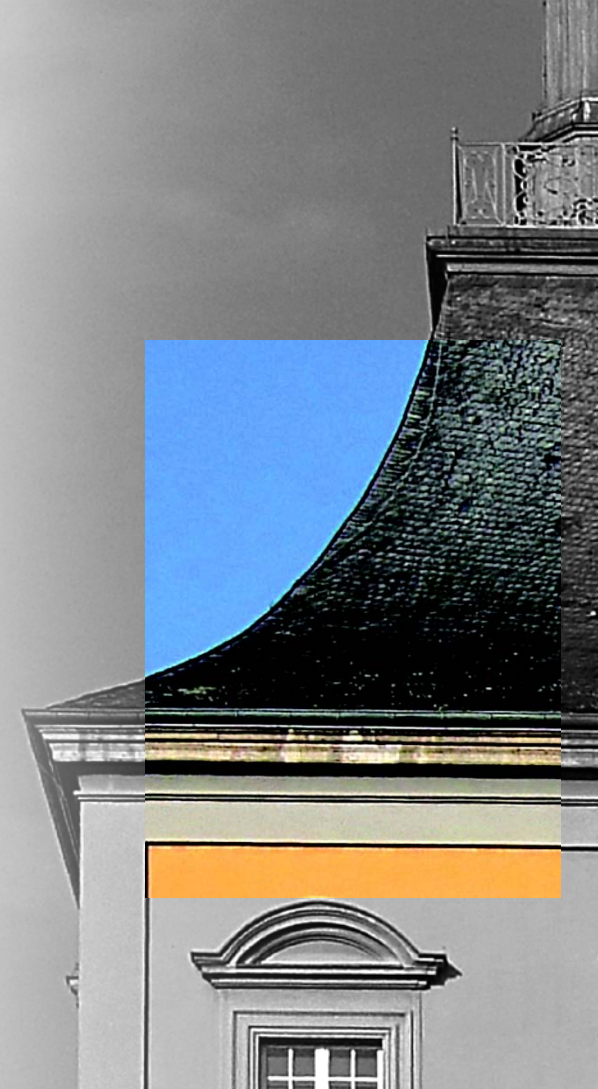


VORLESUNG  
**NETZWERKSICHERHEIT**

**SOMMERSEMESTER 2023**

**MO. 14-16 UHR**



- **Geschichte des Internets**
  - ARPANet
  - NSFNet
  - „Das Internet“
- Internet Protocol
- Wie Alice und Bob über das Internet kommunizieren
- Netzwerk von Netzwerken (Autonome Systeme)
- Netzwerk-Routing Protokolle
- Border Gateway Protocol

# DIE ENTSTEHUNG DES INTERNETS

- „The ARPANet was the first transcontinental, high-speed computer network.“  
-- (Eric S. Raymond, author and software developer)

# DIE ENTSTEHUNG DES INTERNETS – TEIL 1

ARPANet (Advanced Research Projects Agency Network)

- Erste Forschung begann bereits im Jahre 1963.

Schon 1969 führte das ARPANet TCP/IP Netzwerk-Protokolle ein.

# DIE ENTSTEHUNG DES INTERNETS – TEIL 1

## ARPANet (Advanced Research Projects Agency Network)

- Erste Forschung
- Schon 1969 führte d

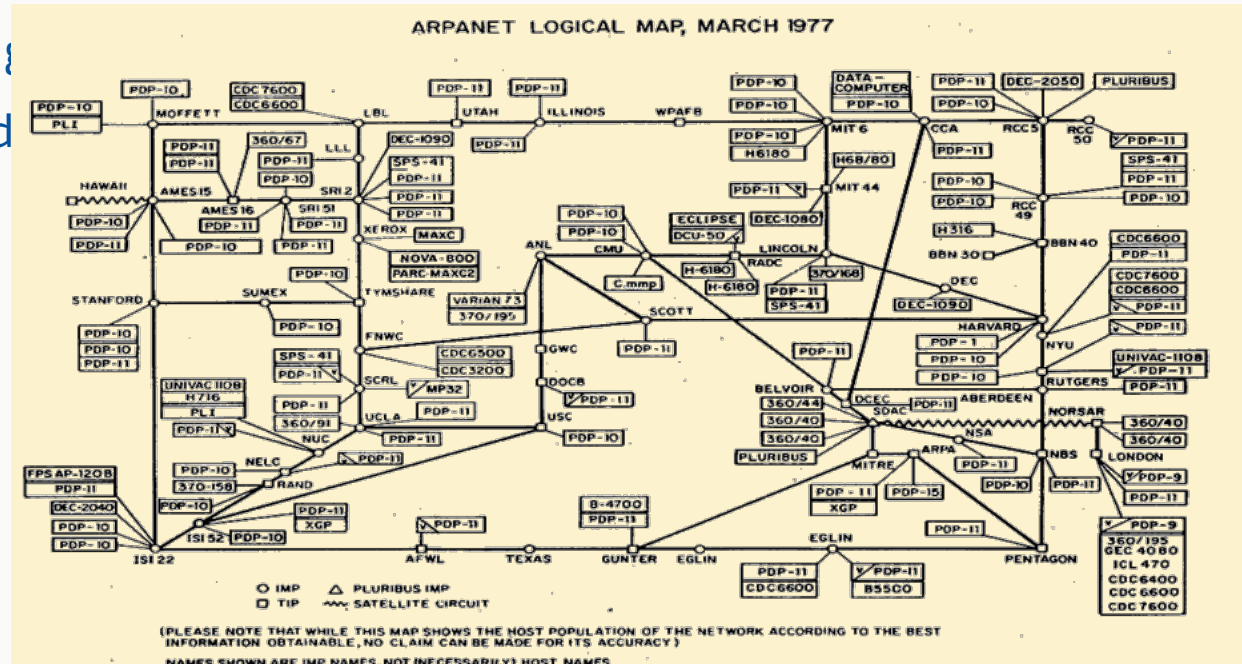


Image ©  
The Computer  
History Museum

# DIE ENTSTEHUNG DES INTERNETS – TEIL 2

## NSFNet (National Science Foundation Network)

- Folgte dem ARPANet 1985 und wurde 1995 vom Internet abgelöst
- Übertragungsgeschwindigkeiten wuchsen von 56 Kbit/s bis auf 45 Mbit/s

### Traditionelle Dienste:

- E-Mail
- News
- Remote login
- File transfer



## DIE ENTSTEHUNG DES INTERNETS – TEIL 3

- Das Internet verbindet heute Millionen von Endgeräten miteinander.
- Endnutzer und Unternehmen bezahlen Internet Service Provider (ISPs), um Zugang zum Internet zu erhalten.
- Service Provider und große Unternehmen (z.B. Google, Microsoft, etc.) betreiben eigene Wide Area Networks („Autonome Systeme“).
- AS sind paarweise miteinander verbunden und bilden mit all diesen Verbindungen den Internet-Backbone.

