

Kommunikationssysteme WS24/25

Übungsblatt 3

Maximilian Amthor

Aufgabe 3.1

Aufgabe 3.1

- ❑ Beschreiben Sie die folgenden Komponenten:
 - ❖ Fiber Modem
 - ❖ Repeater
 - ❖ Bridge
 - ❖ Switch

- ❑ Welche Unterschiede gibt es zwischen den Komponenten?

Lösung zu 3.1

□ **Fiber Modem:**

- ❖ Verbindung eines Computers mit einem entfernten LAN mit Hilfe eines Paares von Fiber Modems und Glasfaserkabel
- ❖ Konvertierung von AUI-Signalen in digitale Signale, Übertragung über das Glasfaserkabel, Rekonvertierung der Signale
- ❖ Keine Änderung der eigentlichen Netzwerkhardware erforderlich
- ❖ Geringer Delay und hohe Bandbreite bei Übertragung über Glasfaserkabel, sodass Distanzen von einigen Kilometern überbrückt werden können

Lösung zu 3.1

□ Repeater:

- ❖ Verbindung zweier Segmente = Verdopplung der Segmentlänge ($2 * 500\text{m} = 1000\text{m}$)
- ❖ Bei Erhalt von Signal aus einem Segment: Erneuerung des Signals und Übertragung in das andere Segment
- ❖ Kein Verständnis von Frameformaten, Kollisionen werden übertragen (nicht in einem Segment isoliert)
- ❖ CSMA/CD-Verfahren beschränkt maximale Verlängerung, da Verzögerungen zu groß werden können
- ❖ Maximal 4 Repeater zwischen zwei kommunizierenden Stationen

Lösung zu 3.1

□ **Bridge:**

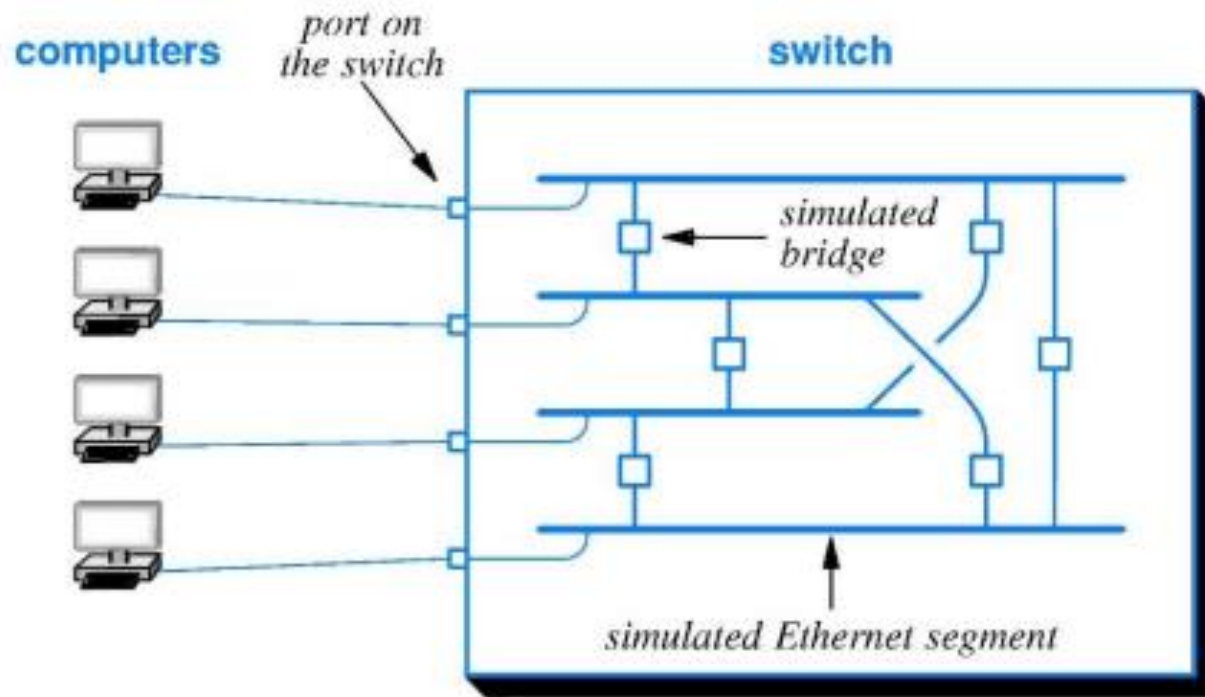
- ❖ Verbindung zweier LAN-Segmente
- ❖ Behandlung aller eingehenden Frames, Verwendung eines Network Interface
- ❖ Fehlerkontrolle der Frames
- ❖ Frame-Filterung, Weiterleitung in anderes Segment nur falls Zielcomputer nicht im selben Segment oder Bridge Zielcomputer nicht kennt
- ❖ Isolation von Problemen (Kollisionen, Interferenzen) in einem Segment

Lösung zu 3.1

□ Switch:

- ❖ Hardware zum Transfer von Frames zwischen vielen Computern, ein Port für jeden Computer
- ❖ Keine Simulation eines geteilten Mediums, sondern Simulation eines bridged LAN mit einem Computer pro Segment
- ❖ Parallele Datenübertragung, maximaler Durchsatz ist $R \cdot N / 2$ (R = Datenübertragungsrate eines Computers, N = Anzahl der Computer)

Konzept einer Switch



Aufgabe 3.2

Aufgabe 3.2

- ❑ LAN mit Bridges
- ❑ Paket wird an nicht existierende Adresse gesendet
- ❑ Wie weit wird es durch die Bridges weitergeleitet?

