Universität
Paderborn
Fachgebiet
Rechnernetze

Wintersemester 2019/2029

Rechnernetze

Übung 7: Internetworking

2019-12-05

---

1. **Virtual LANs**

   (a) **VLAN-Bridges** Einer der Hauptvorteile von Bridges ist die autoconfiguration der Forwarding Tabellen. Funktioniert dies auch noch, wenn die Bridge VLANs weiterleiten soll? (Nehmen Sie an, dass alle Geräte im LAN IEEE 802.1q implementieren.)

   Lösung: Klar – separate Behandlung je nach VLAN Tag. Ports werden zusätzlich automatisch nach VLAN-Tags eingefärbt. Details sind ähnlich den spanning tree Verfahren.

   (b) **Zustand in VLAN-Bridges?**

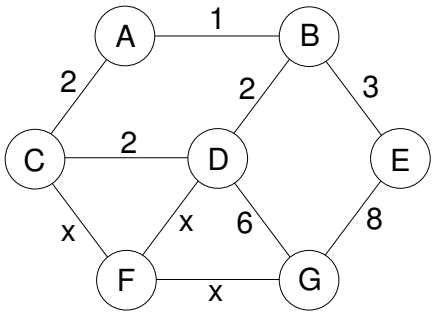
   In welchem Sinne wird durch IEEE 802.1q VLAN Zustand in Bridges bzw. Switches erzeugt? Ist dies eine Verletzung des Prinzips eines zustandslosen Netzes?

   Lösung: Ja, die VLAN-Tabellen kann man als Zustand auffassen, und die VLAN-Ids im Packet als Connection Identifier.

   Andererseits kann man argumentieren, dass dies lediglich eine erweiterte Zieladresse ist. Und es sich ja ohnehin bestenfalls um soft state handelt, der automatisch erzeugt wird. Es also keinen Verbindungsaufbau bzw. Zustandserzeugung und -beseitigung bedarf.

2. **Dijkstra-Algorithmus**

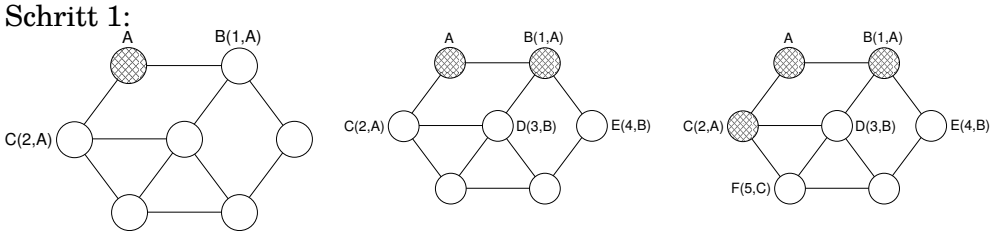
   Gegeben sei folgende Netzwerktopologie:



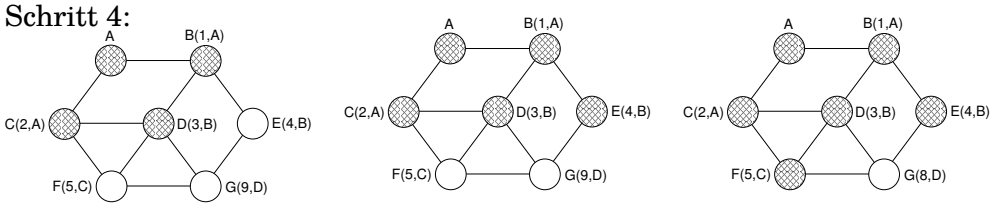
(a) Sei $x = 3$. Bestimmen Sie den kürzesten Weg von A nach G mit Hilfe des Dijkstra-Algorithmus. Geben Sie für jeden Iterationsschritt den Arbeitsknoten und die Knotenmarkierungen an.