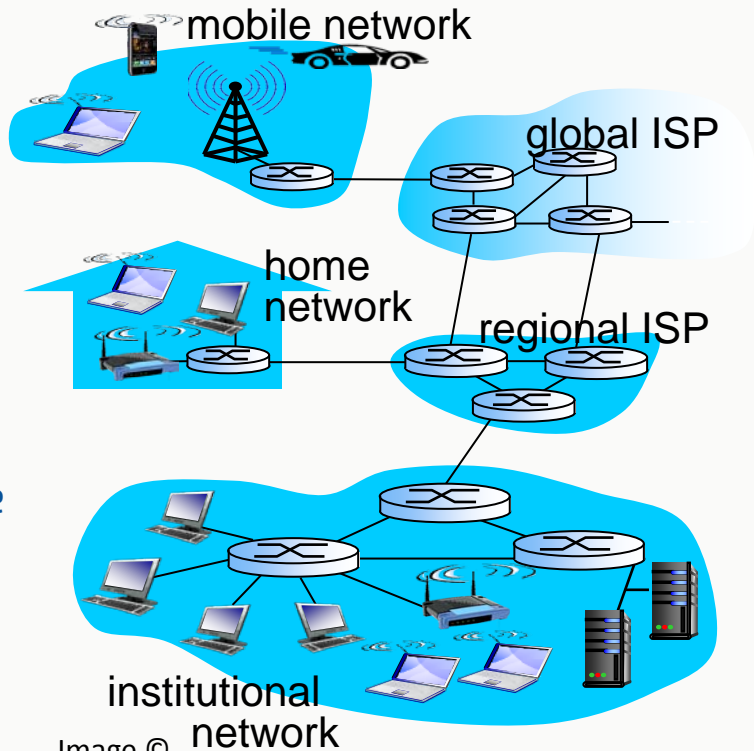
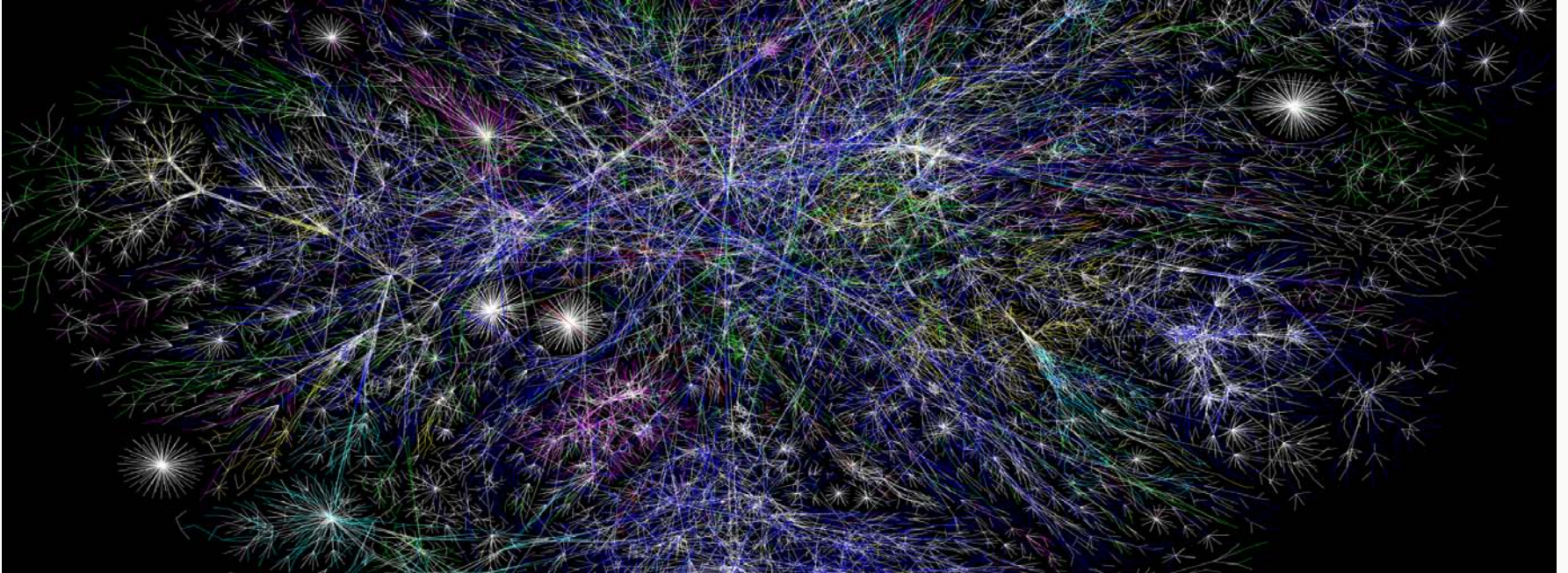


Image ©  
J.F. Kurose & K. W. Ross  
(Book: Computer Networking)



# EINE LANDKARTE DES INTERNETS?



Teil einer Karte des Internets am 15. Januar 2005, gefunden auf [opte.org](http://opte.org)

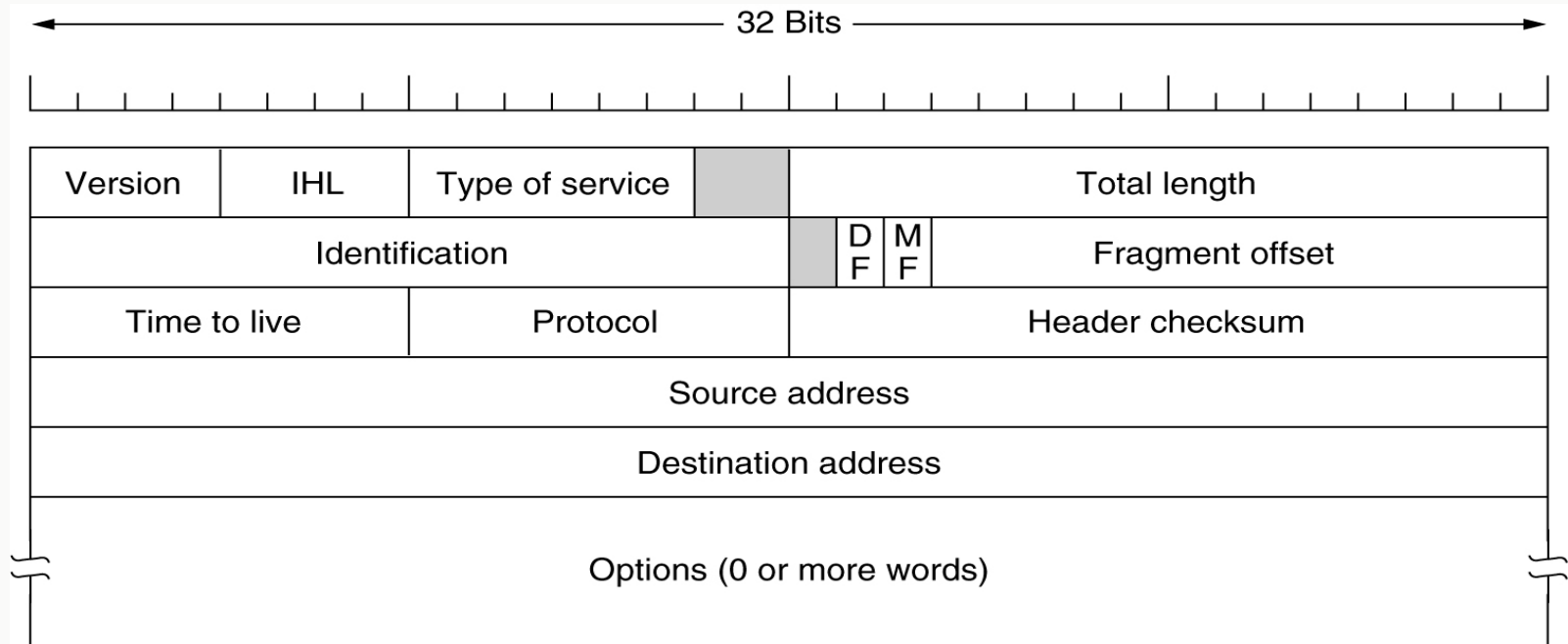


- Geschichte des Internets
  - ARPANet
  - NSFNet
  - „Das Internet“
- **Internet Protocol**
- Wie Alice und Bob über das Internet kommunizieren
- Netzwerk von Netzwerken (Autonome Systeme)
- Netzwerk-Routing Protokolle
- Border Gateway Protocol