Lösung zu 3.2

- Das Senden eines Pakets zu einer nicht existierenden Adresse entspricht dem Verhalten eines mit Bridges versehenen LANs beim Start
 - ❖ Hardwareadressen der Computer in den einzelnen Segmenten nicht bekannt

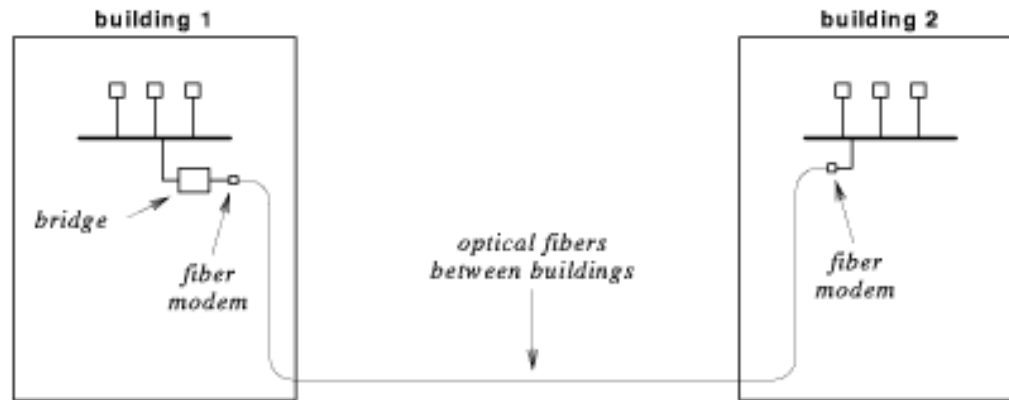
- Paket würde durch das gesamte Netzwerk weitergeleitet werden
 - ❖ In der Annahme, dass sich das Ziel im anschließenden Segment befindet

Aufgabe 3.3

Aufgabe 3.3

- ❑ Um zwei räumlich voneinander entfernte Gruppen von Computern eines LAN zu verbinden, wird ein Glasfaserkabel eingesetzt, an dessen Enden sich optische Modems befinden
- ❑ Warum ist es sinnvoll, zusätzlich eine Bridge einzufügen?

Lösung zu 3.3



- ❑ Filterfunktion der Bridge gestattet Datenverkehr der beiden Computergruppen untereinander
 - ❖ mit der Bridge werden zwei Segmente gebildet
- ❑ Ohne Bridge wäre eine gleichzeitige Kommunikation innerhalb der Computergruppen nicht möglich

Aufgabe 3.4

Aufgabe 3.4

- Folgende Tabelle gibt an, wie viele Bridges ein Switch bei einer Anzahl von 1 ... 5 Ports simuliert:

Anzahl Ports	1	2	3	4	5
Anzahl simulierter Bridges	1	1	3	6	10

- Entwickeln Sie eine Formel, die die Anzahl der simulierten Bridges bei gegebener Anzahl von Ports berechnet!

Lösung zu 3.4

- Sei B_n die Anzahl der simulierten Bridges bei einer Anzahl von $n > 1$ Ports

dann gilt:

$$B_n = \sum_{i=1}^{n-1} i = \frac{n \cdot (n - 1)}{2}$$

- Intuition: Eine simulierte Bridge für jedes mögliche Paar aus n Ports

$$B_n = \binom{n}{2} = \frac{n \cdot (n - 1)}{2}$$

Aufgabe 3.5

Aufgabe 3.5

- Nennen Sie Unterscheidungsfaktoren, die die WAN-Technologie von der LAN-Technologie abgrenzen