Lösung: Iterationsschritte für $x = 3$:

Schritt 1:



Schritt 4:



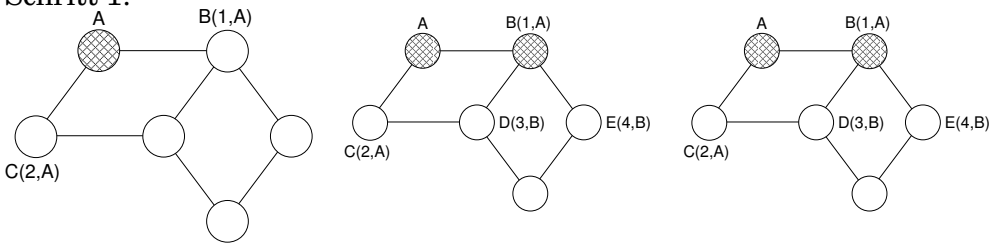
(b) Geben Sie einen Wert für x an, der den Ausfall des Knotens F modelliert.

Lösung: Knotenausfall: $x = \infty$

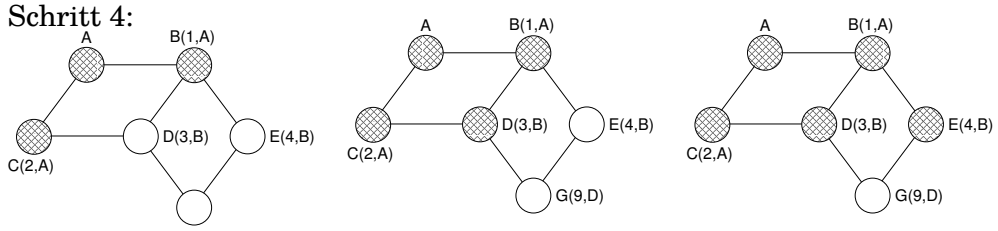
(c) Bestimmen Sie jetzt den kürzesten Weg von A nach G mit Hilfe des Dijkstra-Algorithmus unter der Annahme, dass Knoten F tatsächlich ausgefallen ist. Geben Sie für jeden Iterationsschritt den Arbeitsknoten und die Knotenmarkierungen an.

Lösung: Iterationsschritte für $x = \infty$:

Schritt 1:

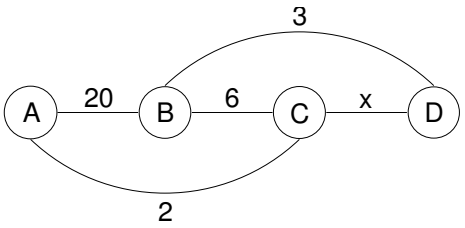


Schritt 4:



3. **Distanzvektor-Wegewahlverfahren (Bellman-Ford Routing)**

Gegeben sei folgende Netztopologie:



Die folgende Tabelle zeigt spaltenweise die initialen Distanzvektoren:

	A	B	C	D
A	$0 \rightarrow A$	-	-	-
B	-	$0 \rightarrow B$	-	-
C	-	-	$0 \rightarrow C$	-
D	-	-	-	$0 \rightarrow D$

(a) Führen Sie das Distanz-Vektorverfahren für $x = 10$ aus. Tragen Sie die endgültigen Distanzvektoren in die Tabelle ein. Notieren Sie sowohl die Kosten als auch den Nachbarn, über den der jeweils beste bekannte Weg verläuft.

Lösung:

1. Schritt:

	A	B	C	D
A	$0 \rightarrow A$	$20 \rightarrow A$	$2 \rightarrow A$	-
B	$20 \rightarrow B$	$0 \rightarrow B$	$6 \rightarrow B$	$3 \rightarrow B$
C	$2 \rightarrow C$	$6 \rightarrow C$	$0 \rightarrow C$	$x \rightarrow C$
D	-	$3 \rightarrow D$	$x \rightarrow D$	$0 \rightarrow D$

