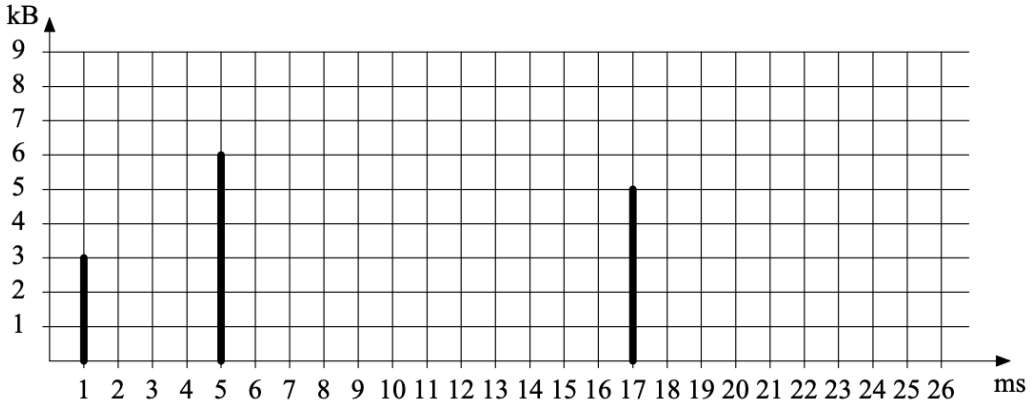
Σ 4,5/5

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

Rechnernetze - Media Networking (WiSe 2023/2024)

Übung 05

1) In einer fiktiven Anwendung möchte ein Sender zu den in der folgenden Tabelle angegebenen Zeitpunkten die jeweils angegebene Datenmenge absenden.



Wie würde dieser Strom bei Anwendung der folgenden Traffic-Shaping-Modelle geformt werden?

a) *Leaky Bucket:* Ab t=0 ms alle 2 ms (max.) 2 kB.

b) *Token Bucket:* Ab t=0 ms alle 2 ms 2 Token für je 1 kB Daten (die ersten 2 Token entstehen also zum Zeitpunkt t=0 ms). Größe des Token Buckets: 4 Token.

Bemerkungen:

- Ihr könnt vereinfacht davon ausgehen, dass diese Umformung für die Anwendung keine Probleme darstellt und beliebige, im Rahmen des Traffic-Shaping-Verfahrens zulässige Paketgrößen geformt werden können.
- Die zu sendenden Informationen sollen immer so früh wie möglich abgesendet werden.
- Notiert auch, welche Informationen des Ausgabestroms zu welchem ursprünglichen Paket gehören.

(2 Punkte)

Aufgabe 1

Aufgabe 1a)

Das Prinzip des "Leaky Bucket" (Undichter Eimer) ist ein bekanntes Konzept, bei dem die zu übertragenden Informationen in den Eimer von oben geschüttet werden, entweder auf einmal oder in kleinen Häppchen. Der Eimer hat unten ein Loch, durch das in regelmäßigen Abständen bestimmte, kleinere Mengen dieses Datenstroms herausfließen, etwa alle t Zeiteinheiten.
In unserem Fall ab darf man ab dem Zeitpunkt t=0ms alle 2ms 2kB abschicken, also erstmalig zum Zeitpunkt t=0ms ist das Verschicken erlaubt.
Da zu dem Zeitpunkt noch keine Daten anstehen, wird auch nichts verschickt.
Zum Zeitpunkt t=1ms stehen 3 kB Daten an. Jedoch können erst nach 2ms Daten verschickt werden.

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

Daher wird erstmalig zum Zeitpunkt t=2ms 2kB Daten verschickt. 1kB bleibt noch im Bucket, bis weitere 2ms vergehen, also zum Zeitpunkt t=4ms können 2kB verschickt werden, jedoch befindet sich nur 1kB im Bucket, aber wird nun nur 1kB verschickt.
Zum Zeitpunkt t=5ms stehen 6kB Daten an. Da alle 2ms Daten verschickt werden dürfen, werden diese 6kB über drei Zeitpunkten zu jeweils 2kB verschickt, nämlich zum Zeitpunkt t=6ms, 8ms und 10ms. Erst zum Zeitpunkt t=17ms stehen wieder Daten an. Hier stehen nun 5kB Daten an, von denen zum Zeitpunkt t=18ms 2kB verschickt werden, dann zum Zeitpunkt t=20ms und schließlich zum Zeitpunkt t=22ms der letzte kB des Datenstroms.