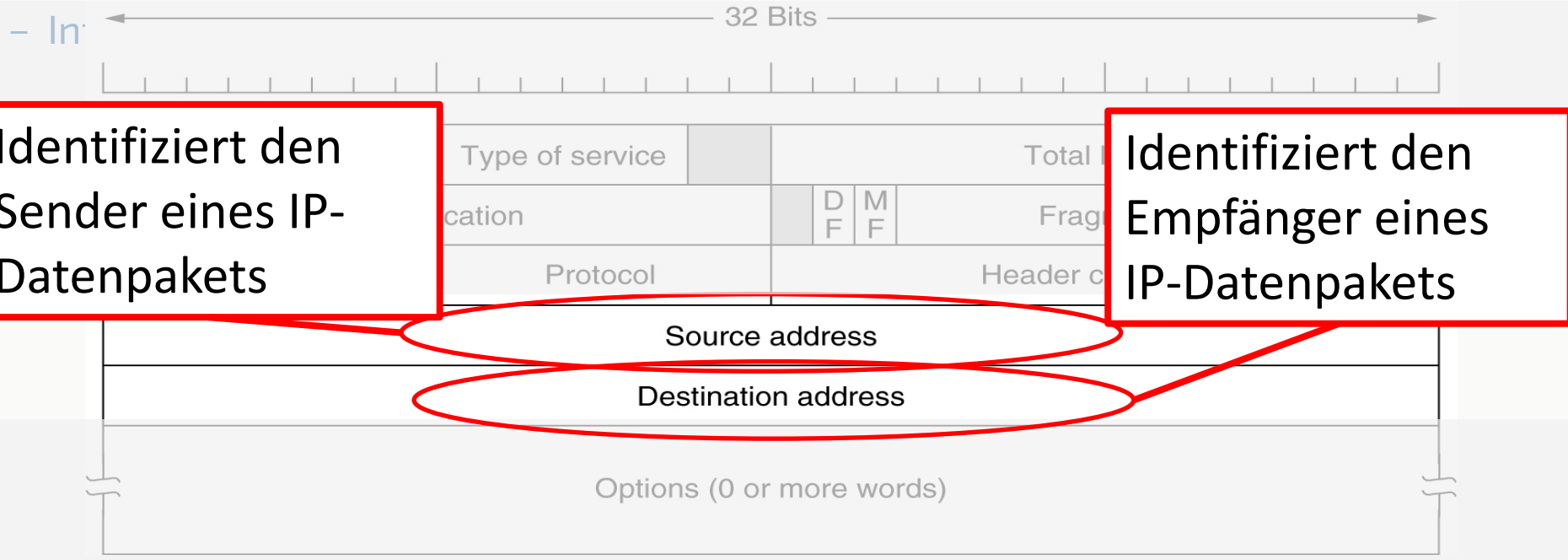
Ein *Protokoll* definiert das *Format* und die *Reihenfolge* von *Nachrichten*, die zwischen zwei oder mehr kommunizierenden Einheiten ausgetauscht werden, sowie die *Handlungen*, die bei der *Übertragung* und/oder beim *Empfang* einer *Nachricht* oder eines anderen Ereignisses unternommen werden.

J.F. Kurose & K. W. Ross  
(Book: Computer Networking)

## Internet Protocol (Layer 3, Network/Internet) Version 4



# DAS INTERNET-PROTOKOLL



- Geschichte des Internets
  - ARPANet
  - NSFNet
  - „Das Internet“
- Internet Protocol
- **Wie Alice und Bob über das Internet kommunizieren**
- Netzwerk von Netzwerken (Autonome Systeme)
- Netzwerk-Routing Protokolle
- Border Gateway Protocol

- Wie Alice und Bob über das Internet kommunizieren

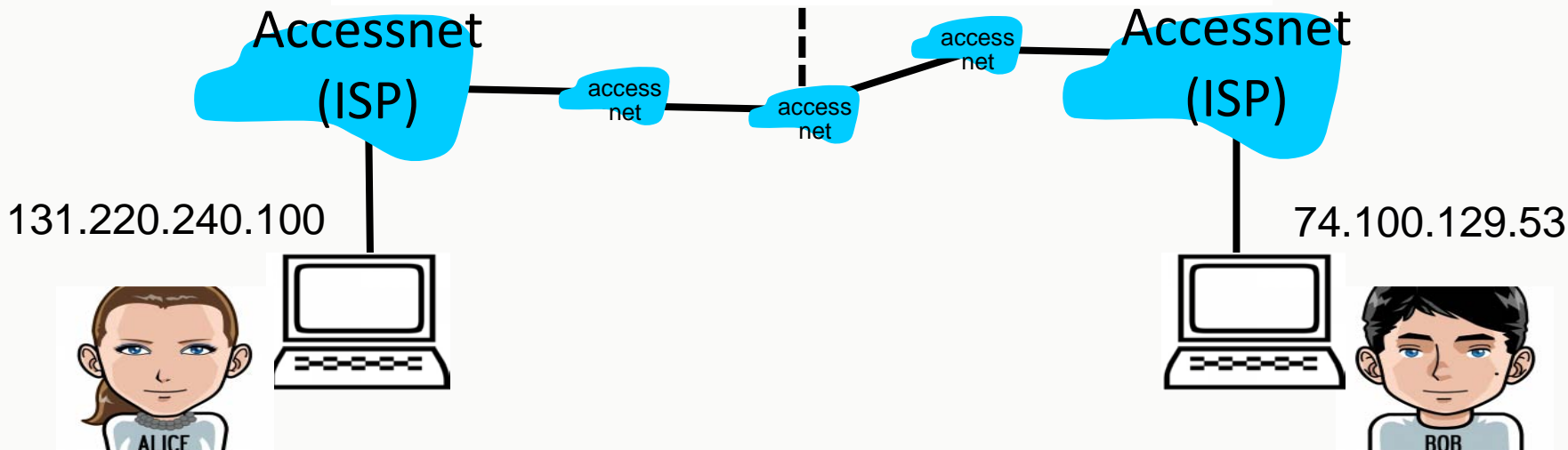
Angenommen, es gibt eine direkte Verbindung zwischen dem ISP von Alice und dem von Bob





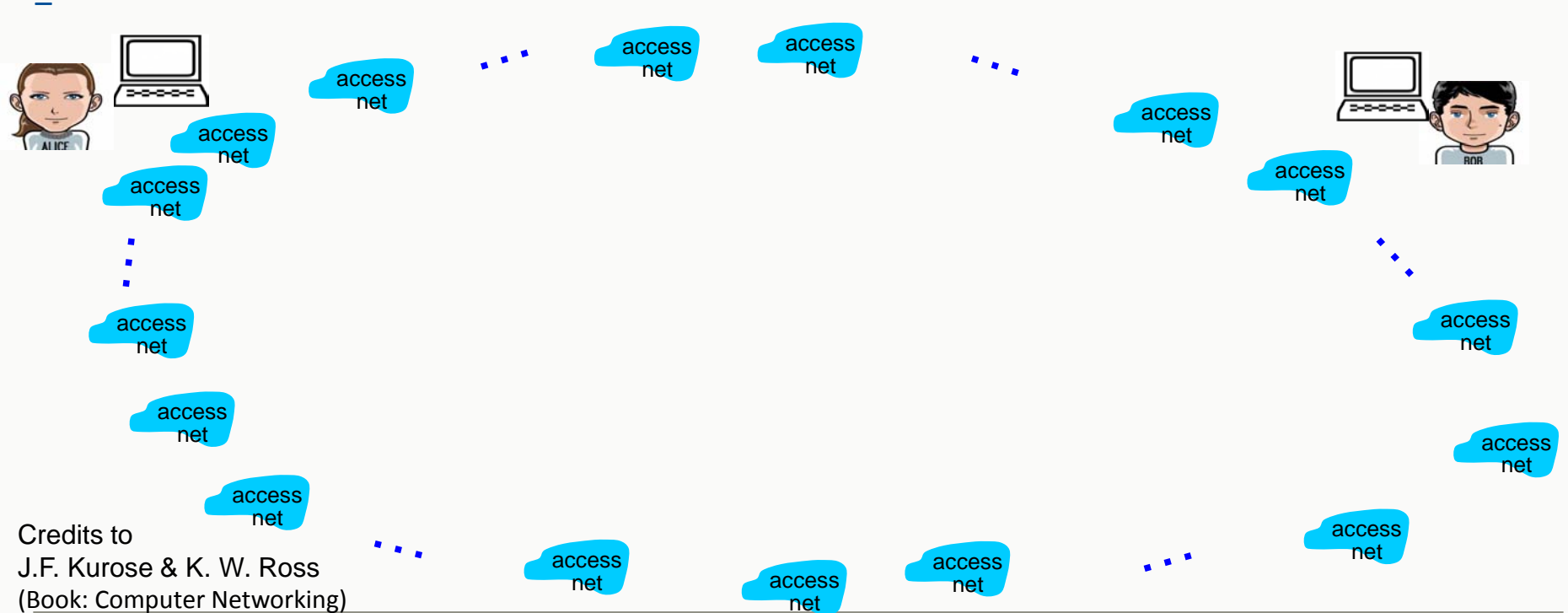
- Wie Alice und Bob über das Internet kommunizieren