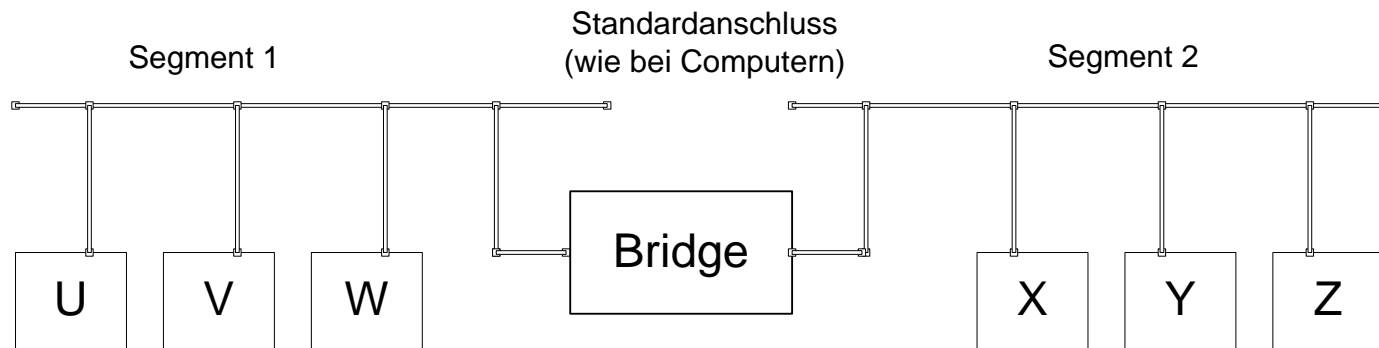
Lösung zu 3.5

- ❑ LAN kann zwar in Reichweite vergrößert werden, (Bridges + Satellit) deshalb aber noch kein WAN!
- ❑ Unterscheidungsfaktoren:
 - ❖ Skalierbarkeit, d.h. beliebiges Wachstum sollte möglich sein
 - ❖ Technologie sollte angemessene Leistung für große Netzwerke bieten
 - ❖ (beides im LAN nicht gegeben)

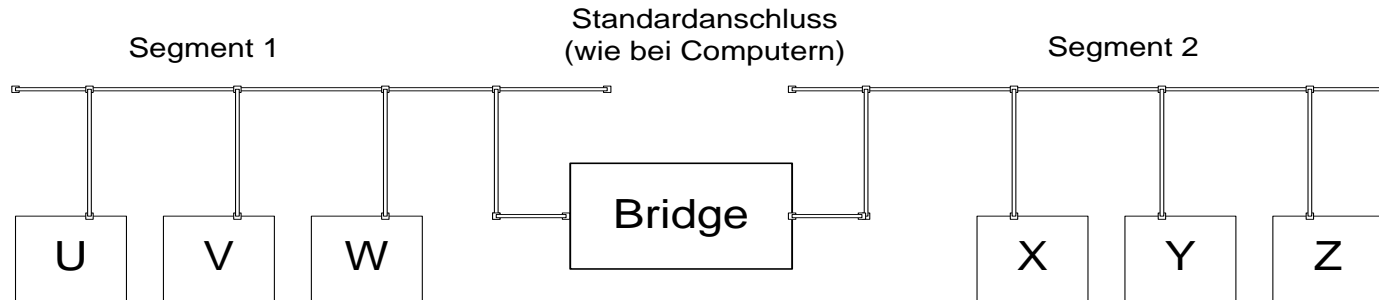
Aufgabe 3.6

Aufgabe 3.6

- Geben Sie die Adresslisten der Bridge für Segment 1 und Segment 2 im abgebildeten LAN an. Führen Sie in jeder Zeile die Computer auf, die die Bridge entsprechend der angegebenen Ereignisse bereits in ihrer Liste erfasst hat

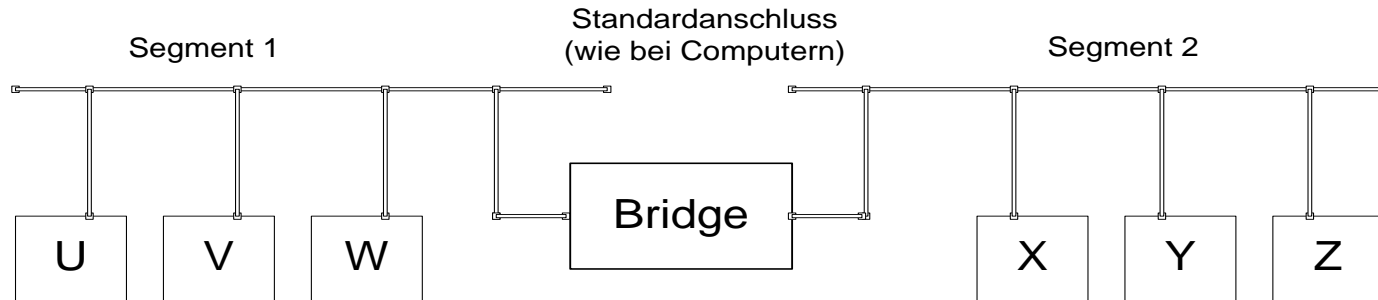


Lösung zu 3.6



Ereignis	Liste von Segment 1	Liste von Segment 2
Anschaltung der Bridge		
U sendet an V		
V sendet an U		
Z überträgt Broadcasting		
Y sendet an V		
Y sendet an X		
X sendet an W		
W sendet an Z		

