Routingprotokolle

In Netzwerktoplogien, die nicht "schleifenfrei" sind, muss es Möglichkeiten geben, mit Hilfe derer Router bestimmen können, welches der beste Weg zu einem Zielrechner ist, dabei müssen selbstverständlich auch Wege über mehrere andere Router berücksichtigt werden.

Es existieren mehrere Protokolle, mit denen Router Informationen austauschen können, um ihrerseits ihre Routingkonfiguration zu optimieren - diese Falle grob in zwei Kategorieren: **Distanzvektor** Routing-Protokolle und **Link-State** Routing-Protokolle.

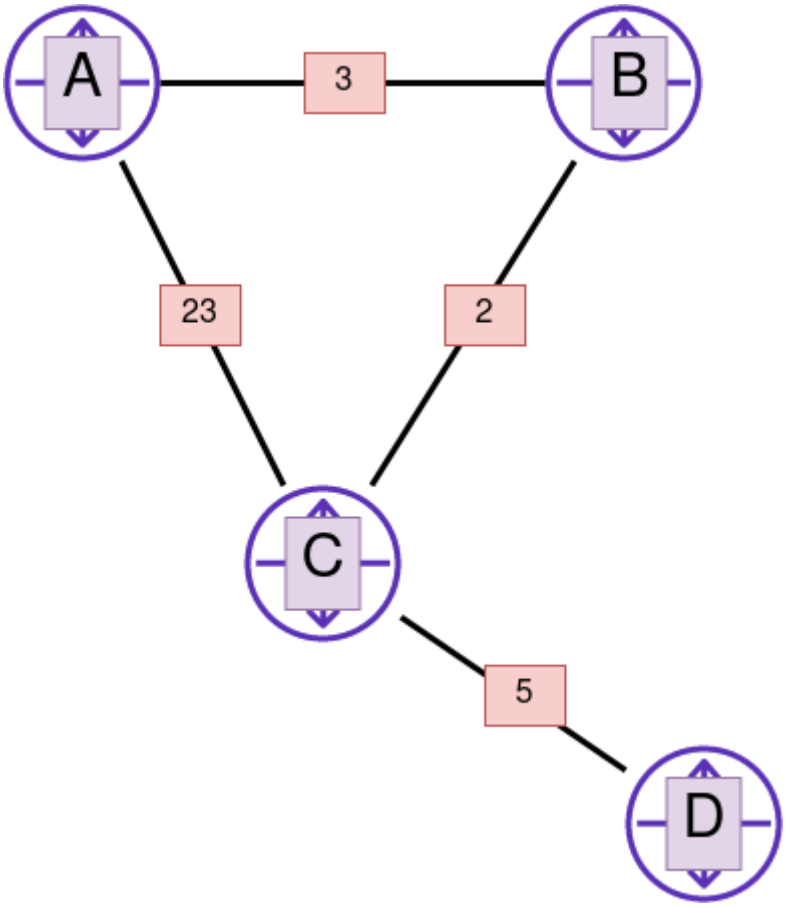
Link-State Protokolle verwenden den Dijkstra-Algorithmus. damit kann sich ein Router einen vollständigen Überblick über das ihn umgebende Netz verschaffen. Beispiele für Link-State Protokolle sind *OSPF (Open Shortest Path First)* oder *IS-IS (Intermediate System to Intermediate System Protocol)*. Da uns der Dijkstra-Algorithmus später ncohmal genauer beschäftigen wird, beschränken wir uns hier auf die Distanzvektor Protokolle.

Distanzvektor Routing-Protokolle basieren auf dem Bellman-Ford Algorithmus. Damit wird der "kürzeste" Weg ausgehend von einem Startknoten bestimmt, wobei die Verbindungen zwischen den Knoten verschiedene "Kosten" haben können. Der "kürzeste" oder "beste" Weg ist der, mit den geringsten Kosten.