Angenommen, es gibt keine direkte Verbindung zwischen dem ISP von Alice und dem von Bob



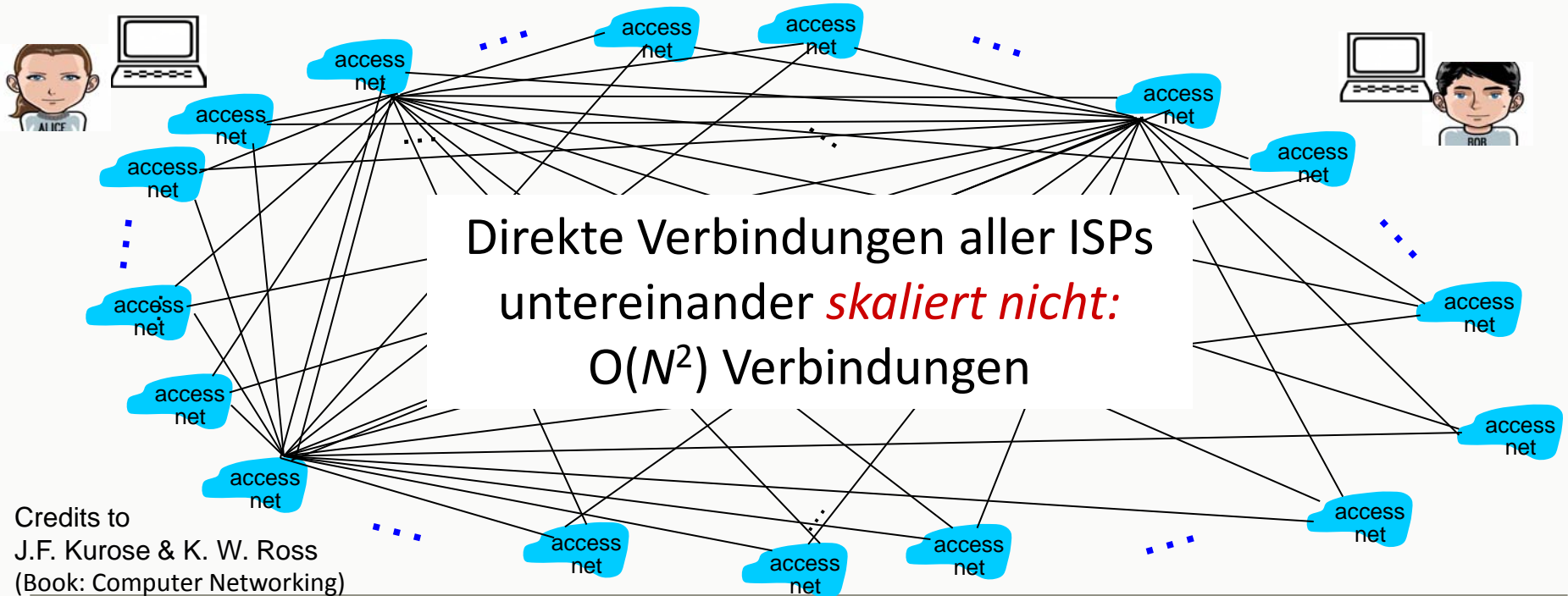
- Geschichte des Internets
  - ARPANet
  - NSFNet
  - „Das Internet“
- Internet Protocol
- Wie Alice und Bob über das Internet kommunizieren
- **Netzwerk von Netzwerken (Autonome Systeme)**
- Netzwerk-Routing Protokolle
- Border Gateway Protocol

# NETZWERK VON NETZWERKEN



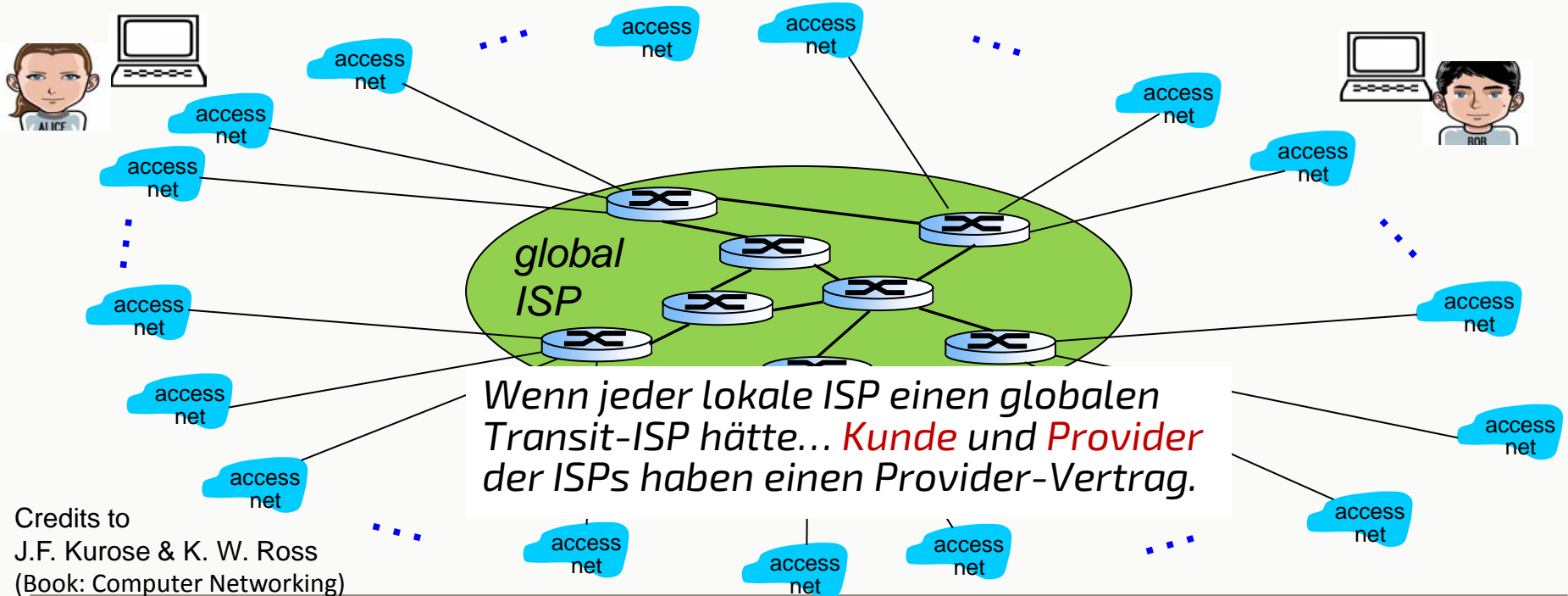
Credits to  
J.F. Kurose & K. W. Ross  
(Book: Computer Networking)