Lösung zu 3.6



Ereignis	Liste von Segment 1	Liste von Segment 2
Anschaltung der Bridge	-	-
U sendet an V	U	-
V sendet an U	U, V	-
Z überträgt Broadcasting	U, V	Z
Y sendet an V	U, V	Z, Y
Y sendet an X	U, V	Z, Y
X sendet an W	U, V	Z, Y, X
W sendet an Z	U, V, W	Z, Y, X

Aufgabe 3.7

Aufgabe 3.7

- Erläutern Sie die folgenden Begriffe:
 - ❖ Store and Forward
 - ❖ Next-Hop Forwarding
 - ❖ Quellenunabhängigkeit (Source Independence)

Lösung zu 3.7

□ **Store and Forward:**

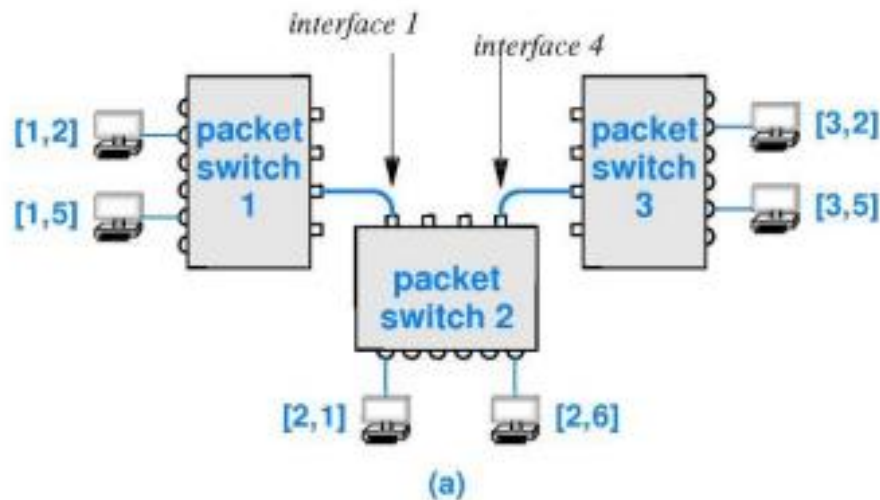
- ❖ Paradigma nach dem Packet Switches in einem WAN arbeiten
- ❖ Speichern von eingehenden Paketen
- ❖ Weiterleitung der Pakete durch Informieren des Prozessors und Bestimmung des Interface, über das das Paket zu seinem Ziel gesendet wird
- ❖ Möglichkeit zur Abspeicherung von Paketen in Warteschlangen, wenn die Verbindung belegt ist

Lösung zu 3.7

□ **Next-Hop Forwarding:**

- ❖ Paradigma nach dem Packet Switches in einem WAN arbeiten
- ❖ Bestimmung des Ziels eingehender Pakete anhand der Zieladresse
- ❖ Packet Switch hat keine kompletten Informationen über alle möglichen Zieladressen, sondern kennt nur den nächsten Hop, über den das Paket schließlich sein Ziel erreichen kann

Lösung zu 3.7



to reach	send to
switch 1	interface 1
switch 2	local delivery
switch 3	interface 4

- (a) Netzwerk mit drei Packet Switches
(b) Next-Hop Weiterleitungstabelle

Lösung zu 3.7

□ Quellenunabhängigkeit:

- ❖ Paradigma nach dem Packet Switches in einem WAN arbeiten
- ❖ Nächster Hop hängt nicht von Quelle des Pakets oder dem bisher gewählten Pfad ab, sondern nur vom Ziel des Pakets
- ❖ Erlaubt schnelles und effizientes Routing, da nur die Zieladresse betrachtet werden muss und nur ein Mechanismus für das Forwarding benötigt wird

Aufgabe 3.8

Aufgabe 3.8

- a) Erklären Sie das Routing in einem WAN
- b) Berechnen Sie mit Dijkstra's Algorithmus die kürzesten Pfade im Gegebenen Graphen (Quelle: Knoten 7)
 - ❖ Geben Sie für jeden Schritt die Datenstrukturen
 - S (Knoten)
 - D (Kantengewichte von der Quelle aus)
 - R (Next Hop von der Quelle aus)
- c) Nennen Sie charakteristische Eigenschaften des Distance Vector Routings und des Link State Routings

Lösung zu 3.8 a)

□ Routing:

- ❖ Prozess der Weiterleitung in einem Netzwerk
- ❖ Erfordert Routingtabelle, in der für jedes mögliche Ziel der nächste Hop bekannt ist (universelles Routing)
- ❖ Routingtabellen können default-Routen beinhalten, wenn alle Ziele denselben nächsten Hop haben
- ❖ Nächster Hop in der Tabelle muss auf dem kürzesten Weg zum Ziel sein (optimales Routing)

Lösung zu 3.8 a)

