

Rechnernetze – Media Networking (WiSe 2023/2024)

Übung 04

Aufgabe 1) Eine Anwendung produziert einen regelmäßigen Datenstrom (alle t Mikrosekunden werden n Bytes auf einmal produziert). Um höchstmögliche Interaktivität zu ermöglichen, müssen die Informationen spätestens nach 100 Millisekunden beim Empfänger ankommen. Die Übertragungsverzögerung beträgt je nach Stausituation im Netz 70 bis 80 Millisekunden. Wie würde diese Anwendung von einer (ATM-ähnlichen) zellenorientierten Vermittlungstechnik der folgenden Zellengrößen (zuzüglich 5 Bytes Header) Gebrauch machen

a) 42 Bytes,

b) 290 Bytes,

wenn t und n die folgenden Werte hätten:

i) $t = 125$ und $n = 1$ (Audio)

ii) $t = 40000$ und $n = 2500$ (Video)?

Beurteilt die angegebenen Zellengrößen für die Fälle i) und ii): Welcher Prozentsatz, der insgesamt für den jeweiligen Datenstrom übertragenen Bytes, wird für Nutzinformationen verwendet? Welche Zellengröße wäre für die beiden betrachteten Szenarien jeweils optimal?

ATM steht für Asynchronous Transfer Mode und ist eine Vermittlungstechnik für Netzknoten (ATM-Switches). Bei ATM werden Ströme beim Sender in "Häppchen" zerhackt und so ein gleichmäßiger Zellstrom gebildet.

ATM-Zellen haben feste Größe, damit die Switches, die Datenpakete vom Sender-Port zum Empfänger-Port weiterleiten, mit einer festen Taktung arbeiten und interne Puffern zum Zwischenspeichern von Zellen einfacher organisiert werden können.

Alle Zellen müssen eine gewisse Menge an Steuerinformationen bekommen.

Wenn die jeweiligen Zellen sehr klein sind, bleibt nicht viel über für die eigentlichen Daten, man hat also einen sehr großen Header-Overhead.

Wenn sie andererseits sehr groß sind, füllen die Daten, die man verschicken will, die Zellen möglicherweise nicht aus, sodass ein großer Verschnitt entsteht. Dies führt zu einer ineffizienten Nutzung der verfügbaren Ressourcen, da ein Teil der Zellen-Kapazität leer bleibt.

Das ist besonders bei Audio-Strömen ärgerlich, da hier regelmäßig kurze Samples entstehen, die regelmäßig und mit geringer Verzögerung transportiert werden sollen.

Es bietet sich dabei nicht an, eine größere Anzahl der Samples erst einmal beim Sender aufzusammeln, um dann eine größere Zelle zu füllen und auf einmal zu schicken.

Die Informationsströme, die nun aus den Anwendungen kommen, müssen innerhalb von ATM in genau diese Häppchen zerlegt werden, das macht der sogenannte ATM Adaption Layer (AAL).

Wie das Zerhacken einzeln geht, hängt davon ab, wie die Ströme beschaffen sind. Wenn es sehr gleichmäßige Ströme einer festen Rate sind, wird man beim Zerhacken mehr oder weniger große Scheiben abschneiden.

Die entstandenen Häppchen werden dann mit den Header-Informationen versehen und mit der Bildübertragungsrate des darunterliegenden links durch das Netz geschoben.

So können sie dann beim Empfängerport richtig identifiziert werden.

Wir unterscheiden 4 Fälle

Fall 1: a) i)

Die Zellgröße beträgt 42 Bytes (+ 5 Bytes Header) = 47 Bytes.

Es werden alle 125 Mikrosekunden 1 Byte auf einmal produziert.