# NETZWERK VON NETZWERKEN – DER NAIVE ANSATZ



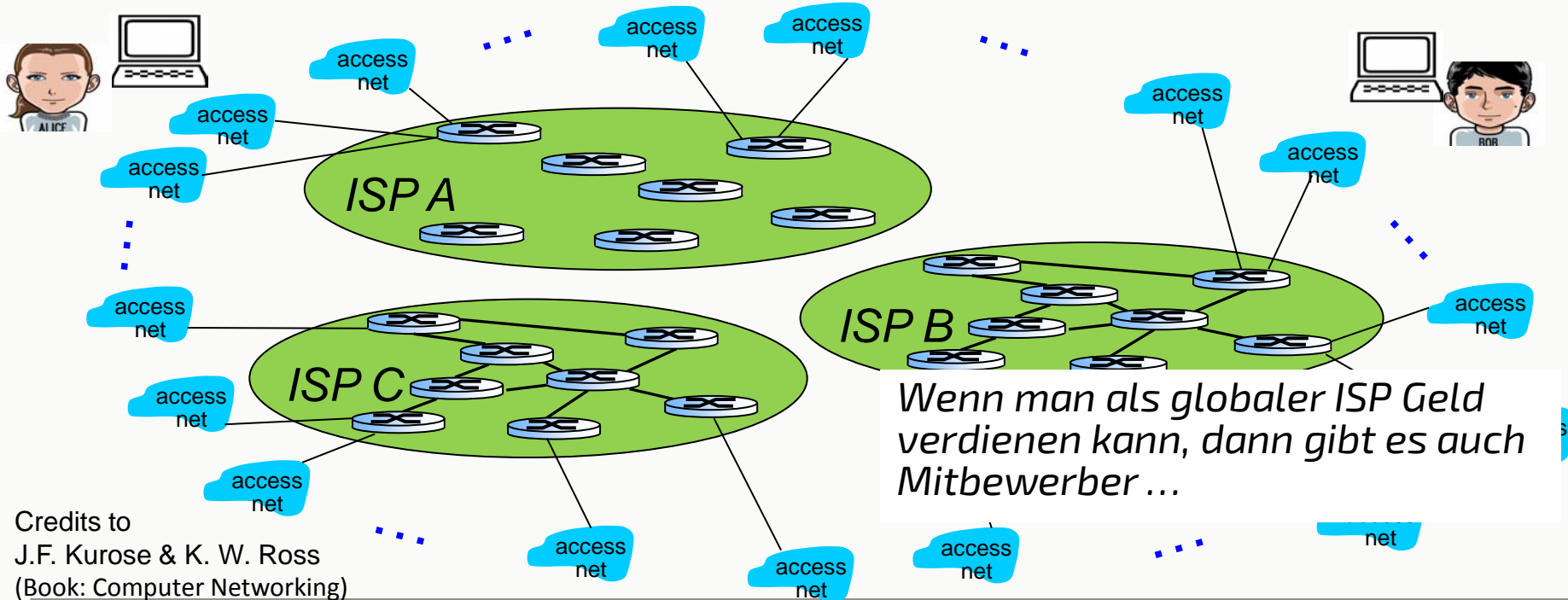
Credits to  
J.F. Kurose & K. W. Ross  
(Book: Computer Networking)

# NETZWERK VON NETZWERKEN – SCHRITT 1



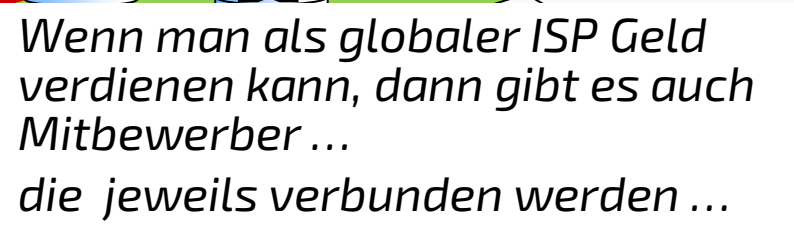
Credits to  
J.F. Kurose & K. W. Ross  
(Book: Computer Networking)

# NETZWERK VON NETZWERKEN – SCHRITT 2

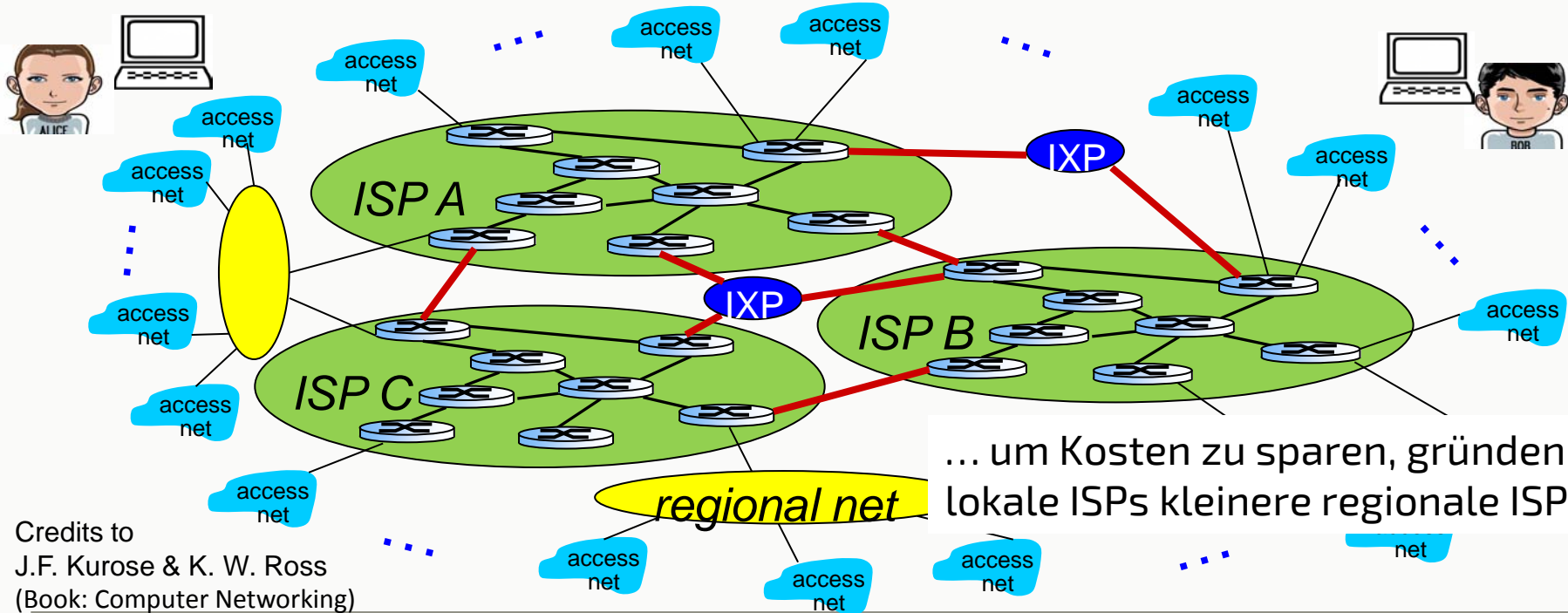


Credits to  
J.F. Kurose & K. W. Ross  
(Book: Computer Networking)





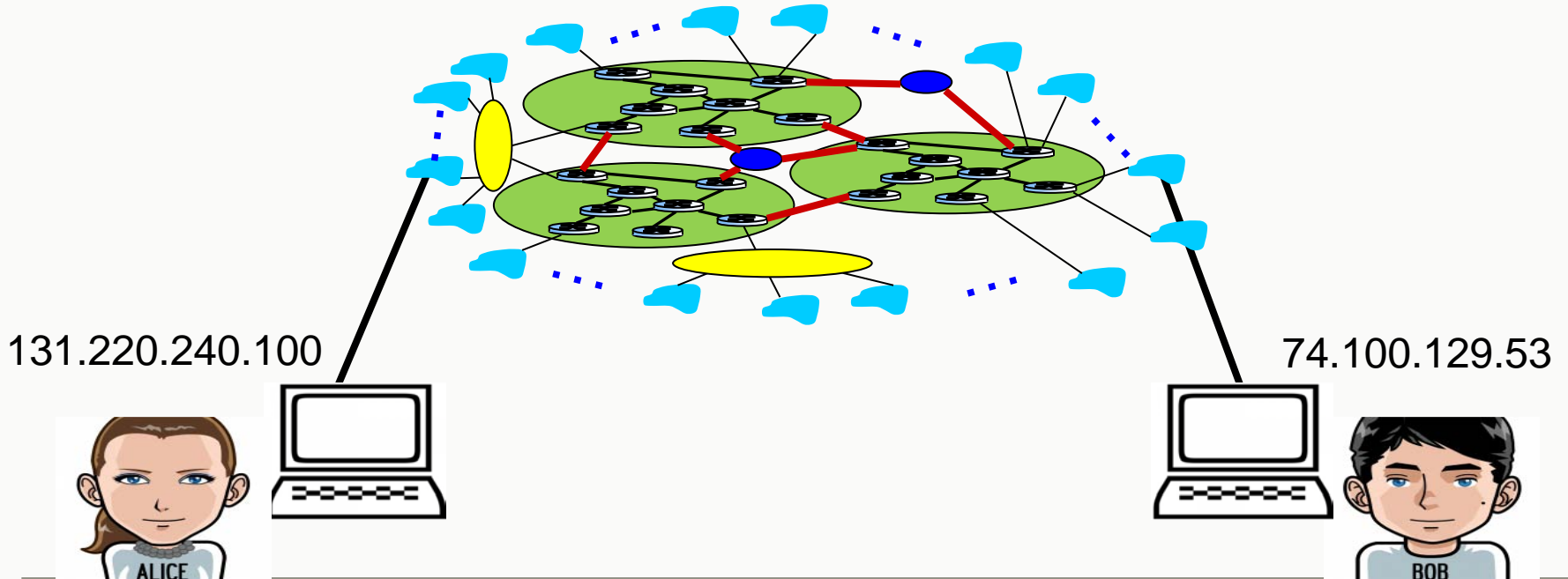
# NETZWERK VON NETZWERKEN – DER LETZTE SCHRITT