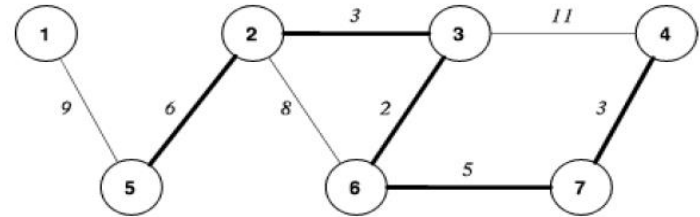
□ Routing:

- ❖ Netzwerk wird als Graph modelliert (Knoten = Packet Switches, Kanten = Verbindung zwischen Packet Switches, Gewicht der Kanten = Kosten (Zeit, Geld, Hop-Anzahl))
- ❖ Nutze Graphalgorithmus (Dijkstra Algorithmus) zur Berechnung der kürzesten Pfade (Pfad mit geringstem Gewicht) = Bestimmung des nächsten Hop bei optimalem Routing

Aufgabe 3.8

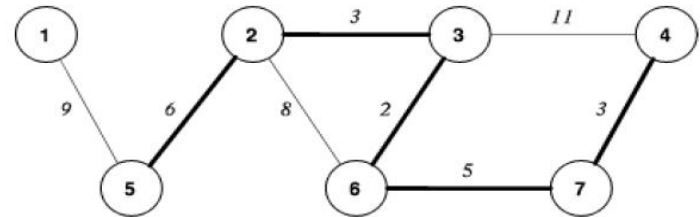
- a) Erklären Sie das Routing in einem WAN
- b) Berechnen Sie mit Dijkstra's Algorithmus die kürzesten Pfade im Gegebenen Graphen (Quelle: Knoten 7)
 - ❖ Geben Sie für jeden Schritt die Datenstrukturen
 - S (Knoten)
 - D (Kantengewichte von der Quelle aus)
 - R (Next Hop von der Quelle aus)
- c) Nennen Sie charakteristische Eigenschaften des Distance Vector Routings und des Link State Routings

Lösung zu 3.8 b)



- ❑ Dijkstra Algorithmus (Initialisierung):
 - ❖ Quelle = Knoten 7
 - ❖ $S = \{1, 2, 3, 4, 5, 6\}$ Array aller Knoten außer der Quelle
 - ❖ Infinity = Summe aller Kantengewichte + 1 = 47+1=48
 - ❖ $D = \{48, 48, 48, 3, 48, 5\}$ Array der Gewicht der Kanten von der Quelle zum entsprechenden Knoten aus S falls existent, sonst Infinity
 - ❖ $R = \{0, 0, 0, 4, 0, 6\}$ Array des nächsten Hop von der Quelle zum entsprechenden Knoten aus S falls existent, sonst 0

Lösung zu 3.8 b)



Umgangssprachlicher Algorithmus:

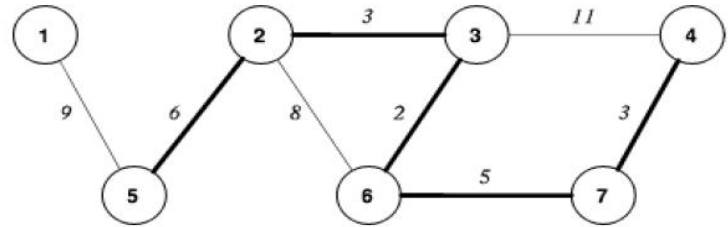
- ❖ So lange S nicht leer ist (Abbruchkriterium):
- ❖ wähle einen Knoten u aus S , der das kleinste Gewicht in D hat.
- ❖ Brich ab, falls das kleinste Gewicht = Infinity ist (keine existierende Kante).
- ❖ Entferne den Knoten u aus S .
- ❖ Berechne für jeden Knoten v , für den es eine Kante (u,v) gibt und der noch in S ist:
 - $C = D[u] + G(u,v)$ wobei $G(u,v) = \text{Gewicht der Kante } (u,v)$
 - Falls C kleiner als Gewicht von $v = D[v]$
 - $R[v] = R[u]$ next Hop nach $v = \text{next Hop nach } u$
 - $D[v] = C$ $G(\text{Quelle}, v) = G(\text{Quelle}, u) + G(u, v)$

Lösung zu 3.8 b)

Dijkstra Algorithmus

Schritt 1

$S = \{1,2,3,4,5,6\}$ $D = \{48,48,48,3,48,5\}$ $R = \{0, 0, 0, 4, 0, 6\}$

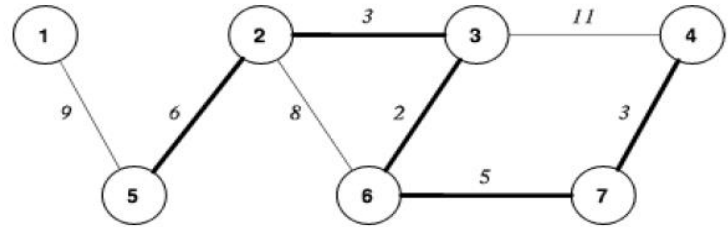


1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 4 aus S → $S = \{1,2,3,5,6\}$
 2. Prüfe jeden Knoten v , für den es $(4, v)$ gibt, die Distanz über Knoten 4: → Knoten 3
 - $C = D[4] + G(4, 3) = 3 + 11 = 14$
 - $C = 14 < D[3] = 48$
→ $R[3] = R[4] = 4$ und $D[3] = C = 14$
- $D = \{48, 48, 14, 3, 48, 5\}$ $R = \{0, 0, 4, 4, 0, 6\}$