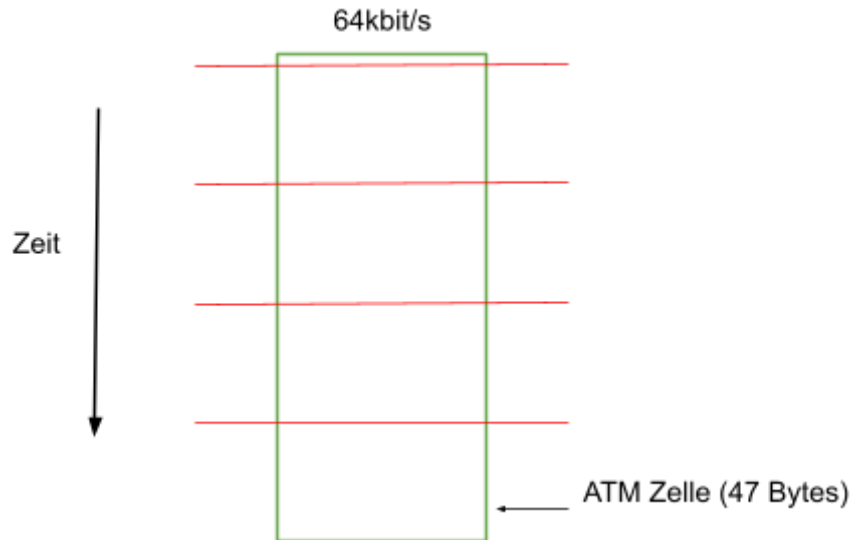
Dabei erinnert dieser Fall an einen Audiostrom.

Es wird 8000 Mal pro Sekunde ein bestimmter Audio-Sample Wert abgegriffen und das in 1 Byte kodiert, sodass sich eine 64 kbit/s Kommunikation ergibt, mit regelmäßigen Muster ($8000 * 8 \text{ Bit} = 64000 \text{ Bit} = 64 \text{ kbit/s}$ übertragen).

Der produziert also 8000 Mal pro Sekunde ein Byte (jede $1/8000$ Sekunde ein Byte):

Wir haben hier 42 Bytes "Platz" für Nutzinformationen und 5 Bytes sind für den Header vorgesehen.

Die Informationsströme müssen innerhalb der ATM in genau diese Teile zerlegt werden.



Da hier die Zellgröße 47 Bytes groß ist, werden die Ströme auch in der Größe geteilt. Der jeweilige Prozentsatz für die Nutzinformationen kann wie folgt bestimmt werden:

Pro Zelle wird 1 Byte von den 42 zur Verfügung stehenden Bytes (für Nutzinformationen) verwendet.

Anzahl der Zellen: $64000 \text{ Bit} / (47 * 8 \text{ Bit}) = 170,21 = 170 \text{ Zellen (pro Sekunde)}$
Mit jeweils 1 Byte (8 Bit) pro Zelle ergibt sich: $170 * 8 \text{ Bit} = 1360 \text{ Bit}$

1360 Bit werden also pro Sekunde als Nutzinformationen verwendet.
Das sind also $1360 \text{ Bit} / 64000 \text{ Bit} = 0,0213 = 2,13\%$

Hier zeigt sich, dass aufgrund der zu großen Zelle der Verschnitt sehr hoch ist und man die Zellen sehr ineffizient füllt.

Ein optimaler Wert wäre in dem Fall eine Zellgröße von 6 Byte, da wir effektiv nur 1 Byte füllen und 5 Byte für den Header benötigen.

Analog wie oben berechnet, werden hier die Datenströme in $64000 / (6 * 8) = 1333,33 = 1333 \text{ Zellen}$ verteilt, mit jeweils 8 Bit, also $1333 * 8 = 10664 \text{ Bits}$

Das wären dann $10664 \text{ Bit} / 64000 \text{ Bit} = 0,16665 = 16,66\%$.

Alternativ könnte man die Zellen auch auffüllen. Da alle 125 Mikrosekunden 1 Byte produziert wird und man 42 Bytes Platz für Nutzinformationen hat, würde eine zusätzliche Verzögerung von $42 * 0,125 \text{ ms} = 5,25 \text{ ms}$ entstehen. Da Verzögerungen unter 100ms noch in Ordnung sind, wäre das eine gute alternative Lösung.

Beim Auffüllen:
(In Rot der Datenstrom)



Abbildung 1

Und wie groß wäre dann der Nutzdatenanteil, nach dem in der Aufgabe gefragt wurde?

