Technische Universität München
Informatik VIII
Netzarchitekturen und Netzdienste
Prof. Dr.-Ing. Georg Carle

Heiko Niedermayer, Marc Fouquet

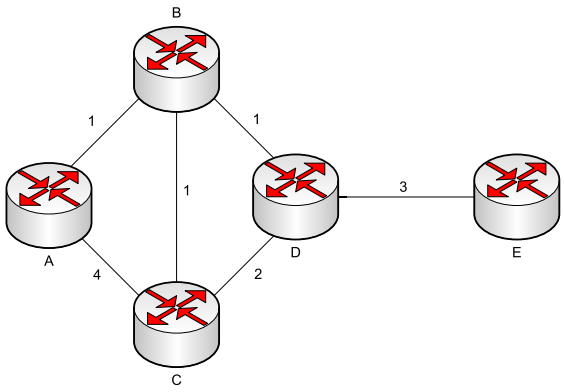
Übungen zur Vorlesung Rechnernetze und Verteilte Systeme

Übungsblatt 6, SS 2010

Abgabe: 14. Juni 2010 (vor der Vorlesung **bis 14:10 Uhr**)

Aufgabe 16 - Distance-Vector-Routing (13 Punkte + 3 Bonuspunkte = 16 Punkte)

Gegeben sei die folgende Topologie mit fünf Knoten A bis E. An den Kanten sind die jeweiligen Kosten angegeben.



Die folgende Tabelle zeigt die Distanzmatrix, die aus den Distanzvektoren der Knoten aufgebaut ist. Jede Zeile enthält die geringsten Pfadkosten zu den verschiedenen Zielen, die einem Knoten aktuell bekannt sind. Die Pfadkosten eines Knotens zu sich selbst sind Null, die Kosten zu allen anderen Knoten sind zunächst unendlich.

	A	B	C	D	E
A	0	∞	∞	∞	∞
B	∞	0	∞	∞	∞
C	∞	∞	0	∞	∞
D	∞	∞	∞	0	∞
E	∞	∞	∞	∞	0

Die Knoten beginnen nun, untereinander Distance-Vector-Routing-Information auszutauschen. Gehen Sie davon aus, dass der Austausch schrittweise verläuft und in jedem Schritt alle Knoten gleichzeitig ihren aktuellen Distanzvektor an ihre Nachbarn weitergeben.

Im ersten Schritt übernehmen die Knoten die Kosten zu ihren direkten Nachbarn (aus lokaler Konfiguration). Anschließend tauschen sie ihre Distanzvektoren (Zeilen in der Matrix) mit ihren direkten Nachbarn aus. Jeder Empfänger addiert die Kantenkosten zu den empfangenen Distanzvektoren hinzu und übernimmt diejenigen Pfade, für die die Pfadkosten zu den gegebenen Zielen eine Verbesserung darstellen.

a) Geben Sie die Distanzmatrix nach jedem Schritt an, bis Konvergenz erreicht ist.

b) Geben Sie für Knoten A eine Routingtabelle nach folgendem Schema an:

Ziel	Nächster Hop	Kosten
...	...	...

Nun soll ein Paket der Länge L Byte von Knoten A zu Knoten C übertragen werden. Die Übertragungsrate auf den Kanten sei $\frac{10}{k}$ Mbit/s, wobei k der jeweilige Wert der Kantenkosten ist. Um eine Weiterleitungsentscheidung zu treffen, benötigt jeder Knoten die Verarbeitungszeit T_V pro Paket. Zusätzliche Pufferzeiten treten im betrachteten Fall nicht auf, weil keine anderen Pakete unterwegs sind. Die Vermittlung erfolgt nach dem Store-and-Forward-Prinzip. Die Ausbreitungsverzögerung auf allen Kanten ist T_A.

c) Wie viel Zeit vergeht zwischen dem Sendebeginn bei Knoten A und dem vollständigen Empfang des Paketes durch Knoten C?

d) Wie lange dauert es, wenn Knoten A das Paket direkt an Knoten C schicken würde?

e) Für welche Paketlängen L führt das Routing über den Pfad mit den geringsten Kosten zu einer schnelleren Übertragung als die Direktverbindung zwischen A und C?
Nehmen Sie hierfür an, dass $T_V = T_A = 1$ms ist.

f) Bisher wurden die Kantenkosten mit der Übertragungsrate in Bezug gesetzt.
Welche anderen technischen, topologischen und verkehrsspezifischen Kriterien könnten sinnvollerweise in die Kantenkosten einfließen?