Lösung zu 3.8 b)

Dijkstra Algorithmus Schritt 2

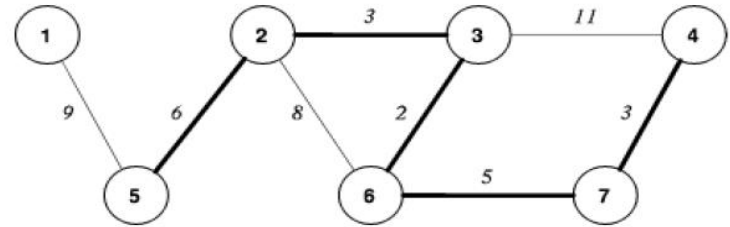
$S = \{1, 2, 3, 5, 6\}$ $D = \{48, 48, 14, 3, 48, 5\}$ $R = \{0, 0, 4, 4, 0, 6\}$



1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 6 aus S → $S = \{1, 2, 3, 5\}$
2. Prüfe jeden Knoten v , für den es $(6, v)$ gibt, die Distanz über Knoten 6: → Knoten 2 und 3
 - $C = D[6] + G(6, 2) = 5 + 8 = 13$
 - $C = 13 < D[2] = 48$
→ $R[2] = R[6] = 6$ und $D[2] = C = 13$

Lösung zu 3.8 b)

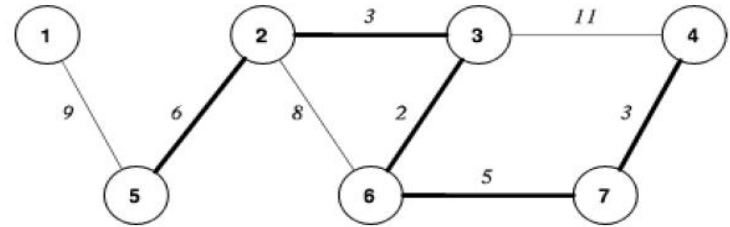
Dijkstra Algorithmus Schritt 2



$S = \{1, 2, 3, 5, 6\}$ $D = \{48, 48, 14, 3, 48, 5\}$ $R = \{0, 0, 4, 4, 0, 6\}$

1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 6 aus S → $S = \{1, 2, 3, 5\}$
 2. Prüfe jeden Knoten v , für den es $(6, v)$ gibt, die Distanz über Knoten 6:
→ Knoten 2 und 3
 - $C = D[6] + G(6, 3) = 5 + 2 = 7$
 - $C = 7 < D[3] = 14$
→ $R[3] = R[6] = 6$ und $D[3] = C = 7$
- $D = \{48, 13, 7, 3, 48, 5\}$ $R = \{0, 6, 6, 4, 0, 6\}$

Lösung zu 3.8 b)



Dijkstra Algorithmus

Schritt 3

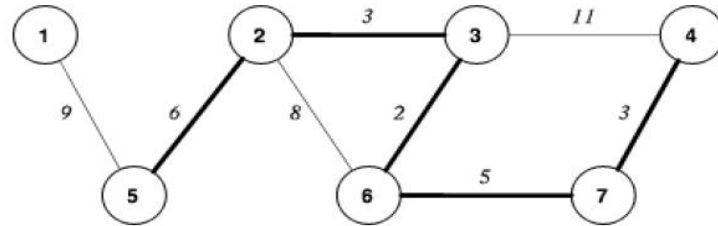
$S = \{1, 2, 3, 5\}$ $D = \{48, 13, 7, 3, 48, 5\}$ $R = \{0, 6, 6, 4, 0, 6\}$

1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 3 aus S → $S = \{1, 2, 5\}$
 2. Prüfe jeden Knoten v , für den es $(3, v)$ gibt, die Distanz über Knoten 3:
→ Knoten 2
 - $C = D[3] + G(3, 2) = 7 + 3 = 10$
 - $C = 10 < D[2] = 13$
→ $R[2] = R[3] = 6$ und $D[2] = C = 10$
- $D = \{48, 10, 7, 3, 48, 5\}$ $R = \{0, 6, 6, 4, 0, 6\}$

Lösung zu 3.8 b)