Credits to  
J.F. Kurose & K. W. Ross  
(Book: Computer Networking)

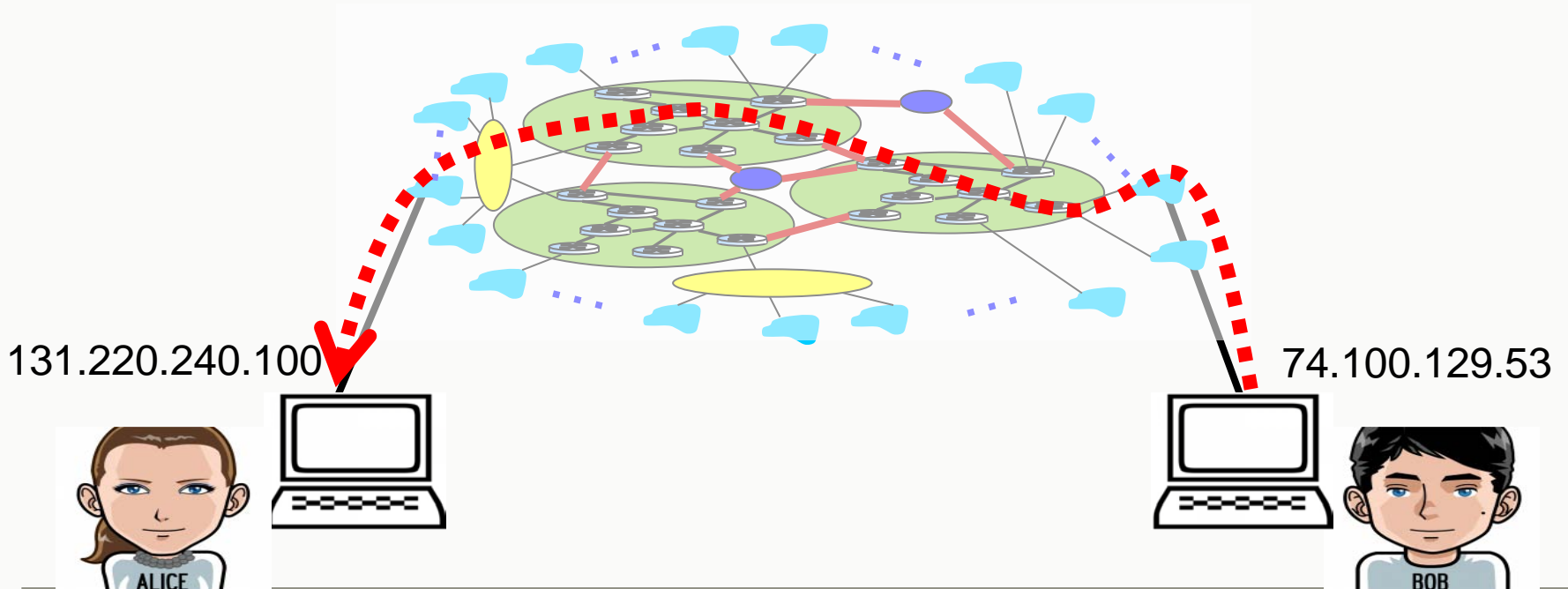
# INTERNET ROUTING

- Also wie kommt jetzt die Nachricht von Bob zu Alice?



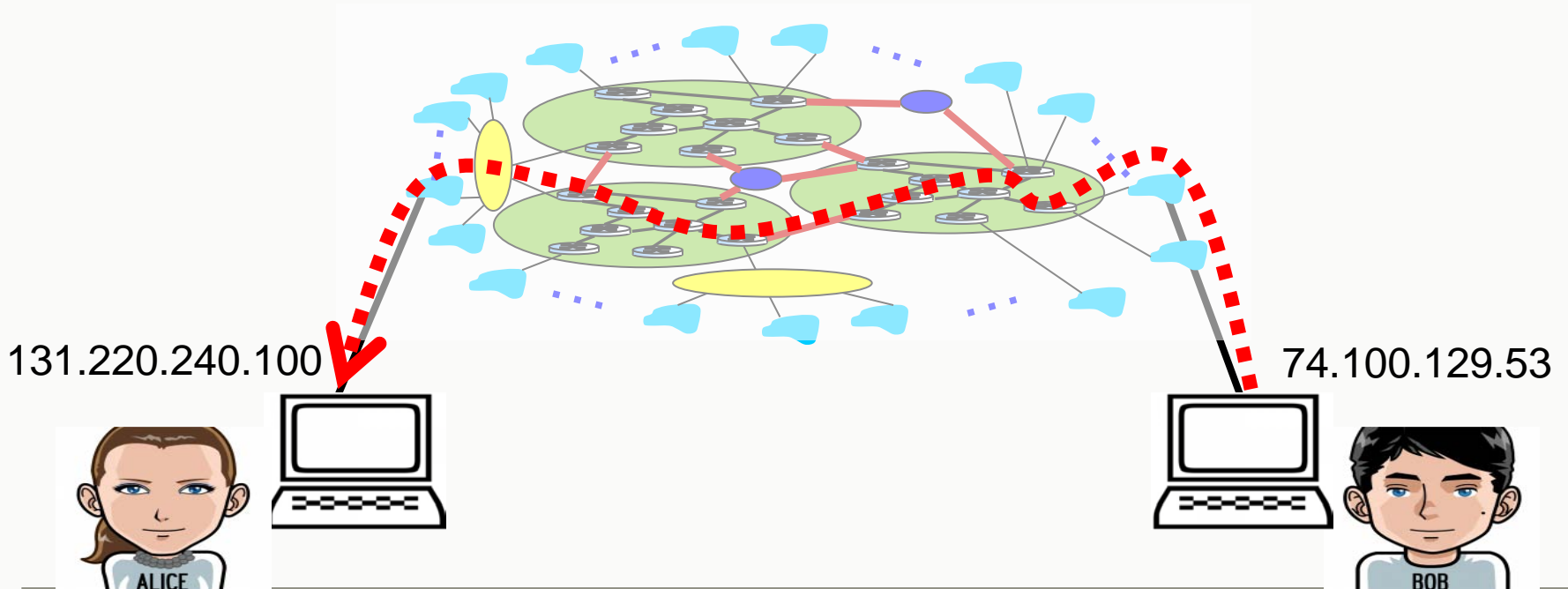
# INTERNET ROUTING

- Also wie kommt jetzt die Nachricht von Bob zu Alice?



# INTERNET ROUTING

- Also wie kommt jetzt die Nachricht von Bob zu Alice?



