

# Algorithmus von Dijkstra

**Voraussetzung:** globale Vorinformation (wird erreicht über “*link state* broadcast”)

**Aufgabe:** Finde kürzesten Weg von u nach z

**Lösung:** Findet alle kürzesten Wege von u zu anderen Knoten, die nicht länger als zu z sind

**Dynamisierbarkeit:** kann geänderte Werte für Weiterreise richtig nutzen

## 1 **Initialization:**

2 Done = {u}

3 for all nodes v do:

4 if v adjacent to u

5 then  $C_u(v) = c(u,v)$

6 else  $C_u(v) = \infty$

7

## 8 **Loop**

9 find w not in Done such that  $C_u(w)$  is a minimum

10 add w to Done

11 update  $C_u(v)$  for all v adjacent to w and not in Done:

12  $C_u(v) = \min( C_u(v), C_u(w) + c(w,v) )$

13 **until all nodes in Done (resp. z in Done)**

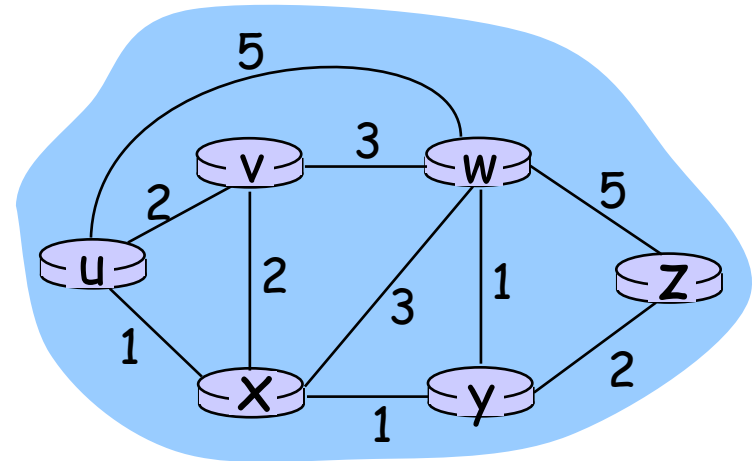
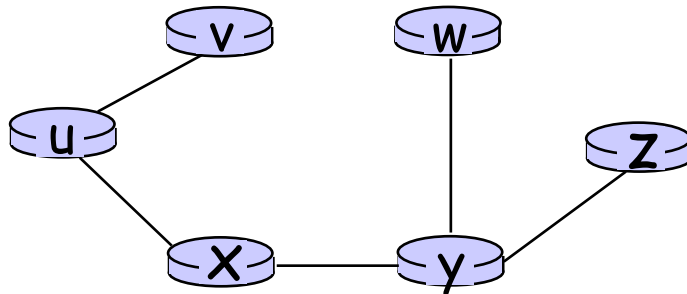
## **Notation:**

- **$c(x,y)$ :** Kantenkosten von x nach y ( $\infty$  wenn nicht benachbart)
- **$C_u(v)$ :** Berechnete Kosten von u bis v
- **$p_u(v)$ :** Vorgänger auf dem berechneten Weg von u nach v
- **Done:** Menge der Knoten, zu denen die Kosten endgültig feststehen

# Algorithmus von Dijkstra

**Beispiel: Weg von u nach z**

**Gerüst der kürzesten Wege von u:**

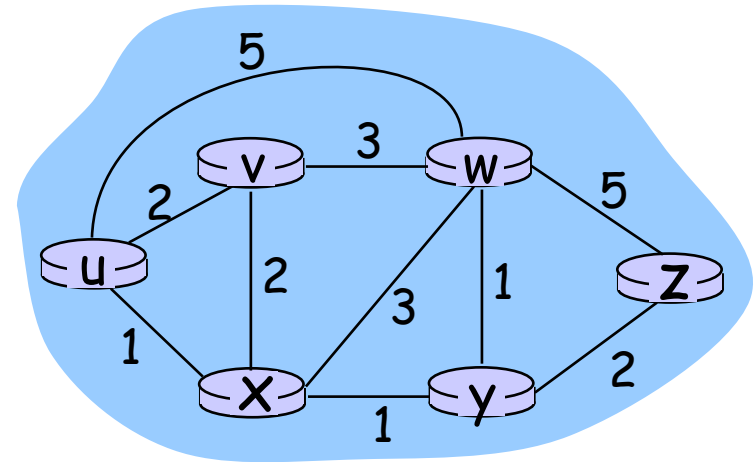
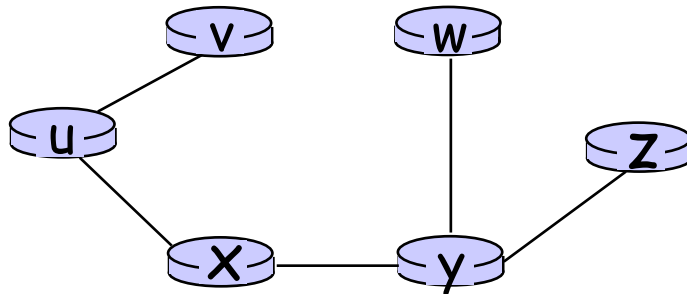