Dijkstra Algorithmus Schritt 4

$S = \{1, 2, 5\}$ $D = \{48, 10, 7, 3, 48, 5\}$ $R = \{0, 6, 6, 4, 0, 6\}$

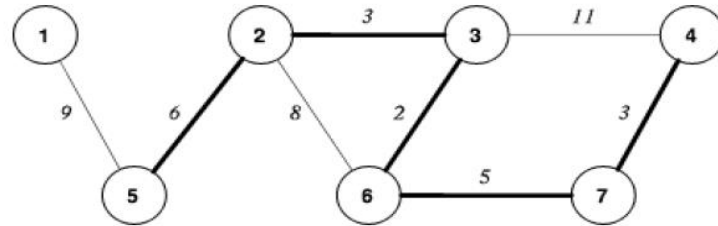


1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 2 aus S → $S = \{1, 5\}$
 2. Prüfe jeden Knoten v , für den es $(2, v)$ gibt, die Distanz über Knoten 2: → Knoten 5
 - $C = D[2] + G(2, 5) = 10 + 6 = 16$
 - $C = 16 < D[5] = 48$
→ $R[5] = R[2] = 6$ und $D[5] = C = 16$
- $D = \{48, 10, 7, 3, 16, 5\}$ $R = \{0, 6, 6, 4, 6, 6\}$

Lösung zu 3.8 b)

Dijkstra Algorithmus Schritt 5

$S = \{1, 5\}$ $D = \{48, 10, 7, 3, 16, 5\}$ $R = \{0, 6, 6, 4, 6, 6\}$

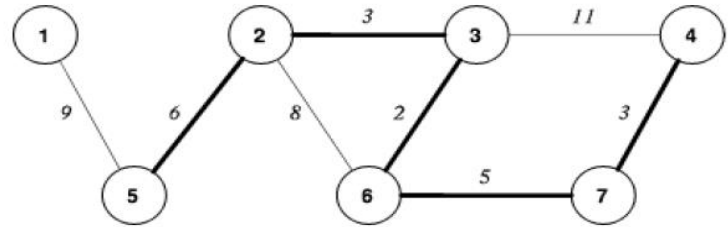


1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 5 aus $S \rightarrow S = \{1\}$
 2. Prüfe jeden Knoten v , für den es $(5, v)$ gibt, die Distanz über Knoten 5: → Knoten 1
 - $C = D[5] + G(5, 1) = 16 + 9 = 25$
 - $C = 25 < D[1] = 48$
→ $R[1] = R[5] = 6$ und $D[1] = C = 25$
- $D = \{25, 10, 7, 3, 16, 5\}$ $R = \{6, 6, 6, 4, 6, 6\}$

Lösung zu 3.8 b)

Dijkstra Algorithmus Schritt 6

$S = \{1\}$ $D = \{25, 10, 7, 3, 16, 5\}$ $R = \{6, 6, 6, 4, 6, 6\}$



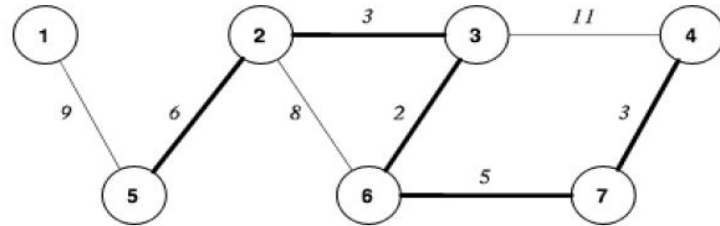
1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ Entferne Knoten 1 aus S → $S = \{ \}$
2. Prüfe jeden Knoten v , für den es $(1, v)$ gibt, die Distanz über Knoten 1: → keine Knoten

Lösung zu 3.8 b)

Dijkstra Algorithmus Schritt 7

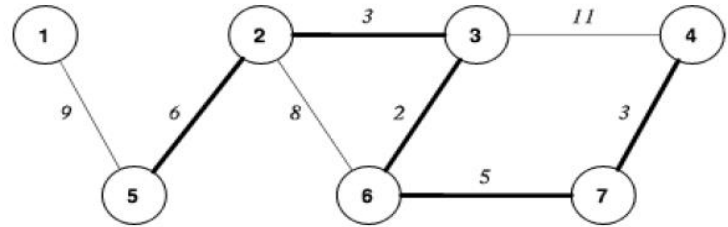
$S = \{0\}$ $D = \{25, 10, 7, 3, 16, 5\}$ $R = \{6, 6, 6, 4, 6, 6\}$

1. Wähle Knoten u aus S , der das kleinste Gewicht in D hat
→ $S = \{\}$ → Abbruchbedingung



Lösung zu 3.8 b)

