. Ist es möglich, dass ein Empfänger mehrere Kopien desselben Rahmens auch dann akzeptiert, wenn keiner der Rahmen (Nachrichten oder Bestätigung) verloren wurde?

24 Ein Kanal hat eine Übertragungsgeschwindigkeit von 4 kbit/s und eine Übertragungszeit von 20 ms. Bei welchen Rahmengrößen ergibt das Stop-and-Wait-Protokoll eine Effizienz von mindestens 50%?

25 Kann ein Sender in Protokoll 3 den Timer starten, wenn dieser bereits läuft? Falls ja, warum kann es dazu kommen? Wenn nicht, warum ist dies unmöglich?

26 Über eine 3.000 km lange T1-Leitung werden mit Protokoll 5 64-Byte-Rahmen übertragen. Wie viele Bits sollte die Sequenznummer haben, wenn die Übertragungsgeschwindigkeit 6 s/km beträgt?

27 Nehmen Sie ein Schiebefensterprotokoll, das so viele Bits für Sequenznummern verwendet, dass die Sequenznummern nie wieder bei 0 anfangen müssen. Welche Beziehungen müssen zwischen den vier Fensterenden und der Fenstergröße bestehen, die konstant und für Sender und Empfänger gleich sei?

28 Würden sich Auswirkungen auf die Richtigkeit des Protokolls oder seine Effizienz ergeben, wenn die Prozedur *between* in Protokoll 5 die Bedingung $a \leq b \leq c$ anstatt der Bedingung $a \leq b < c$ überprüfen würde? Erläutern Sie Ihre Antwort.

29 In Protokoll 6 wird bei Ankunft eines Datenrahmens geprüft, ob die Sequenznummer von der erwarteten abweicht und *no_nak* wahr (*true*) ist. Falls beide Bedingungen zutreffen, wird ein NAK gesendet. Andernfalls wird der Hilfstimer gestartet. Angenommen, die `else`-Klausel würde weggelassen. Würde sich diese Änderung auf die Korrektheit des Protokolls auswirken?

30 Angenommen, die `while`-Schleife mit drei Anweisungen am Ende von Protokoll 6 würde entfernt. Wirkt sich dies auf die Korrektheit des Protokolls aus oder nur auf die Leistung? Erläutern Sie Ihre Antwort.

31 Die Entfernung von der Erde zu einem entfernten Planeten beträgt ungefähr 9×10^{10} m. Wie groß ist die Kanalauslastung, wenn ein Stop-and-Wait-Protokoll zur Rahmenübertragung auf einer Punkt-zu-Punkt-Verbindung mit 64 Mbit/s benutzt wird? Nehmen Sie an, dass die Rahmengröße 32 KB beträgt, die Lichtgeschwindigkeit ist 3×10^{8} m/s.

32 Nehmen Sie an, dass in der vorherigen Aufgabe anstelle von Stop-and-Wait ein Schiebefensterprotokoll benutzt wird. Für welche gesendete Fenstergröße beträgt die Verbindungsausnutzung 100%? Sie dürfen die Protokollverarbeitungszeiten beim Sender und Empfänger vernachlässigen.

33 In Protokoll 6 hat der Code für *frame_arrival* einen Abschnitt für NAKs. Dieser Abschnitt wird aufgerufen, wenn der ankommende Rahmen ein NAK ist und eine weitere Bedingung erfüllt wird. Beschreiben Sie einen Ablauf, bei dem die Erfüllung dieser anderen Bedingung wichtig ist.

34 Betrachten Sie den Betrieb von Protokoll 6 über eine perfekte (d.h. fehlerfreie) 1-Mbit/s-Leitung. Die maximale Rahmengröße ist 1.000 Bit. Neue Pakete werden im Abstand von einer Sekunde erzeugt. Das Timeout-Intervall steht auf 10 ms. Würde man den speziellen Bestätigungstimer weglassen, entstünden unnötige Timeouts. Wie oft würde die durchschnittliche Nachricht übertragen werden?

35 In Protokoll 6 gilt $MAX\_SEQ = 2^n - 1$. Diese Bedingung ist sicher wünschenswert, um Header-Bits effizient zu verwenden. Wir haben aber nicht gezeigt, dass sie notwendig ist. Funktioniert das Protokoll noch richtig mit beispielsweise $MAX\_SEQ = 4$?