- Geschichte des Internets
  - ARPANet
  - NSFNet
  - „Das Internet“
- Internet Protocol
- Wie Alice und Bob über das Internet kommunizieren
- Netzwerk von Netzwerken (Autonome Systeme)
- **Netzwerk-Routing Protokolle**
- Border Gateway Protocol



- Routing-Protokolle definieren den Austausch von Netzwerk-Erreichbarkeitsinformationen (Network Layer Reachability Information; NLRI)
- Link-State und Distanz-/Pfad-Vektor-Protokolle basieren auf einer graphbasierten Repräsentation des Netzwerks

## Link-State

- Router verwalten die vollständige Topologie und Link-Kosten des gesamten Netzwerks.
- Jeder Router berechnet die Pfade mit geringsten Kosten zu allen anderen Knoten.
- Dijkstra's algorithm

## Pfad-Vektor

- Jeder Knoten kommuniziert regelmäßig den eigenen Pfad-Vektor an direkt verbundene AS.
- Der Empfänger aktualisiert die eigenen Pfad-Vektor-Einträge entsprechend.
- Bellman-Ford-Gleichung

# LINK-STATE ROUTING PROTOCOL – DIJKSTRA'S ALGORITHM

- Der Algorithmus nach Dijkstra rechnet auf einem Graphen und erstellt eine Weiterleitungstabelle für jeden Knoten
- Notation:
  - $c(x,y)$ : Kosten von Knoten  $x$  zu  $y = \infty$ , wenn nicht direkt verbunden.
  - $D(v)$ : Aktuelle Kosten des Gesamtpfads zu  $v$ .
  - $p(v)$ : Vorgängerknoten auf dem Pfad zu  $v$ .
  - $N'$ : Menge der Knoten, zu denen der günstigste Pfad bekannt ist.

# LINK-STATE ROUTING PROTOCOL – DIJKSTRA'S ALGORITHM

## Algorithmus:

- 1:  $N' = \{u\}$
- 2: for all nodes  $v$
- 3:   if  $v$  adjacent to  $u$
- 4:     then  $D(v) = c(u,v)$
- 5:     else  $D(v) = \infty$
- 6: Loop
- 7:   find  $w$  not in  $N'$  such that  $D(w)$  is minimum
- 8:     add  $w$  to  $N'$
- 9:     update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$ :
- 10:        $D(v) = \min( D(v), D(w) + c(w,v) )$
- 11:       if  $(D(w) + c(w,v)) < D(v)$ :  $p(v) = w$
- 12: until all nodes in  $N'$

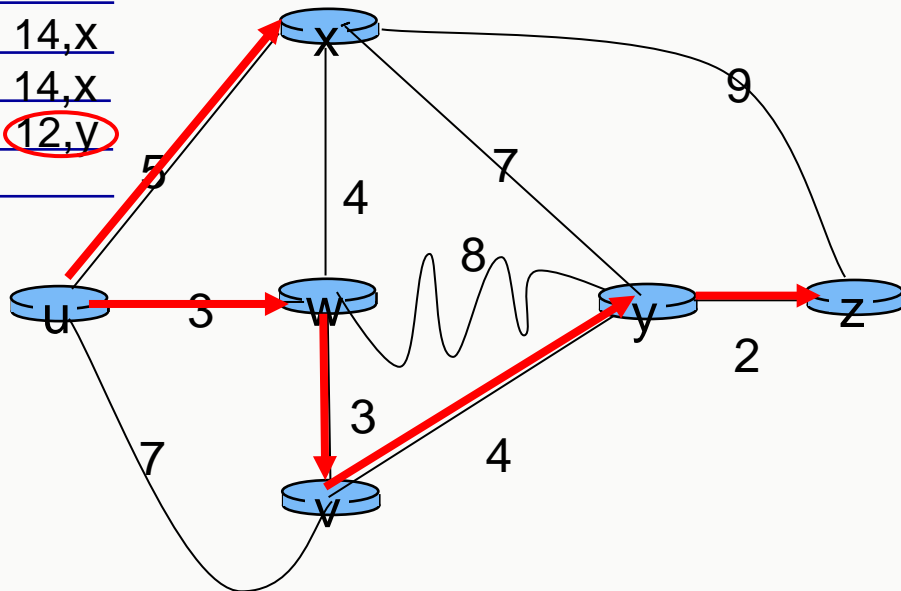
# DIJKSTRA ALGORITHMUS - BEISPIEL

| Step | N'     | D(v)<br>p(v) | D(w)<br>p(w) | D(x)<br>p(x) | D(y)<br>p(y) | D(z)<br>p(z) |
|------|--------|--------------|--------------|--------------|--------------|--------------|
| 0    | u      | 7,u          | 3,u          | 5,u          | $\infty$     | $\infty$     |
| 1    | uw     | 6,w          |              | 5,u          | 11,w         | $\infty$     |
| 2    | uwx    | 6,w          |              |              | 11,w         | 14,x         |
| 3    | uwxv   |              |              |              | 10,v         | 14,x         |
| 4    | uwxvy  |              |              |              | 12,y         |              |
| 5    | uwxvyz |              |              |              |              |              |

## notes:

- ❖ construct shortest path tree by tracing predecessor nodes
- ❖ ties can exist (can be broken arbitrarily)

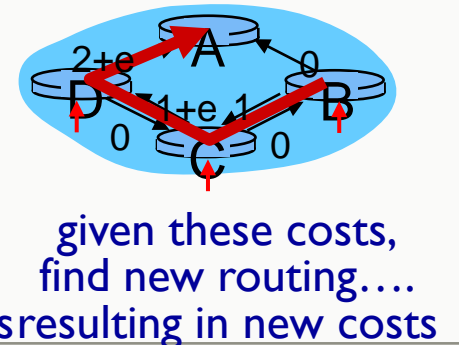
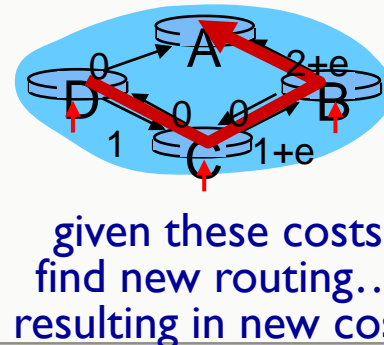
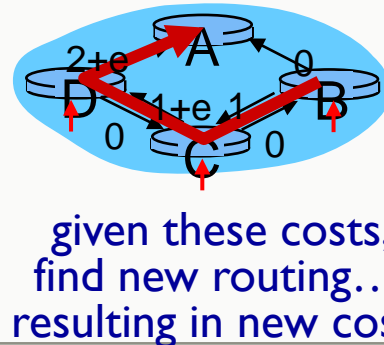
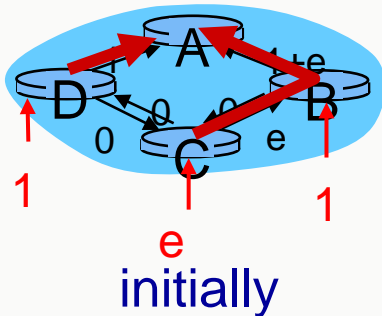
Example from  
J. F. Kurose & K. W. Ross  
(Book: Computer Networking)



# DIJKSTRA ALGORITHM – KOMPLEXITÄT UND PROBLEME

- Komplexität bei  $n$  Knoten:
  - Jeder Durchlauf: Prüfe alle  $w$ , die noch nicht in  $N'$  sind
  - $n(n+1)/2$  Vergleiche:  $O(n^2)$
  - Es gibt effizientere Methoden mit kürzerer Laufzeit:  $O(n \log n)$
- Oszillation möglich:

Example from  
J. F. Kurose & K. W. Ross  
(Book: Computer Networking)

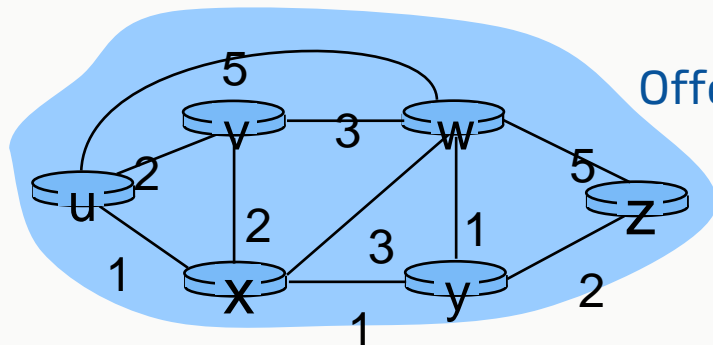


# DISTANZ-VEKTOR ROUTING PROTOCOL – BELLMAN FORD GLEICHUNG

- Bereits im ARPANet als „RIP“ genutzt
- Idee: Dynamische Programmierung
- Ansatz:
  - let
  - $dx(y) := \text{cost of least-cost path from } x \text{ to } y$
  - then
  - $dx(y) := \min_v \{ c(x,v) + dv(y) \}$
- Notation:
  - $\min_v$ : taken over all neighbors  $v$  of  $x$ .
  - $c(x,v)$ : cost to neighbor  $v$ .
  - $dv(y)$ : cost from neighbor  $v$  to node  $y$ .