Die Abbildung 1 visualisiert das beschriebene Ergebnis.
Die Farben (rot, grün und blau) visualisieren, welche Daten zu welchen Paketen gehören. Rot für das erste Paket (1ms), grün für das zweite (5ms) und blau für das dritte Paket (17ms).

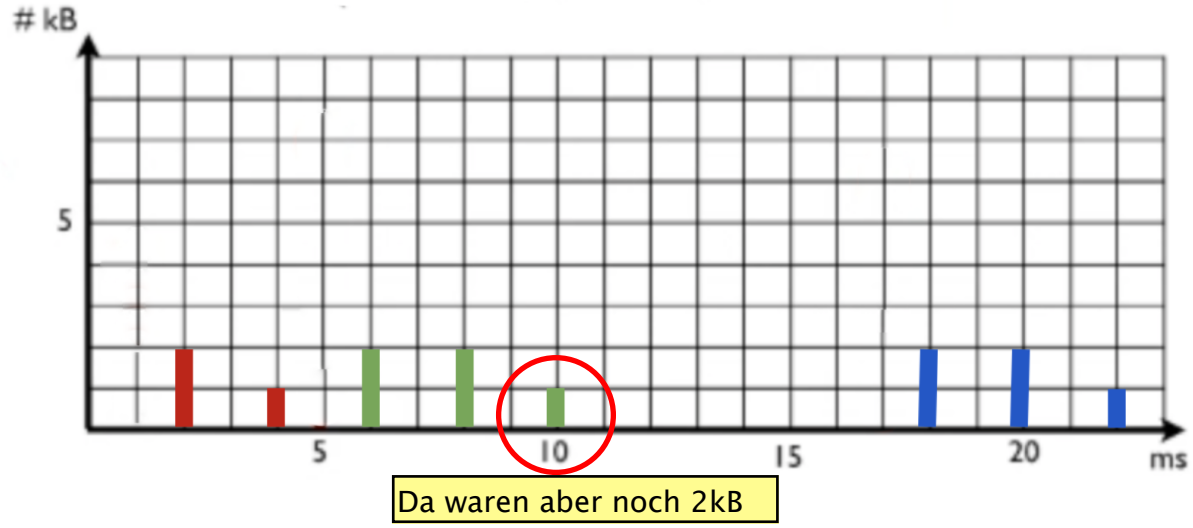


Abbildung 1: Leaky Bucket-Verformung

Das Prinzip von Leaky Bucket eignet sich besonders gut für sehr regelmäßige Datenströme, bei denen die Übertragung in vordefinierten Intervallen erfolgt. Dabei wird der Datenstrom gleichmäßig über die Zeit verteilt. Es ist jedoch wichtig zu beachten, dass das Leaky Bucket-Modell keine Flexibilität für Übertragungsspitzen (Bursts) bietet. Das Modell ist daher weniger geeignet für Situationen, in denen es zu vorübergehenden, deutlichen Anstiegen im Datenverkehr kommt.

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

Aufgabe 1b)
Bei der Anwendung des Token Bucket als Traffic Shaping Modell würde der Strom wie folgt geformt werden.
Dabei ist zu beachten, dass beim Token Bucket in regelmäßigen Abständen Senderechte (Die Token) in den Token Bucket. Dabei entsprechen die Abstände 2ms und die Anzahl der Token genau zwei.
Jeder Token steht für ein Senderecht von 1kB.