Distanzvektor Routing-Protokolle geben in regelmäßigen Abständen (ca. alle 30 Sekunden und bei einer Änderung der Topologie) eine Kopie der eigenen Routing-Tabelle an ihren Nachbarn weiter. Auf diese Weise "wandern" Topologie-Änderungen durch das Netz.

Den Austausch der Routing-Tabellen kann man sich so vorstellen: Router A gibt seine Informationen zu Router B weiter. Router B gleicht die neuen Informationen von Router A mit den ihm bekannten Informationen ab und fügt schließlich seine Distanzvektorkosten (z.B. Anzahl der Hops) hinzu. Nun gibt Router B die „aktualisierte“ Routing-Tabelle an Router C weiter. Dieser Vorgang wiederholt sich bei jedem benachbarten Router.

Schritt für Schritt



T=0: Knoten A erzeugt seine initiale Kostenmatrix. Sie enthält nur unsere direkten Nachbarn B und C mit den uns bekannten Kosten. Wir schicken daraufhin unsere neuen besten Pfade (B mit Kosten 3, C mit Kosten 23) an unsere direkten Nachbarn. Die anderen drei Knoten machen dasselbe aus ihrer Sicht.

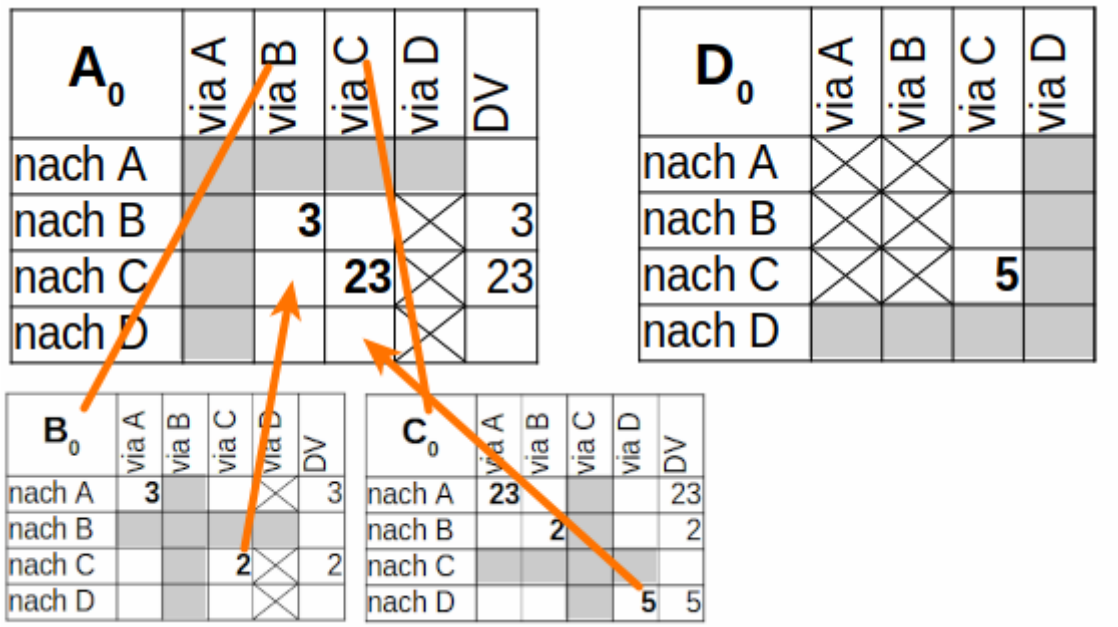
A_0	via A	via B	via C	via D	DV
nach A					
nach B		**3**			3
nach C			**23**		23
nach D					

B_0	via A	via B	via C	via D	DV
nach A	**3**				3
nach B					
nach C			**2**		2
nach D					

C_0	via A	via B	via C	via D	DV
nach A	**23**				23
nach B		**2**			2
nach C					
nach D				**5**	5

D_0	via A	via B	via C	via D
nach A				
nach B				
nach C			**5**	
nach D				

T=1: Router A hat von den Routern B und C Datenpakete erhalten und weiß jetzt, zu welchen Kosten er D und wie er C und B jeweils auch erreichen könnte. Im Fall der Zielrouter C und D ist das sogar ein neuer bester Pfad. Im nächsten Schritt überträgt Router A diese Information wieder an seine Nachbarn. Alle anderen Knoten machen dasselbe.



