

Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme Übungsblatt 7 (3. Juni – 7. Juni 2013)

Hinweis: Die mit * gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

Aufgabe 1 Subnetting

Der Moepinet AG werden die Adressbereiche 131.159.32.0/22 und 131.159.36.0/24 zugewiesen. Für die Aufteilung dieses Adressbereichs ist die Moepinet AG selbst verantwortlich. Nach einer sorgfältigen Bedarfsanalyse ergeben sich die folgenden Anforderungen an die Subnetze und die Mindestanzahl **nutzbarer** IP-Adressen:

Subnetz	NET 1	NET 2	NET 3	NET 4	NET 5
IPs	300	300	15	40	4

Bei der Erhebung dieser Zahlen wurde die an das jeweilige Router-Interface zu vergebende IP-Adresse bereits berücksichtigt.

a)* Geben Sie jeweils die erste und letzte IP-Adresse der beiden vergebenen Adressbereiche an.

- 131.159.32.0/22:
Erste IP: 131.159.32.0 (Netzadresse)
Letzte IP: 131.159.35.255 (Broadcast-Adresse)
- 131.159.36.0/24:
Erste IP: 131.159.36.0 (Netzadresse)
Letzte IP: 131.159.36.255 (Broadcast-Adresse)

b) Wie viele IP-Adressen stehen der Moepinet AG insgesamt zur Verfügung? Können alle davon zur Adressierung von Hosts verwendet werden?

- 131.159.32.0/22: $2^{32-22} = 2^{10} = 1024$ Adressen
- 131.159.36.0/24: $2^{32-24} = 2^8 = 256$ Adressen

Insgesamt stehen also $1024 + 256 = 1280$ Adressen zur Verfügung. Allerdings sind die erste (Netzadresse) und letzte Adresse (Broadcast-Adresse) eines jeden Netzes nicht zur Adressierung von Hosts nutzbar. Es stehen also zunächst maximal $1022 + 254 = 1276$ Adressen zur Hostadressierung zur Verfügung.

c)* Ist es möglich, den von den beiden Adressblöcken gebildeten Adressbereich in einem einzigen Subnetz zusammenzufassen?

Nein. Es gibt keine passende Subnetzmaske, welche den insgesamt 1280 Adressen großen Block definieren könnte. Die Größe eines Netzes kann stets nur eine Zweierpotenz sein.

d)* Teilen Sie nun die beiden Adressbereiche gemäß der Bedarfsanalyse auf, so dass Subnetze der passenden Größe entstehen. Gehen Sie mit den Adressen so sparsam wie möglich um. Es soll am Ende ein möglichst großer zusammenhängender Adressbereich für zukünftige Nutzung frei bleiben. Für jedes Subnetz ist anzugeben:

- die Größe des Subnetzes
- die Anzahl nutzbarer Adressen
- das Subnetz in Präfixschreibweise
- die Subnetzmaske in Dotted-Decimal-Notation
- die Netz- und Broadcastadresse

Um die Vorgaben zu erfüllen, müssen wir die Subnetze gemäß ihrer Größe in absteigender Reihenfolge bearbeiten. Andernfalls könnten wir die folgende Situation erhalten:

- An Netz 3 wird der Adressbereich 131.159.36.0/27 vergeben.
- Vergibt man nun aber an Netz 4 den Bereich 131.159.36.32/26, macht man einen Fehler. Um dies zu verstehen, muss man sich die Binärschreibweise der Netzadresse und Subnetz-Maske ansehen:
 131.159.36.0010 0000 (IP)
 255.255.255.1100 0000 (Subnetz-Maske)
 Eine UND-Verknüpfung beider Zeilen ergibt, dass die IP-Adresse 131.159.36.32 in das Subnetz 131.159.36.0/26 fällt!
- Wir müssten also den Bereich 131.159.36.64/26 an Netz 4 vergeben. Dann allerdings entstünde eine Lücke zwischen Netz 3 und Netz 4.
- Vergibt man die Adressen gemäß der Größe der Subnetze in absteigender Reihenfolge, umgeht man das Problem. Dieses Vorgehen könnte natürlich wieder anderen Kriterien widersprechen – beispielsweise der Vergabe zusammenhängender Adressblöcke an einzelne Niederlassungen.