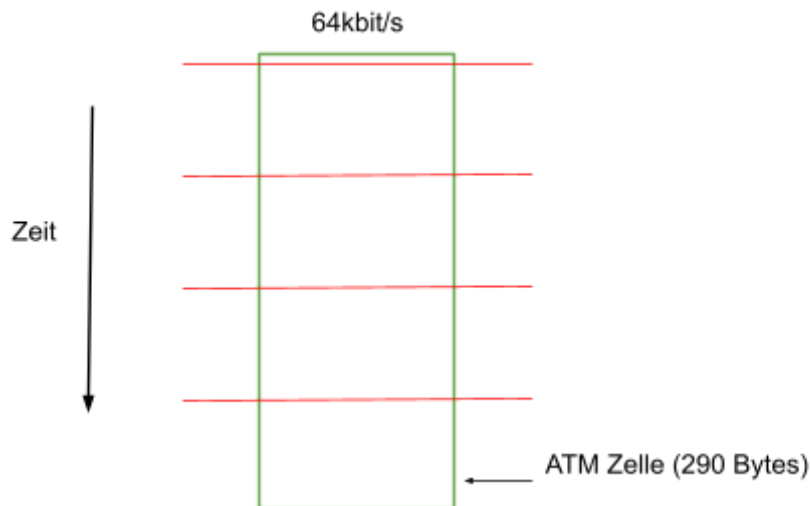
Analog ergibt sich Fall 2: b) i)

Auch ein Audio ähnlicher Strom, wobei diesmal die Zellgröße 290 Bytes + 5 Bytes Header beträgt.

Analog zu Fall 1 ergibt sich folgende Rechnung.

Anzahl der Zellen: $64000 \text{ Bit} / (295 * 8 \text{ Bit}) = 27,12 = 27 \text{ Zellen (pro Sekunde)}$

Mit jeweils 1 Byte (8 Bit) pro Zelle ergibt sich: $27 * 8 \text{ Bit} = 216 \text{ Bits}$



1360 Bit werden also pro Sekunde als Nutzinformationen verwendet.

Das sind also $216 \text{ Bit} / 64000 \text{ Bit} = 0,003375 = 0,3375\%$

Hier wäre auch eine Zellgröße von 6 analog wie bei Fall 1 optimal.

Jedoch wäre eine Auffüllung der Zellen problematisch.

Dabei müsste man 290 Byte auffüllen, statt 42, wodurch ich eine Verzögerungszeit von 36,25ms ergibt (siehe Abbildung 2).



Abbildung 2

Fall 3: a) ii)

Hier wäre aufgrund der Verzögerungszeiten nur ein Auffüllen bis 160 Bytes möglich

Die optimale Paketgröße wäre also 160 Bytes.

Der zweite Strom ii) erinnert an einen regelmäßigen Videostrom.

Wir haben hier wieder eine Zellgröße von 42 Bytes. Alle 40 ms werden 2500 Bytes produziert.

Somit ergeben sich $2500 \text{ Byte} * 25 = 62500 \text{ Bytes pro Sekunde}$, also 500000 Bit.

Die Rechnung ist analog wie in den bisherigen Fällen:

Anzahl der Zellen: $500000 \text{ Bit} / (42 * 8 \text{ Bit}) = 1329,79 = 1329 \text{ Zellen (pro Sekunde)}$

Hier wird jedoch die Zelle komplett gefüllt, also haben wir 42 Byte Nutzinformationen pro Zelle.

Damit ergeben sich $42 \text{ Byte} * 1329 * 8 = 446544 \text{ Bits}$

446544 Bit werden also pro Sekunde als Nutzinformationen verwendet.

Das sind also $446544 \text{ Bit} / 500000 \text{ Bit} = 0,8931 = 89,3\%$

Hier müsste jeder Burst auf 60 Pakete aufgeteilt werden.

Fall 4: b) ii)

Analog wie Fall 3 mit einer Zellgröße, die 295 Byte groß ist.

Anzahl der Zellen: $500000 \text{ Bit} / (295 * 8 \text{ Bit}) = 211,86 = 211 \text{ Zellen (pro Sekunde)}$

Auch hier wird die Zelle komplett gefüllt.

Also 290 Byte Nutzinformationen, wodurch sich folgende Anzahl an Bits ergibt:

$290 \text{ Byte} * 8 * 211 = 489520 \text{ Bits}$

489520 Bit werden also pro Sekunde als Nutzinformationen verwendet.

Das sind also $489520 \text{ Bit} / 500000 \text{ Bit} = 0,979 = 97,9\%$

Die optimale Zellgröße in Fall 3 und Fall 4 wäre bei 296 Bytes (zuzüglich 5 Byte Header), also bei 301 Bytes erreicht.