Damit ergibt sich nach dem ersten Schritt die folgende Situation.

A_1	via A	via B	via C	via D	DV
nach A					
nach B		**3**	25		3
nach C		**5**	23		5
nach D			28		28

B_1	via A	via B	via C	via D	DV
nach A	**3**		25		3
nach B					
nach C	26		**2**		2
nach D			**7**		7

C_1	via A	via B	via C	via D	DV
nach A	23	**5**			5
nach B	26	**2**			2
nach C					
nach D				**5**	5

D_1	via A	via B	via C	via D
nach A			**28**	
nach B			**7**	
nach C			**5**	
nach D				

(A1)

Vollziehe die Entstehung der Routingtabelle eines anderen Knotens (B,C oder D) nach. Von wem sind Infos beim Knoten angekommen? Welche Schlüsse können daraus gezogen werden?

T=2: Router A erhält wiederum von Router B ein neues Datenpaket und weiß jetzt, dass B den Router D günstiger erreichen kann. Wir tragen die Kosten in unsere Matrix ein und werden diesen neuen besten Pfad wieder an unsere Nachbarn verbreiten.

A_1	via A	via B	via C	via D	DV
nach A					
nach B		**3**	25		3
nach C		**5**	23		5
nach D			28		28

B_1	via A	via B	via C	via D	DV
nach A	**3**		25		3
nach B					
nach C	26		**2**		2
nach D			**7**		7

C_1	via A	via B	via C	via D	DV
nach A	23	**5**			5
nach B	26	**2**			2
nach C					
nach D				**5**	5

Nach Schritt 2 sieht die Situation also so aus:

A_2	via A	via B	via C	via D	DV
nach A					
nach B		**3**	25		3
nach C		**5**	23		5
nach D		**10**	28		10

B_2	via A	via B	via C	via D	DV
nach A	**3**		7		3
nach B					
nach C	8		**2**		2
nach D	31		**7**		7

C_2	via A	via B	via C	via D	DV
nach A	23	**5**		33	5
nach B	26	**2**		12	2
nach C					
nach D	51	9		**5**	5

D_2	via A	via B	via C	via D
nach A			**10**	
nach B			**7**	
nach C			**5**	
nach D				

Jetzt hat jeder Router eine Tabelle, aus der hervorgeht, welches der kürzeste Distanzvektor zu einem anderen Ziel im Netz ist und kann anhand dessen feststellen, an welchen Router er ein Datenpaket als nächstes schicken muss.

(A2)

Überprüfe, dass sich im Zeitschritt **T=3** für die kürzesten Routen keine neuen Erkenntnisse mehr ergeben.

(A3)

Vollziehe nach dem Schema oben schrittweise nach, was auf Router A passiert wenn sich die Kosten der Verbindung zwischen B und C auf 25 erhöhen.

Material

01_dvalgo.odp	151.7 KiB	15.12.2021 16:52
01_dvalgo.pdf	118.2 KiB	15.12.2021 16:52
auswahl_030.png	37.0 KiB	12.10.2020 18:10
dv.png	15.2 KiB	12.10.2020 17:37
dva1.png	31.5 KiB	12.10.2020 17:49
dva2.png	37.0 KiB	12.10.2020 18:02
dva3.png	38.1 KiB	12.10.2020 18:12
transfer1.png	54.3 KiB	12.10.2020 18:00
transfer2.png	45.0 KiB	12.10.2020 18:10

Beispiel: https://de.wikipedia.org/wiki/Distanzvektoralgorithmus

From:
https://www.info-bw.de/ -

Permanent link:
https://www.info-bw.de/faecher:informatik:oberstufe:netzwerke:wegedurchsnetz_i:start?rev=1635445675

Last update: **28.10.2021 18:27**