

Aufgabe 4.1

Die Vermittlungsschicht hat die Aufgabe, *Pakete* von der Quelle bis zum Ziel – eventuell über mehrere Teilstrecken – zu übertragen.

Aufgabe 4.2

Ein Host sendet ein Paket an einen Router. Der speichert das Paket und kontrolliert die Prüfsumme. Danach wird das Paket auf einer geeigneten Ausgangsleitung zum nächsten Router weitergeleitet, bis es den Zielhost erreicht hat.

Aufgabe 4.3

- *verbindungslos*: Jedes Paket wird unabhängig von den anderen befördert.
- *verbindungsorientiert*: Der Quellhost baut eine *virtuelle Verbindung* zum Zielhost auf. Dann werden alle Pakete über diese Verbindung befördert. Schliesslich wird die virtuelle Verbindung wieder abgebaut.

Aufgabe 4.4

- Den Prozess, der die ankommenden Pakete verwaltet und weiterleitet.
- Den Prozess, der entscheidet, auf welchem (optimalen) Pfad die Pakete weitergeleitet werden sollen.

Aufgabe 4.5

Algorithmus von Dijkstra:

- Wähle A als ersten Arbeitsknoten.
 - $B = (A, 2)$ [Erreiche B von A aus mit 2 Einheiten.]
 - $G = (A, 6)$

Markiere A als erledigt.

- Wähle B als nächsten Arbeitsknoten, da er unter den noch nicht erledigten Knoten (B, G) die kürzeste Distanz 2 hat.
 - $C = (B, 9)$
 - $E = (B, 4)$

Markiere B als erledigt.

- Wähle E als nächsten Arbeitsknoten, da er unter den noch nicht erledigten Knoten (C, E, G) die kürzeste Distanz 4 hat.

- $F = (E, 6)$
- $G = (E, 5)$ [$G = (A, 6)$ kann relaxiert werden.]

Markiere E als erledigt.

- Wähle G als nächsten Arbeitsknoten, da er unter den noch nicht erledigten Knoten (C, G, F) die kürzeste Distanz 5 hat.

- $H = (G, 9)$

Markiere G als erledigt.

- Wähle F als nächsten Arbeitsknoten, da er unter den noch nicht erledigten Knoten (C, F, H) die kürzeste Distanz 6 hat.

- $H = (F, 8)$ [$H = (G, 9)$ kann relaxiert werden.]

- Der Pfad von B nach C muss nicht relaxiert werden, da er gleich lang ist, wie der von F nach C .

Markiere F als erledigt.

- Wähle H als nächsten Arbeitsknoten, da er unter den noch nicht erledigten Knoten (C, H) die kürzeste Distanz 8 hat.

- $D = (H, 10)$

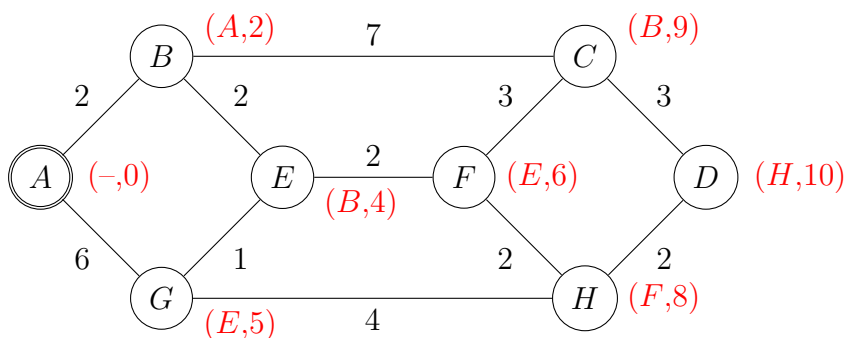
Markiere H als erledigt.

- Wähle C als nächsten Arbeitsknoten, da er unter den noch nicht erledigten Knoten (C, D) die kürzeste Distanz 9 hat.

- Der letzte Hop von C nach D lässt sich nicht relaxieren.