2. Schritt: Distanzvektoren für $x = 10$:

	A	B	C	D
A	$0 \rightarrow A$	$8 \rightarrow C$	$2 \rightarrow A$	$12 \rightarrow C$
B	$8 \rightarrow B$	$0 \rightarrow B$	$6 \rightarrow B$	$3 \rightarrow B$
C	$2 \rightarrow C$	$6 \rightarrow C$	$0 \rightarrow C$	$9 \rightarrow B$
D	$12 \rightarrow C$	$3 \rightarrow D$	$9 \rightarrow B$	$0 \rightarrow D$

3. Schritt: Distanzvektoren für $x = 10$:

	A	B	C	D
A	$0 \rightarrow A$	$8 \rightarrow C$	$2 \rightarrow A$	$11 \rightarrow B$
B	$8 \rightarrow C$	$0 \rightarrow B$	$6 \rightarrow B$	$3 \rightarrow B$
C	$2 \rightarrow C$	$6 \rightarrow C$	$0 \rightarrow C$	$9 \rightarrow B$
D	$11 \rightarrow C$	$3 \rightarrow D$	$9 \rightarrow B$	$0 \rightarrow D$

(b) Die Verbindung x verbessert sich zu $x = 1$. Führen Sie das Distance Vector-Verfahren erneut aus, basierend auf den Distanzvektoren aus Ihrer Lösung zu Teil 3a.

Lösung: 2. Schritt: Distanzvektoren für $x = 1$:

	A	B	C	D
A	$0 \to A$	$6 \to D$	$2 \to A$	$3 \to C$
B	$6 \to C$	$0 \to B$	$4 \to D$	$3 \to B$
C	$2 \to C$	$4 \to D$	$0 \to C$	$1 \to C$
D	$3 \to C$	$3 \to D$	$1 \to D$	$0 \to D$

4. **Count-to-Infinity Problem**

   Gegeben sei ein Netz der Form A—B—C—D—E, bei dem alle Kanten die Kosten 1 haben. Nehmen Sie an, der Knoten E sei zu Beginn abgeschaltet und alle anderen Knoten haben korrekte Routing-Tabellen.

   (a) Knoten E wird eingeschaltet. Zeigen Sie, wie die Routing-Tabellen in den anderen Knoten auf diese Änderung hin durch den Distance Vector Algorithmus geändert werden.

   (b) Was geschieht, wenn Knoten E wieder abgeschaltet wird (nehmen Sie an, dass zuvor der Algorithmus konvergiert ist.)? Wie reagiert der Algorithmus auf diese Änderung? Vergleichen Sie die Konvergenzgeschwindigkeit in beiden Fällen und erklären Sie, warum man hier vom *Count-to-Infinity* Problem spricht.

   Lösung: E wird eingeschaltet: D erhält als erstes die Information, dass E jetzt existiert und trägt in seiner Routing Tabelle eine Verbindung zu E mit Distanz 1 ein. D kommuniziert diese Information an C, der eine Verbindung zu E mit der Entfernung 2 einträgt. Entsprechend trägt B eine Verbindung mit Distanz 3 und A mit 4 ein. Es dauert also 4 Durchgänge, bis der Algorithmus konvergiert. Zu beachten ist, dass C u.a. an D kommuniziert, das es von C nach E eine Verbindung mit Distanz 2 gibt, so dass D nun zu E zwei Verbindungen kennt, mit Distanz 1 direkt zu E und mit Distanz 3 über C!
   E wird abgeschaltet: D verliert die direkte Verbindung zu E, denkt aber noch, dass es ja eine Verbindung mit Distanz 3 über C gibt. D kommuniziert also nun an C, dass die kürzeste Verbindung zu E nun die Distanz 3 hat, woraufhin C seine Routingtabelle ändert in die Distanz 4 zu E. Diese Information wird an D kommuniziert und veranlasst D, seine Distanz zu E (über C!) auf 5 zu ändern, u.s.w. Daher kommt auch der Name des Problems, da die Router sich gegenseitig bis „unendlich“ hochzählen.

5. Bearbeiten Sie das Notebook zu Bellmann-Ford Routing.