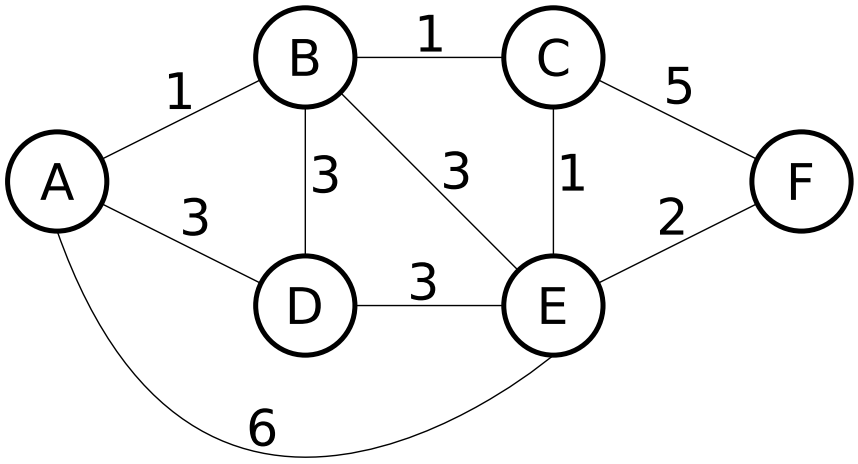
Nun fällt die Verbindung zwischen den Knoten D und E aus. Die Knoten D und E bemerken dies und setzen die entsprechenden Pfadkosten auf unendlich.

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

h) In der Vorlesung wurden *Split Horizon*, *Triggered Updates*, und *Path Vector* als mögliche Gegenmaßnahmen für das Count to Infinity-Problem genannt. Informieren Sie sich über die Funktionsweise eines dieser Verfahren und erläutern Sie es.

Aufgabe 17 - Link State Routing - Algorithmus von Dijkstra (7 Punkte)

Wie in der Vorlesung beschrieben, werden beim Link State Routing von jedem Router Informationen über seine direkte Nachbarschaft (Link State = Zustand der Links des Routers) an alle anderen Router des Netzes geflutet. Am Ende dieses Prozesses kennt jeder Router einen Topologiegraphen des gesamten Netzes. Das folgende Bild zeigt ein Beispiel für einen solchen Graphen, die Zahlen an den Kanten geben dabei die Link-Kosten an.



In dieser Aufgabe beschäftigen wir uns mit den Schritten die notwendig sind, um aus diesem Topologiegraphen eine Routingtabelle zu erstellen. Zunächst muss ein Router dafür einen minimalen Spannbaum in diesem Graphen finden, d.h. einen Baum mit dem Router als Wurzel, in dem jeder Knoten mit minimalen Kosten erreicht wird. Dazu wird üblicherweise der Algorithmus von Dijkstra verwendet, den Sie im folgenden Listing finden:

```
1  DIJKSTRA
2
3  INPUT
4  Q : set of all nodes
5  A : start node
6
7  OUTPUT
8  p[V] : predecessor of each node V in the shortest-path tree
9
10 NOTATION
11 c(U,V) : link costs from node U to node V,
12          infinity if there is no direct connection
13 d[V]   : distance from root A to node V in the shortest-path tree
14 p[V]   : predecessor of V in the shortest path tree
15
16 BEGIN
17     FOR all nodes V in Q DO d[V]=inf; p[V]=undef; ENDFOR
18
19     d[A]=0;
20     p[A]=null;
21
22     N={A}; // MinHeap regarding d[]
23
24     WHILE N.notEmpty() DO
25         V = N.popMinimum();                 // get node in N with lowest d
26         FOR all nodes U adjacent to V DO
27             IF d[U] > d[V]+c(V,U) THEN
28                 d[U] = d[V]+c(V,U);
29                 p[U] = V;
30                 N.addOrUpdate(U);           // add U to N or
31                                             // update d if U was already in N
32             ENDIF
33         ENDFOR
34     ENDWHILE
35 END
```

a) Verwenden Sie den oben angegebenen Algorithmus um einen Shortest Path Tree in der Beispieltopologie zu finden. **Geben Sie dabei in einer Tabelle für jeden Durchlauf der WHILE-Scheife die Inhalte von N, d und p an**.

Step	N	A	B	C	D	E	F
0	{A}	d=0, p=null	d=inf, p=undef	d=..., p=...	...	...	...

b) Geben Sie einen Algorithmus in Pseudocode an, der mit Hilfe der Datenstruktur p (Vorgänger jedes Knotens im Shortest Path Tree) eine Routingtabelle nach folgendem Schema erstellt:

Ziel	Nächster Hop
A	null
B	B
C	B
...	...

Aufgabe 18 - Unterschiede zwischen Distance-Vector- und Link-State-Routing (4 Punkte)

Welches sind die wichtigsten Unterschiede zwischen Distance-Vector- und Link-State-Routing? Beschreiben Sie die Funktionsweise beider Verfahren mit Ihren eigenen Worten. Welche Informationen werden jeweils ausgetauscht? An wen werden diese Informationen verschickt? Wie sind Rechenaufwand und Konvergenzeigenschaften zu bewerten?