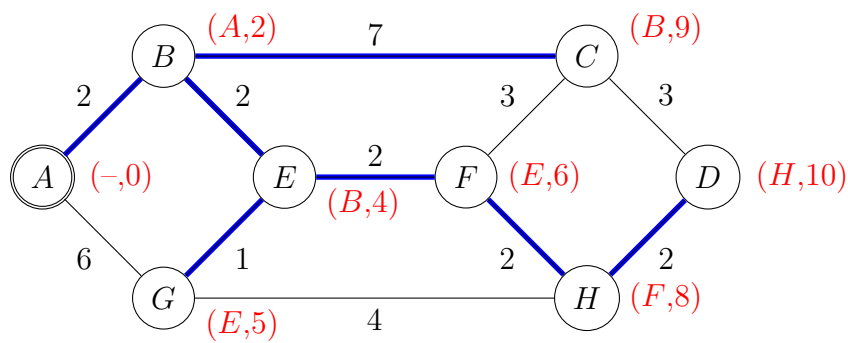
Markiere C (und damit auch D) als erledigt.

Schlussresultat:



Aufgabe 4.6

Minimal Spanning Tree:



Aufgabe 4.7

- *Funktionsweise*: Das ankommende Paket wird auf allen Leitungen gesendet, ausser derjenigen, auf der es angekommen ist.
- *Vorteile*: Einfach, schnell und robust
- *Nachteil*: Das Netz wird mit einer grossen Zahl von Paketen (Duplikaten) überschwemmt.

Aufgabe 4.8

- *Flusskontrolle*: Ein zu schneller Sender soll daran gehindert werden, einen zu langsamen Empfänger mit Daten zu überschwemmen.
- *Überlastung*: Wenn die Anzahl der ins Netz eingespeisten Pakete grösser ist, als die Übertragungskapazität dieses Netzes.

Flusskontrolle findet zwischen zwei Hosts statt, während Überlastung ein ganzes (Teil-)Netz betrifft.

Aufgabe 4.9

- *Warnbit*: Ein Router schreibt ein sogenanntes Warnbit in die weiterzuleitenden Pakete. Dieses Warnbit wird vom Empfänger in die Empfangsbestätigung an den Sender kopiert, so dass der Sender auf diesem Weg erfährt, dass das Netz überlastet ist.
- *Choke-Pakete*: Der Router verschickt direkt an den Sender ein Choke-Paket, das ihn auffordert, die Datenmenge zu drosseln.
- *Lastabwurf*: Der Router verwirft Pakete

Aufgabe 4.10

- Ein hoher Jitter ist schlecht bei der Übertragung von Audio- und Videodaten *in Echtzeit*.
- Bei E-Mail und anderen asynchronen Anwendungen fällt ein hoher Jitter praktisch nicht ins Gewicht.

Aufgabe 4.11

- Beim Leaky-Bucket-Algorithmus werden die in unregelmässigen Abständen anfallenden Daten vor dem Versand gepuffert (in einer Warteschlange gespeichert) und von dort mit einer gleichmässigen Datenrate ins Netzwerk eingespeist.
- Der Leaky-Bucket-Algorithmus dient dazu, einen ungleichmässigen Datenstrom in ein Netzwerk zu glätten.

Aufgabe 4.12

ID=13	FO=0	MF=1				
-------	------	------	--	--	--	--

ID=13	FO=6	MF=0				
-------	------	------	--	--	--	--

Aufgabe 4.13

dec	bin
17	00010001
5	00000101
3	00000011
135	10000111

- (a) 17.5.3.135 00010001.00000101.00000011.10000111
(b) 255.255.255.128 11111111.11111111.11111111.10000000
(c) 17.5.3.128 00010001.00000101.00000011.10000000
(d) 17.5.3.255 00010001.00000101.00000011.11111111
(e) $2^{(32-25)} - 2 = 2^7 - 2 = 128 - 2 = 126$ Hosts (*)

(*) Die Adressen für Netzwerk und Broadcast dürfen nicht vergeben werden