Dabei würde sich eine gesamtzahl von $296 \text{ Byte} * 8 * 211 = 499648 \text{ Bits}$ ergeben.

Womit $499648 \text{ Bit} / 500000 \text{ Bit} = 0,9993 = 99,93\%$ Nutzinformationen in den übertragenen Bytes enthalten.

Die restlichen 352 Bits, also 44 Bytes, wären Verschnitt.

Hier müsste jeder Burst auf 9 Pakete aufgeteilt werden.

Die optimale Paketgröße wäre 2500 Bytes.

1 Punkt

?

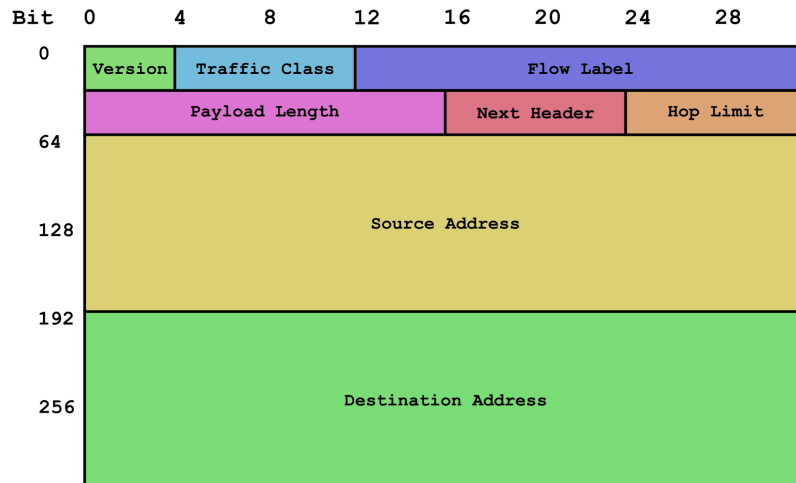
Aufgabe 2) Eine fiktive Netzkopplung nutzt einen fiktiven Subnetztyp mit einer Paketgröße von 50 Bytes und eigenem Fragmentierungsmechanismus. In dieser Netzkopplung soll auf der Basis von MPLS, IPv6 und UDP eine fiktive Anwendung aufgesetzt werden. Um die erforderliche Anzahl der Pakete zu minimieren, sollen die IPv6- und UDP-Header dabei soweit wie möglich reduziert werden (Header Compression). a) Welche Felder können nach Einrichten des Label Switched Paths (LSP) eingespart werden, welche dagegen nicht, wenn die LSPs Data-driven sind? Kurze Begründung. b) Was wäre anders, wenn der LSP Topology-driven wäre?

Aufgabe 2

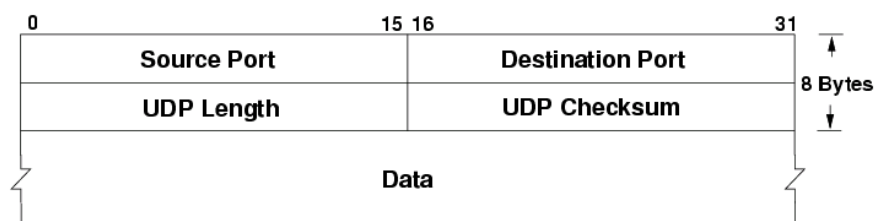
Beim MPLS-Protokoll wird jedem Datenpaket ein sogenanntes "Label" zugewiesen. Anhand dieses Etiketts wird der Weg bestimmt, den das Paket durch das Netz nehmen soll. Im Vergleich zu herkömmlichen Routing-Protokollen, die auf IP-Adressen basieren, ermöglicht MPLS eine schnellere Verarbeitung der Pakete, da Entscheidungen auf der Grundlage der Labels getroffen werden. Dieser Ansatz wird als "Label Switched Path" (LSP) bezeichnet.