Auch hier geht es wie bei a) ab t=0ms los, das heißt, die ersten beiden Token fallen zu dem Zeitpunkt in den Bucket rein.
Die maximale Bucket Größe ist auf 4 Token beschränkt.
Zum Zeitpunkt t=0ms liegen also 2 von maximal 4 Token im Bucket. Zu dem Zeitpunkt stehen keine Daten an.
Für t=1ms gilt ebenfalls 2/4 Token, jedoch stehen nun 3kB Daten an. Es werden also 2 Token aufgebraucht zum Senden von 2kB, wodurch nur noch 1kB anliegen und 0/4 Token vorhanden sind.
Zum Zeitpunkt t=2ms fallen wieder 2 Token, wodurch der letzte anstehende Kilobyte versendet werden kann.
Ein Token bleibt dabei übrig.

t=3ms: 1 von 4 Token vorhanden, keine Daten liegen an.
t=4ms: 3 von 4 Token vorhanden, keine Daten liegen an.
t=5ms: 3 von 4 Token vorhanden, 6 kB Daten liegen an. Dadurch werden 3 Token verbraucht und 3 kB liegen weiterhin an.
t=6ms: 2 von 4 Token vorhanden, 3kB Daten liegen an. Dadurch werden 2 Token verbraucht und 1 kB liegt weiterhin an.
t=7ms: 0 von 4 Token vorhanden, 1kB an Daten liegt an.
t=8ms: 2 von 4 Token vorhanden, 1kB an Daten liegt an. Dadurch wird ein Token verbraucht und ein Token bleibt im Bucket.
t=9ms: 1 von 4 Token vorhanden, keine Daten liegen an.
Zum Zeitpunkt t=12ms ist der Bucket voll, da erst zum Zeitpunkt t=17 5kB Daten vorliegen, wird davor nichts passieren, außer das sich der Token Bucket einmal füllt und die zusätzlichen Token verworfen werden.
Zum Zeitpunkt t=17ms: sind 4 von 4 Token vorhanden, 5kB Daten liegen an. Dadurch werden 4 Token verbraucht und 1kB an Daten liegt weiterhin an.
Zum Zeitpunkt t=18ms sind 2 von 4 Token vorhanden und 1kB an Daten liegt an. Ein Token wird verbraucht und ein Token bleibt übrig.

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

Die Abbildung 2 visualisiert das beschriebene Ergebnis.

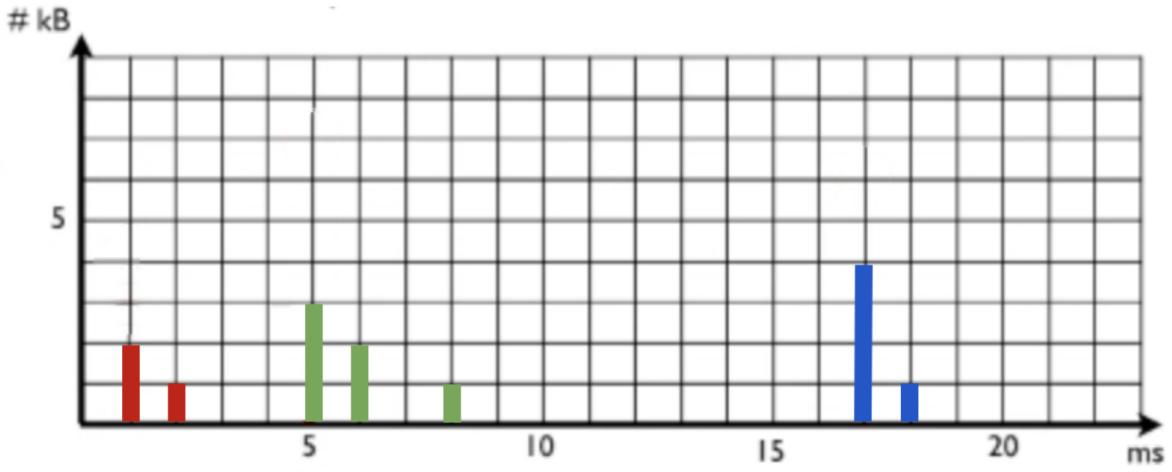
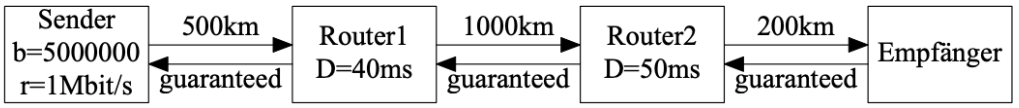


Abbildung 2: Token Bucket-Verformung

Die Farben (rot, grün und blau) visualisieren, welche Daten zu welchen Paketen gehören. Rot für das erste Paket (1ms), grün für das zweite (5ms) und blau für das dritte Paket (17ms).