Subnetz	NET 1	NET 2	NET 3	NET 4	NET 5
Bedarf	300	300	15	40	4
Größe	512	512	32	64	8
Nutzbar	510	510	30	62	6
Präfixnotation	131.159.32.0/23	131.159.34.0/23	131.159.36.64/27	131.159.36.0/26	131.159.36.96/29
Subnetzmaske	255.255.254.0	255.255.254.0	255.255.255.224	255.255.255.192	255.255.255.248
Netzadresse	131.159.32.0	131.159.34.0	131.159.36.64	131.159.36.0	131.159.36.96
Broadcast	131.159.33.255	131.159.35.255	131.159.36.95	131.159.36.63	131.159.36.103

Aufgabe 2 Statisches Routing

Gegeben sei die Netztopologie in Abbildung 1. Die Router R1, R2 und R3 sollen jeweils die höchste nutzbare IP-Adresse in den drei Subnetzen NET1, NET2 und NET3 erhalten. Zur Verbindung zwischen den Routern stehen Transportnetze mit jeweils nur zwei nutzbaren Adressen zur Verfügung. Der Router mit dem lexikographisch kleineren Namen (z. B. R1 < R2) soll hier die kleinere IP-Adresse erhalten.

a)* Weisen Sie jedem Interface der Router R1, R2, R3 und G jeweils eine IP-Adresse zu (Router G nur Interface IF0). Tragen Sie die Adressen direkt in Abbildung 1 ein.