Step	Done	$C_u(v), p(v)$	$C_u(w), p(w)$	$C_u(x), p(x)$	$C_u(y), p(y)$	$C_u(z), p(z)$
0	u	2,u	5,u	1,u	$\infty$	$\infty$
1	ux	2,u	4,x		2,x	$\infty$
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

# Algorithmus von Dijkstra

**Beispiel:**

**Gerüst der kürzesten Wege von u:**



**Forwarding table von u:**

destination	link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

**Forwarding table von x:**

destination	link
y	(x,y)
w	(x,y)
z	(x,y)

**Forwarding table von y:**

destination	link
w	(y,w)
z	(y,z)

# Algorithmus von Dijkstra

## Analyse

Algorithmische Komplexität für  $n$  Knoten und  $m$  Kanten:

findet die kürzesten Wege von  $u$  zu jedem anderen Knoten in Zeit  $O(m + n \log n)$

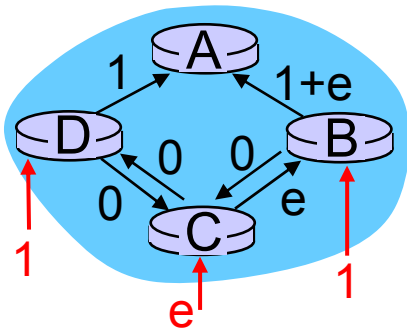
- Es ist kein Algorithmus bekannt, der das Problem für feste  $u$  und  $z$  schneller löst

Bei Anwendung auf jede Startecke  $u$ :

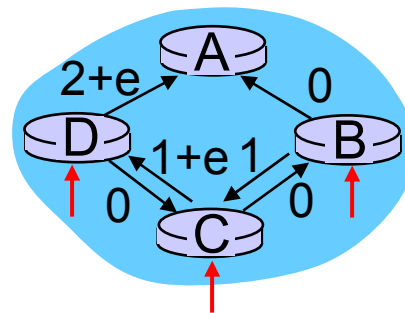
Gesamtzeit  $O(nm + n^2 \log n)$

- Für die Berechnung aller Wege gibt es etwas schnellere Verfahren

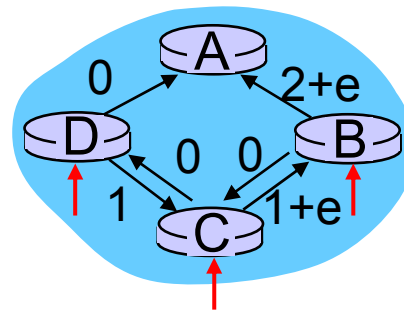
Bei Kostenmaß = Verkehrsaufkommen: **Oszillierung möglich !**



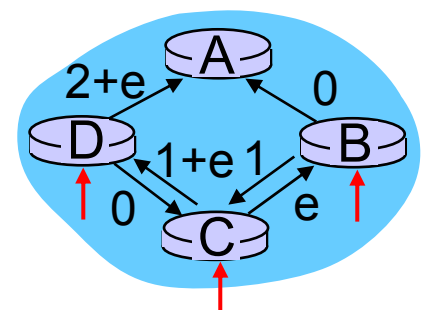
Startsituation:  
Kontinuierliche Versendung  
der angegebenen Paketzahl  
nach A



B und C entdecken  
besseren Weg



B, C, D entdecken  
besseren Weg



B, C, D entdecken  
besseren Weg, ....