36 Rahmen mit 1.000 Bit werden über einen 1-Mbit/s-Kanal über einen geostationären Satelliten versendet, dessen Übertragungszeit von der Erde weg 270 ms beträgt. Die Bestätigungen erfolgen immer huckepack auf Datenrahmen. Die Header sind sehr kurz. Es werden 3-Bit-Sequenznummern verwendet. Welche maximale Kanalauslastung ergibt sich bei

a. Stop-and-Wait?

b. Protokoll 5?

c. Protokoll 6?

37 Betrachten Sie ein Protokoll, das Huckepacktransport, eine Sendefenstergröße von 4 und 400-Bit-Rahmen verwendet. Dieses Protokoll dient dazu, Daten über einen 200-kbit/s-Kanal mit einer unidirektionalen Übertragungsverzögerung von 4 ms zu übertragen. Leider hat der Empfänger keine Daten zum Zurücksenden. Er muss seine Bestätigungen in eigenen Rahmen senden. Wie lange kann der Empfänger vor dem Senden maximal warten, sodass die Bandbreiteneffizienz nicht unter 50% fällt?

38 Berechnen Sie den Teil der Bandbreite, der für den zusätzlichen Overhead (Header und erneute Übertragungen) von Protokoll 6 auf einem stark benutzen 50-kbit/s-Satellitenkanal verschwendet wird. Es werden Datenrahmen mit 40 Header- und 3.960 Datenbits verwendet. Gehen Sie davon aus, dass die Signalübertragungszeit von der Erde zum Satelliten 270 ms beträgt. ACK-Rahmen kommen nicht vor. NAK-Rahmen bestehen aus 40 Bits. Die Fehlerrate liegt für Datenrahmen bei 1% und die von NAK-Rahmen ist verschwindend gering. Die Sequenznummern bestehen aus 8 Bits.

39 Betrachten Sie einen fehlerfreien Satellitenkanal mit 64 kbit/s zur Übertragung von Rahmen mit 512 Byte in eine Richtung. Die mit dem Rückverkehr ankommenden Bestätigungen sind sehr klein. Wie groß ist der maximale Durchsatz für die Fenstergrößen 1, 7, 15 und 127? Die Übertragungszeit Erde–Satellit beträgt 270 ms.

40 Ein 100 km langes Kabel läuft mit T1-Geschwindigkeit. Die Übertragungsgeschwindigkeit im Kabel beträgt 2/3 der Lichtgeschwindigkeit. Wie viele Bits passen in das Kabel?

41 Geben Sie mindestens einen Grund an, warum PPP Byte Stuffing anstelle von Bit Stuffing verwendet, um unabsichtliche Flagbytes innerhalb der Nutzdaten daran zu hindern, Konfusion zu verursachen.

42 Wie hoch ist der minimale Overhead beim Senden eines IP-Pakets über PPP? Berücksichtigen Sie nur den von PPP selbst eingebrachten Overhead, nicht den des IP-Headers. Wie hoch ist der maximale Overhead?

43 Ein 100-Byte-IP-Paket wird über eine Teilnehmeranschlussleitung mithilfe des ADSL-Protokollstapels übertragen. Wie viele ATM-Zellen werden übertragen? Beschreiben Sie kurz deren Inhalte.

44 Das Ziel dieser Praxisübung ist, einen Fehlererkennungsmechanismus unter Verwendung des im Text beschriebenen CRC-Standardalgorithmus zu implementieren. Schreiben Sie zwei Programme, *generator* und *verifikation*. Das *generator*-Programm liest von der Standardeingabe eine Zeile ASCII-Text ein, die eine n-Bit-Nachricht als Folge von Nullen und Einsen enthält. Die zweite Zeile ist das k-Bit-Polynom, ebenfalls in ASCII. Die Ausgabe erfolgt auf der Standardausgabe als Zeile mit ASCII-Text mit $n+k$ Nullen und Einsen, die die zu übertragende Nachricht darstellen. Dann wird das Polynom wie eingelesen ausgegeben. Das *verifikation*-Programm liest die Ausgabe des *generator*-Programms und gibt eine Meldung aus, ob diese korrekt ist oder nicht. Schreiben Sie dann ein Programm *aenderung*, das in der ersten Zeile das im Argument des Programms angegebene Bit verändert (das am weitesten links stehende Bit habe die Nummer 1), den Rest der beiden Zeilen aber korrekt kopiert. Durch die Eingabe von

```
generator <file | verifikation
```

sollten Sie feststellen können, dass die Nachricht korrekt ist; durch die Eingabe von

```
generator <file | aenderung arg | verifikation
```

sollten Sie eine Fehlermeldung erhalten.

Die MAC-Teilschicht (Medium Access Control)

4

Copyright