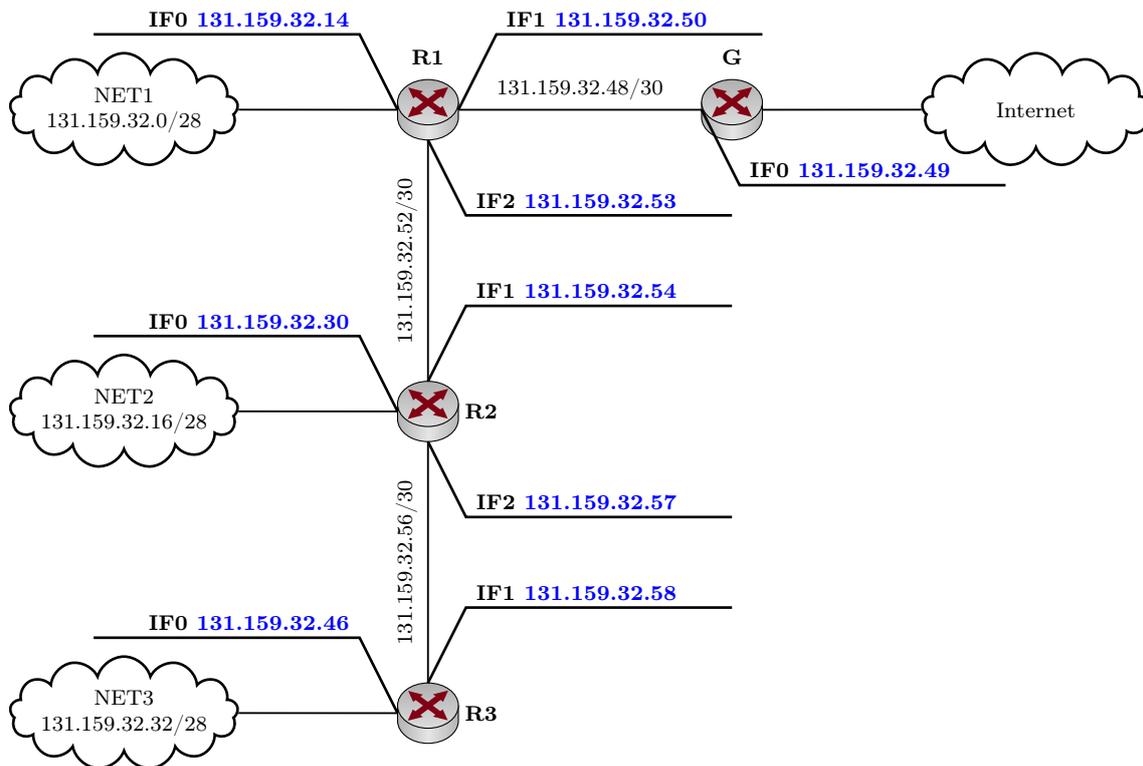


Abbildung 1: Netztopologie

Siehe [Abbildung 1](#).

Die Routingtabelle von R3 sei wie folgt gegeben:

Routingtabelle für R3:

Destination	Gateway	Genmask	Iface
131.159.32.32	0.0.0.0	255.255.255.240	IF0
131.159.32.56	0.0.0.0	255.255.255.252	IF1
0.0.0.0	131.159.32.57	0.0.0.0	IF1

Der Eintrag 0.0.0.0 in der Spalte „Gateway“ bedeutet, dass kein Gateway benötigt wird (Netz ist direkt angeschlossen). Die letzte Zeile ist der Eintrag für den sog. **Default-Gateway**. Dorthin werden Pakete an all diejenigen Netze weitergeleitet, für die keine bessere Route bekannt ist.

b) Geben Sie die Routingtabellen der Router R1 und R2 an. Fassen Sie dabei einzelne Routen soweit möglich zusammen und sortieren Sie die Einträge absteigend in der Länge des Präfixes.

Routingtabelle für R1:

Destination	Gateway	Genmask	Iface
131.159.32.48	0.0.0.0	255.255.255.252	IF1
131.159.32.52	0.0.0.0	255.255.255.252	IF2
131.159.32.0	0.0.0.0	255.255.255.240	IF0
131.159.32.16	131.159.32.54	255.255.255.240	IF2
131.159.32.32	131.159.32.54	255.255.255.240	IF2
0.0.0.0	131.159.32.49	0.0.0.0	IF1

Routingtabelle für R2:

Destination	Gateway	Genmask	Iface
-------------	---------	---------	-------

```

131.159.32.52  0.0.0.0      255.255.255.252 IF1
131.159.32.56  0.0.0.0      255.255.255.252 IF2
131.159.32.16  0.0.0.0      255.255.255.240 IF0
131.159.32.0   131.159.32.53 255.255.255.240 IF1
131.159.32.32  131.159.32.58 255.255.255.240 IF2
0.0.0.0        131.159.32.53 0.0.0.0         IF1

```

Hinweis: Router R1 könnte zusätzlich noch eine Route zum Transportnetz 131.159.32.56/30 besitzen. Dies ist notwendig, sofern R1 direkt mit R3 über die IP-Adresse 131.159.32.58 kommunizieren können soll. Für die Erreichbarkeit der Subnetze spielt dies allerdings keine Rolle.

Die Route ins Netz 131.159.32.0 wird bereits von der Default-Route abgedeckt und kann weggelassen werden, beide Routen denselben Gateway nutzen würden.

c) Geben Sie die Einträge der Routingtabelle von Router G an, welche über Interface IF0 erreichbar sind (wieder zusammengefasst und nach Präfix sortiert).