Dijkstra Algorithmus Zusammenfassung



Quelle = Knoten 7

Schritt 1: $S = \{1,2,3,4,5,6\}$	$D = \{48,48,48,3,48,5\}$	$R = \{0, 0, 0,4, 0,6\}$
Schritt 2: $S = \{1,2,3,5,6\}$	$D = \{48,48,14,3,48,5\}$	$R = \{0, 0, 4,4, 0,6\}$
Schritt 3: $S = \{1,2,3,5\}$	$D = \{48,13,7,3,48,5\}$	$R = \{0,6,6,4, 0,6\}$
Schritt 4: $S = \{1,2,5\}$	$D = \{48,10,7,3,48,5\}$	$R = \{0,6,6,4, 0,6\}$
Schritt 5: $S = \{1,5\}$	$D = \{48,10,7,3,16,5\}$	$R = \{0,6,6,4,6,6\}$
Schritt 6: $S = \{1\}$	$D = \{25,10,7,3,16,5\}$	$R = \{6,6,6,4,6,6\}$
Schritt 7: $S = \{\}$	$D = \{25,10,7,3,16,5\}$	$R = \{6,6,6,4,6,6\}$

Aufgabe 3.8

- a) Erklären Sie das Routing in einem WAN
- b) Berechnen Sie mit Dijkstra's Algorithmus die kürzesten Pfade im Gegebenen Graphen (Quelle: Knoten 7)
 - ❖ Geben Sie für jeden Schritt die Datenstrukturen
 - S (Knoten)
 - D (Kantengewichte von der Quelle aus)
 - R (Next Hop von der Quelle aus)
- c) Nennen Sie charakteristische Eigenschaften des Distance Vector Routings und des Link State Routings

Lösung zu 3.8 c)

- Distance Vector Routing (Algorithmus nach Bellmann-Ford)
 - ❖ ("Teile deinen Nachbarn mit, wie die Welt aussieht")
Der Distance-Vector-Algorithmus arbeitet "extrovertiert"

- Jeder Router führt eine Routing-Tabelle, in die er Kosten aller ihm bekannten Verbindungen einträgt, d.h. lokale Information besteht aus
 - ❖ Next-Hop Routingtabelle
 - ❖ Entfernung zu jedem Switch

Lösung zu 3.8 c)

- ❑ Distance Vector Routing (Algorithmus nach Bellmann-Ford)
- ❑ Die Routing-Tabelle wird regelmäßig zwischen benachbarten Routern vollständig ausgetauscht,
 - ❖ d.h. Router senden periodisch Topologieinformationen und andere Router erneuern ihre Routingtabellen anhand der empfangenen Informationen

Lösung zu 3.8 c)

- ❑ Distance Vector Routing (Algorithmus nach Bellmann-Ford)

- ❑ **Anwendung:**
 - ❖ In RIP (Routing Information Protocol) genutzt
 - gilt als veraltet - wird für Routing innerhalb autonomer Systeme genutzt
 - ❖ In BGP (Border Gateway Protocol) genutzt
 - heute mit Path-Vector Algorithmus - de facto Standard für Routing zwischen autonomen Systemen (z.B. ISP untereinander)

Lösung zu 3.8 c)

□ Vorteile:

- ❖ einfach zu implementieren
- ❖ weitestgehend wartungsfrei
- ❖ selbstorganisierend

Lösung zu 3.8 c)

❑ Nachteile:

- ❖ bei größeren Netzen in relativ kurzer Zeit sehr große Übertragungsmengen
- ❖ Kann Konvergenzprobleme haben (Count-To-Infinity-Effekt)
- ❖ Inkonsistenzen in Routing-Tabellen (wenn bei teilweisen Netzwerkstörungen die Routing-Tabellen einzelner Router durch andere Router überschrieben werden, die die Störungen noch nicht erkannt haben)

Durch Erweiterungen (Reverse Poison, Split Horizon, Hold-down) werden Nachteile teilweise gelindert

Lösung zu 3.8 c)

- ❑ **Fazit:** Distance-Vector-Algorithmus trotz seiner anfänglichen Einfachheit, vor allem in großen, weit verzweigten und relativ störungsanfälligen Netzwerken mit Vorsicht zu genießen

Lösung zu 3.8 c)

- Link State Routing (Algorithmus nach Dijkstra)
- Jeder Router führt eine eigene Routing-Tabelle, in der die gesamte Topologie des Netzwerks abgebildet ist

Lösung zu 3.8 c)

- Änderungen von Verbindungen werden per Broadcast über sogenannte Link State Announcement allen Routern im Netzwerk mitgeteilt
-> globale Topologie kann ständig erneuert werden

- **Anwendung:**
 - ❖ In OSPF (Open Shortest Path First) genutzt
 - ❖ heute de facto Standard für das Routing innerhalb autonomer Systeme

Lösung zu 3.8 c)

□ Vorteile:

- ❖ Router führen unabhängig Berechnungen durch
-> Netzwerktopologie wird von Routenberechnung getrennt
- ❖ konvergiert schnell und gut skalierbar (besser als RIP)

□ Nachteile:

- ❖ Sehr komplex

□ Fazit: Routing-Tabellenänderungen werden beim Link State Algorithmus moderat und zielgerichtet weitergegeben

- > eignet sich für sehr große Netzwerke und Routing-Tabellen