Aufgabe 2a)

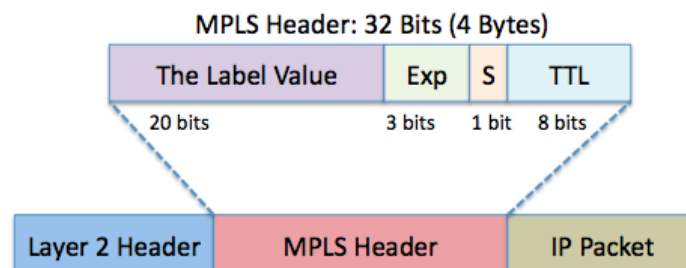
Um festzustellen, welche Felder nach dem Einrichten des LSP eingespart werden können, muss man sich zunächst die Header-Struktur der beteiligten Protokolle ansehen:



IPv6-Header



UDP-Header



MPLS-Header

Angenommen, jeder Router unterstützt das MPLS-Protokoll, d.h. er versucht standardmäßig, das Paket auf der Basis dieses Protokolls weiterzuleiten. Bei einem datengesteuerten LSP sind Angaben wie Quelladresse, Zieladresse, Quellport, Zielport und Protokoll erforderlich, um ein Label zu erstellen. Man könnte nun sagen, dass die Zieladresse und die Quelladresse in einem IPv6-Header weggelassen werden können, da sie bereits im Label enthalten sind. Allerdings handelt es sich hier um das IPv6-Protokoll, bei dem die Adressen 128 Bit lang sind. In einen MPLS-Header passt jedoch nur ein Label mit einer maximalen Größe von 20 Bit. Es wäre zwar möglich, das Tupel mit den Angaben zu hashen, um auf eine feste Größe zu kommen, aber dann wären die Angaben nicht mehr ohne weiteres lesbar. Daher müssen die Adress-Felder im IPv6-Header verbleiben.



Bei einem Data-Driven LSP identifiziert das Label den End-zu-End-Datenstrom, der klassisch durch das Quintupel von IP-Adressen, Port-Nummern und Protokollbezeichner bezeichnet wird. Das heißt aber nicht, dass die konkreten Werte des Quintupels im Label enthalten sind.

Anders verhält es sich bei den IPv6-Feldern wie *Traffic Class*, *Flow Label* und *Hop Limit*. Das Feld *Traffic Class* dient dazu, Quality of Service (QoS) zu ermöglichen, indem verschiedenen Arten von Datenverkehr unterschiedliche Prioritäten zugewiesen werden können. So könnten z.B. Echtzeitdaten wie Sprach- oder Videodaten vorrangig behandelt werden. Ein ähnliches Feld ist aber auch in einem MPLS-Header enthalten, sodass es eingespart werden kann. Das nächste Feld, *Flow Label*, wird verwendet, um den Fluss der Pakete zwischen einer Quelle und einem Ziel zu identifizieren. Damit können Pakete, die zu einem bestimmten Datenfluss gehören, gekennzeichnet werden. Im Grunde erfüllt es einen ähnlichen Zweck wie das Feld Label in einem MPLS-Header und kann daher weggelassen werden. Das letzte Feld *Hop Limit* ist vorhanden, um die Lebensdauer eines Datenpakets zu begrenzen und Endlosschleifen im Netzwerk zu verhindern. Das Feld ist vergleichbar mit dem TTL-Feld eines MPLS-Headers und kann daher auch entfallen.

Aus einem UDP-Header können zwei Felder, nämlich *Source Port* und *Destination Port*, gestrichen werden, da sie bereits im entsprechenden Label vorhanden sind. Der Egress-Router bzw. End-Router muss dann die Portangaben wieder in den UDP-Header des Pakets einfügen, damit der Endrechner Zugriff auf diese Informationen hat.

Aufgabe 2b)

Bei einem topologiegesteuerten LSP werden die Entscheidungen über die für Datenpakete verwendete Route auf der Basis einer vordefinierten Netztopologie getroffen. Anstelle von datengesteuerten LSPs, bei denen die Pfadauswahl auf dem Inhalt der Datenpakete beruht, wird hier der Pfad statisch oder dynamisch anhand der eingerichteten Netztopologie bestimmt. Bei den hier eingesetzten Labels handelt es sich um Surrogate, d.h. Labels, die nicht aus Daten abgeleitet sind. Für uns bedeutet dies zunächst, dass alle Felder, die bei der datengesteuerten LSP weggelassen wurden, auch hier wegfallen können. Allerdings müssen die Felder *Source Port* und *Destination Port* wieder in den UDP-Header aufgenommen werden, da sie an anderer Stelle, d.h. im Label, nicht mehr zu finden sind.

nee

1,5 Punkte