Partielle Routingtabelle für G:

```

Destination  Gateway      Genmask      Iface
131.159.32.48  0.0.0.0      255.255.255.252 IF0
131.159.32.32  131.159.32.50 255.255.255.240 IF0
131.159.32.0   131.159.32.50 255.255.255.224 IF0

```

Aufgabe 3 Zum Nachdenken für daheim: Count-to-Infinity

Gegeben sei die in Abbildung 2 dargestellte Topologie mit den vier Routern A bis D. Die Linkkosten sind jeweils an den Kanten angegeben. Wir notieren die Routingtabellen im folgenden Format:

Schritt	Router A	Router B	Router C	Router D
0	[(0,A) (/,/ (/,/ (/,/]	[(/,/ (0,B) (/,/ (/,/]	[(/,/ (/,/ (0,C) (/,/]	[(/,/ (/,/ (/,/ (0,D)]

Die Tupel (x, y) geben dabei die Kosten sowie den Next-Hop zum Ziel an. Das Ziel sei implizit durch die Reihenfolge gegeben, d. h. jeder Vektor in der Tabelle gibt Distanz und Next-Hop zu den Zielen A, B, C und D in dieser Reihenfolge an. Zu Beginn seien die Routingtabellen noch leer, d. h. die Router kennen noch nicht einmal ihre direkten Nachbarn. Dies wird durch die Schreibweise $(/, /)$ angedeutet. Sich selbst erreichen die Router natürlich mit Kosten 0.

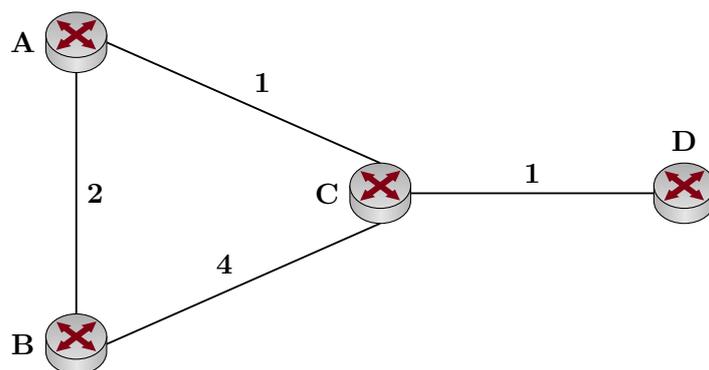


Abbildung 2: Netztopologie

Die Router beginnen nun damit, in periodischen Zeitabständen ihre Distanz-Vektoren mit ihren direkten Nachbarn auszutauschen. Dabei schickt beispielsweise Router B ein Update an Router C,

welches lediglich die Distanz zum jeweiligen Ziel enthält (nicht aber den Next-Hop). Wenn nun Router A ein solches Update von B erhält und darin eine Route zu D finden würde, so wüsste A, dass er D über B erreicht. Die Kosten zu D entsprechen dann den Kosten zu B zuzüglich der Kosten, mit denen B das Ziel erreichen kann.

Im Folgenden wollen wir dieses Verhalten untersuchen. Da das Ergebnis allerdings davon abhängt, in welcher Reihenfolge Updates ausgetauscht werden, treffen wir die idealisierte Annahme, dass alle Router exakt zeitgleich ihre Updates verschicken.

a)* Geben Sie gemäß obiger Tabelle die Routingtabellen aller vier Router in den folgenden Schritten an. Brechen Sie ab, sobald ein konvergenter Zustand erreicht ist.

Schritt	Router A	Router B	Router C	Router D
0	[(0,A) (/,) (/,) (/,)]	[(/,) (0,B) (/,) (/,)]	[(/,) (/,) (0,C) (/,)]	[(/,) (/,) (/,) (0,D)]
1	[(0,A) (2,B) (1,C) (/,)]	[(2,A) (0,B) (4,C) (/,)]	[(1,A) (4,B) (0,C) (1,D)]	[(/,) (/,) (/,) (1,C) (0,D)]
2	[(0,A) (2,B) (1,C) (2,C)]	[(2,A) (0,B) (3,A) (5,C)]	[(1,A) (3,A) (0,C) (1,D)]	[(2,C) (5,C) (1,C) (0,D)]
3	[(0,A) (2,B) (1,C) (2,C)]	[(2,A) (0,B) (3,A) (4,A)]	[(1,A) (3,A) (0,C) (1,D)]	[(2,C) (4,C) (1,C) (0,D)]

b) Welcher (Graph-)Algorithmus findet hier Verwendung?