2 Punkte

2) In einem fiktiven Glasfasernetz wird IntServ eingesetzt.



Nach Aufsetzen einer Reservierung mit den oben angegebenen Parametern und maximal gefülltem Token Bucket startet der Sender mit einem Burst von 5MB.

Wann sind diese Informationen vermutlich vollständig beim Empfänger angekommen (ab Startpunkt des Sendens)?

(1 Punkt)

Abbildung 3: Aufgabe 2

Bei IntServ handelt es sich um die Internettechnologie für ein Reservierungsverfahren, welches nach dem Grundprinzip arbeitet, dass nicht die Sender, sondern die Empfänger für den Datenstrom der Sender die Reservierung vornehmen.

Damit Empfänger diesen Reservierungsantrag überhaupt stellen können, müssen die Sender im Vorfeld ihre FlowSpec bekanntgeben, um zu sagen, wie der Strom jetzt beschaffen sein soll.
Innerhalb von IntServ gibt es das Guaranteed Service-Modell.

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

Hierbei versucht man, Reservierungen zu etablieren, die mit garantierten Verzögerungszeiten arbeiten.
In einem Guaranteed-Szenario würde man versuchen, mithilfe zusätzlicher Parameter genauer zu spezifizieren, wie groß die Verzögerung sein wird.
Dabei beginnt der Sender mit seinen beiden Parametern b und r.
Die Router unterwegs würden noch aufaddieren, was sie an zusätzlichen Verzögerungen produzieren bei so einem Strom
Die Einheit der Verzögerung hängt vom Kontext ab.
Bei Parameter D handelt es sich um eine Zeitverzögerung, die beschreibt, wie lange ein Router intern braucht, um ein empfangenes Paket zu verarbeiten.

Beim Starten des Sendens mit einem Burst von b Bytes, die ebenso wie auch alle weiteren verzögerungen, die in Bytes berechnet werden, mit einer Rate von R abgearbeitet werden, entsteht nun eine Gesamtverzögerung, die durch die Formel:

$$\frac{b+C}{R} + D + S \ (+\text{Wegstrecke})$$

D: Summe der Zeitverzögerung (Wie lange braucht ein Router intern
um ein Empfängerpaket zu verarbeiten)
S: Zusätzlich erlaubte Verzögerung
b: maximale Burst (Es können nicht mehr als b Bytes gleichzeitig zum Senden anstehen/im Bucket sein)
r: welche Datenrate für diesen Strom reserviert werden sollte. Damit wird indirekt bestimmt, welchen Anteil der gesamten Übertragungsrate der einzelnen Links diesem Strom zur Verfügung gestellt werden soll.
R: Welche Bandbreite soll tatsächlich reserviert werden?
Bei R>r: Mehr reservieren, um Übertragungsspitzen abzufangen

Die Wegstrecke ergibt sich bei einer Gesamtstrecke von 1000 km + 500 km + 200 km zu 1700 km.
In Glasfaser beträgt die Übertragungsgeschwindigkeit ungefähr 200.000 km/s
Daraus ergibt sich eine Weg-Verzögerungszeit von
1700km / 200.000 km/s = 0,0085s = 8,5ms

Für D gilt: 40ms (Router1) + 50ms (Router2) = 90ms
b = 5 Mbit = 5.000.000 Bit
r = 1 Mbit /s = 1.000.000 Bit/s

Das waren 5 MByte

→ (5000000 Bit+0)/1000000 Bit/s + 90ms + 0 + 8,5ms
= 5s + 98,5ms = 5,0985s

Also 5,0985s, nachdem der Sender startet, sind die Informationen beim Empfänger angekommen.