Es kommt eine verteilte (dezentrale) Implementierung des Algorithmus von Bellman-Ford zum Einsatz.

Nun fällt die Verbindung zwischen den Knoten C und D aus. Die Knoten C und D bemerken dies und setzen die entsprechenden Pfadkosten auf unendlich.

c) Was passiert in den folgenden Schritten, in denen die aktiven Knoten weiter ihre Distanzvektoren austauschen? Geben Sie nach jedem Schritt die Distanztabelle an, bis das weitere Ergebnis klar ist.

Schritt	Router A	Router B	Router C	Router D
4	[(0,A) (2,B) (1,C) (2,C)]	[(2,A) (0,B) (3,A) (4,A)]	[(1,A) (3,A) (0,C) (/,)]	[(/,) (/,) (/,) (0,D)]
5	[(0,A) (2,B) (1,C) (6,B)]	[(2,A) (0,B) (3,A) (4,A)]	[(1,A) (3,A) (0,C) (3,A)]	[(/,) (/,) (/,) (0,D)]
6	[(0,A) (2,B) (1,C) (4,C)]	[(2,A) (0,B) (3,A) (7,C)]	[(1,A) (3,A) (0,C) (7,A)]	[(/,) (/,) (/,) (0,D)]
7	[(0,A) (2,B) (1,C) (8,C)]	[(2,A) (0,B) (3,A) (6,A)]	[(1,A) (3,A) (0,C) (5,A)]	[(/,) (/,) (/,) (0,D)]
8	[(0,A) (2,B) (1,C) (6,C)]	[(2,A) (0,B) (3,A) (9,C)]	[(1,A) (3,A) (0,C) (9,A)]	[(/,) (/,) (/,) (0,D)]
9	[(0,A) (2,B) (1,C) (10,C)]	[(2,A) (0,B) (3,A) (8,A)]	[(1,A) (3,A) (0,C) (7,A)]	[(/,) (/,) (/,) (0,D)]
10	[(0,A) (2,B) (1,C) (8,C)]	[(2,A) (0,B) (3,A) (11,C)]	[(1,A) (3,A) (0,C) (11,A)]	[(/,) (/,) (/,) (0,D)]
⋮	⋮	⋮	⋮	⋮

d)* In der Vorlesung wurden **Split Horizon**, **Triggered Updates** und **Path Vector** als mögliche Gegenmaßnahmen für das Count-to-Infinity-Problem genannt. Erläutern Sie in der Gruppe die Funktionsweise dieser Verfahren.

- **Split Horizon**

„Bewerbe eine Route nicht über das Interface, über das die Route ursprünglich gelernt wurde.“
Im konkreten Fall würden also Router A und B keine Route zu D an C schicken, womit in diesem Beispiel zumindest kein Count-to-Infinity mehr auftreten kann.

- **Triggered Update**

Anstelle Routing-Updates nur in periodischen Zeitabständen zu versenden (bei RIP standardmäßig 30s), werden Updates sofort geschickt, wenn Topologieänderungen erkannt werden. Dies löst das Count-to-Infinity-Problem zwar nicht, beschleunigt den Vorgang aber. Das Problem besteht darin, dass kurzzeitig viel Datenverkehr durch Routingupdates verursacht wird. Triggered Updates werden von den meisten Routingprotokollen unterstützt (RIP erst ab Version 2).

- **Path Vector**

Es wäre möglich, dass sich jeder Router bei einem Update merkt, woher das Update kam und diese Information bei Routing-Updates mit einschließt. Auf diese Weise könnte jeder Router den vollständigen Pfad nachvollziehen, den ein Paket zum jeweiligen Ziel nehmen wird. Entdeckt ein Router sich selbst in diesem Pfad, so weiß er, dass eine Schleife vorhanden ist. Er kann das Update in diesem Fall verwerfen. Dieses Verfahren wird von BGP eingesetzt.