0,5 Punkte

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

3) In einem fiktiven Netz wird ein DiffServ-ähnliches Verfahren eingesetzt. Darin können Datenpakete fester Größe übertragen werden, die mit einem von sechs verschiedenen *DSCPs (Differentiated Services Code Points)* ausgezeichnet sind (a,b,c,d,e,f). Die DSCPs werden in einem Router auf die folgenden *PHBs (Per Hop Behaviors)* abgebildet:

a Alle Pakete müssen ankommen; eine Verzögerung von 1 Zeitintervall ist o. k. (aber nicht mehr).

b Die Pakete werden mit maximal 2 Zeitintervallen Verzögerung weitergeleitet. Zuverlässigkeit ist gewünscht, aber nicht zwingend erforderlich.

c Die Pakete werden immer direkt (d. h. ohne zusätzliche Verzögerung) weitergeleitet. Sollte das gerade nicht gehen, werden sie verworfen.

d Alle Pakete müssen ankommen, aber die Verzögerung spielt keine Rolle.

e Wie b, aber mit geringerer Priorität.

f Die Pakete werden nach dem *Best-Effort*-Prinzip behandelt.

Der Router hat Pufferplatz für 3 Pakete.

Auf den drei Eingabe-Ports liegen die nachfolgend angegebenen Paketströme an, die alle auf denselben Ausgabe-Port weitergeleitet werden sollen (die verwendeten Paketbezeichner in der Tabelle enthalten zur eindeutigen Kennzeichnung jeweils Eingabeport, Zeitpunkt und DSCP).

Eingabeport	**Zeitpunkt**			
	1	**2**	**3**	**4**
A	A1b	A2c	A3a	A4e
B	B1a	B2c	B3e	B4d
C	C1d	C2f	C3a	C4d

Wie sieht der Ausgabestrom aus? Sind die Anforderungen erfüllbar? Begründung.

Hinweis: Die Entscheidungen für das Weiterleiten vs. Puffern vs. Verwerfen werden direkt getroffen, also sobald die betreffenden Pakete an den Eingabe-Ports anliegen.

Abbildung 4: Aufgabe 3

Der Ausgabestrom lässt sich übersichtlich in einer Tabelle darstellen:

Zeitpunkt t	Paketstrom	Puffer (max. 3)	Verworfen
1	B1a	A1b, C1d	
2	A2c	A1b, C1d, C2f	B2c
3	A3a	C1d, C3a, B3e	A1b, C2f
4	C3a	C1d, B4d, C4d	B3e, A4e

Erklärung (nach Zeit t):

1. Pakete des Stroms B1a werden sofort weitergeleitet, da der Ausgangskanal derzeit frei ist und der Strom die höchste Priorität unter den anderen hat. Die Pakete der anderen Ströme werden gepuffert.

Cihasev Adam
Erdogan İbrahim
Jaber Muhamed

2. Pakete des Stroms A2c werden weitergeleitet. Die Pakete von B2c haben die gleiche Priorität wie die von A2c, da es aber nur einen Ausgangskanal gibt und Verzögerungen nicht zulässig sind, werden sie verworfen.
3. Alle Pakete mit DSCP a müssen mit einer maximalen Verzögerung von 1 Zeitintervall ankommen. Somit müssen Pakete des Stroms A3a sofort weitergeleitet werden, da hier zwei Paketströme mit DSCP a anstehen und die Pufferung beider Paketströme die Verzögerungszeit für einen von ihnen überschreiten und die Garantie für QoS brechen würde. Der zweite Paketstrom C3a wird gepuffert, indem Paketströme mit nicht zwingender Zuverlässigkeit (also A1b, C2f) verworfen werden.
4. Paketstrom C3a wird weitergeleitet. Die Paketströme mit DSCP d unterliegen einer Zuverlässigkeitspflicht und werden daher gepuffert. Die niederprioren Paketströme B3e und A4e werden verworfen, um Platz zu schaffen. In den nächsten Zeitschritten werden die verbleibenden gepufferten Paketströme nacheinander weitergeleitet.

Die Auswertung zeigt, dass alle Anforderungen erfüllt werden können.
Damit sieht der Ausgabestrom folgendermaßen aus: B1a, A2c, A3a, C3a, C1d, B4d, C4d

Das sind keine Ströme, sondern einzelne Pakete.

2 Punkte