# BELLMAN-FORD-GLEICHUNG - BEISPIEL



Offensichtlich:  $dv(z) = 5$ ,  $dx(z) = 3$ ,  $dw(z) = 3$

B-F-Gleichung:

$$\begin{aligned} du(z) &= \min \{ c(u,v) + dv(z), \\ &\quad c(u,x) + dx(z), \\ &\quad c(u,w) + dw(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

Der Nachbarknoten mit dem kleinsten Kostenwert wird der Next-Hop des kürzesten Pfads in der "Forwarding"-Tabelle

# DISTANZ-VEKTOR ROUTING PROTOCOL – BELLMAN-FORD-GLEICHUNG

$D_x(y)$  = geschätzte Kosten von  $x$  nach  $y$

- $x$  verwaltet Distanz-Vektor  $D_x = [D_x(y): y \in N]$

Knoten  $x$

- kennt die Kosten zu jedem Nachbarn  $v$ :  $c(x, v)$
- verwaltet für jeden Nachbarn  $v$  einen Distanz-Vektor  $D_v = [D_v(y): y \in N]$

Idee:

- Jeder Knoten sendet seinen  $D_x$  regelmäßig an seine Nachbarn.
- Wenn  $x$  von seinem Nachbarn einen neuen Distanz-Vektor erhält, aktualisiert er seinen:  
 $D_x(y) \leftarrow \min_v \{c(x, v) + D_v(y)\}$  for each node  $y \in N$ .
- Normalerweise konvergiert  $D_x(y)$  zu den tatsächlich niedrigsten Kosten  $dx(y)$ .

# DISTANZ-VEKTOR - BEISPIEL

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

**node x table**

|      | cost to  |          |          |
|------|----------|----------|----------|
| from | x        | y        | z        |
| x    | 0        | 2        | 7        |
| y    | $\infty$ | $\infty$ | $\infty$ |
| z    | $\infty$ | $\infty$ | $\infty$ |

**node y table**

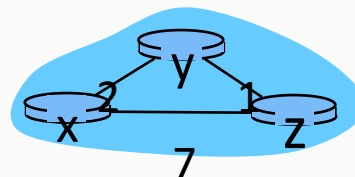
|      | cost to  |          |          |
|------|----------|----------|----------|
| from | x        | y        | z        |
| x    | $\infty$ | $\infty$ | $\infty$ |
| y    | 2        | 0        | 1        |
| z    | $\infty$ | $\infty$ | $\infty$ |

**node z table**

|      | cost to  |          |          |
|------|----------|----------|----------|
| from | x        | y        | z        |
| x    | $\infty$ | $\infty$ | $\infty$ |
| y    | $\infty$ | $\infty$ | $\infty$ |
| z    | 7        | 1        | 0        |

**node x table**

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 3 |
| y    | 2       | 0 | 1 |
| z    | 7       | 1 | 0 |



# DISTANZ-VEKTOR - BEISPIEL

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$

**node x  
table**

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 7 |
| y    | ∞       | ∞ | ∞ |
| z    | ∞       | ∞ | ∞ |

**node y  
table**

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | ∞       | ∞ | ∞ |
| y    | 2       | 0 | 1 |
| z    | ∞       | ∞ | ∞ |

**node z  
table**

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | ∞       | ∞ | ∞ |
| y    | ∞       | ∞ | ∞ |
| z    | 7       | 1 | 0 |

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 3 |
| y    | 2       | 0 | 1 |
| z    | 7       | 1 | 0 |

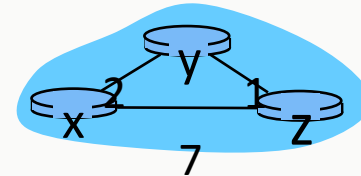
|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 7 |
| y    | 2       | 0 | 1 |
| z    | 7       | 1 | 0 |

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 7 |
| y    | 2       | 0 | 1 |
| z    | 3       | 1 | 0 |

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 3 |
| y    | 2       | 0 | 1 |
| z    | 3       | 1 | 0 |

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 3 |
| y    | 2       | 0 | 1 |
| z    | 3       | 1 | 0 |

|      | cost to |   |   |
|------|---------|---|---|
| from | x       | y | z |
| x    | 0       | 2 | 3 |
| y    | 2       | 0 | 1 |
| z    | 3       | 1 | 0 |



time

## Änderungen der Kosten

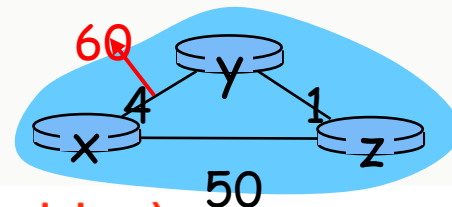
- Knoten erkennt lokales Update der Kosten...
- ... aktualisiert Erreichbarkeitsinformationen und berechnet neuen Distanz-Vektor für betroffene Ziele...
- ... und informiert die Nachbarn bei Bedarf.

“Good news travel fast”:

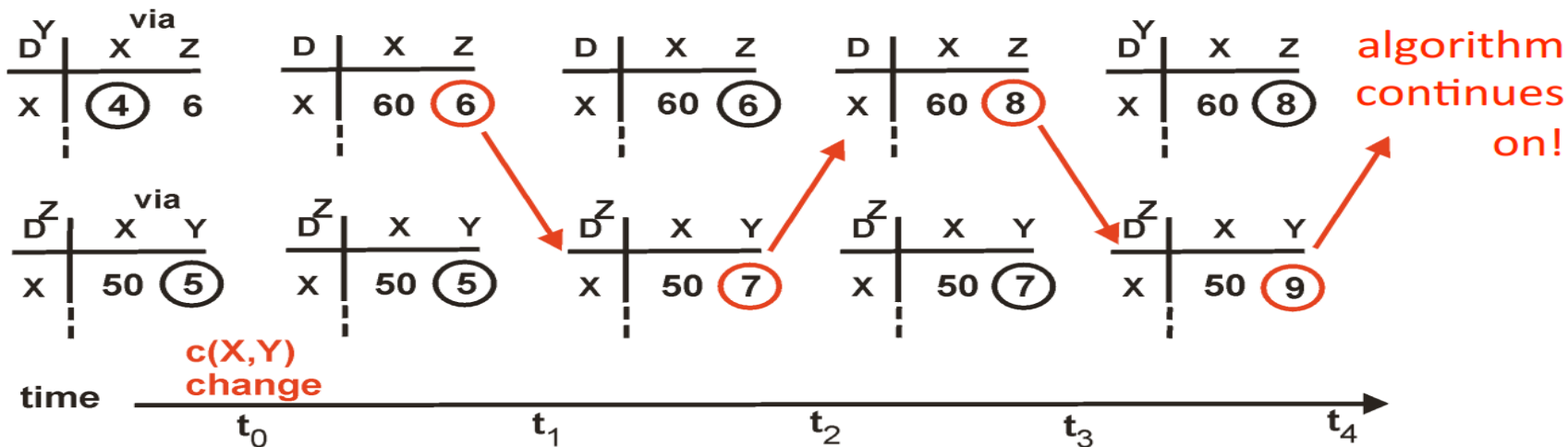
- $t_0$  : y erkennt Kostenupdate, aktualisiert eigene DV und informiert Nachbarn.
- $t_1$  : z erhält Update von y, aktualisiert seine Tabelle, berechnet neue Kosten zu x und sendet eigenen DV an alle Nachbarn.
- $t_2$  : y erhält Update von z, aktualisiert seine Tabelle. Die Kosten für y ändern sich nicht, also sendet y auch keine Nachricht an z.

# DISTANZ-VEKTOR ROUTING PROTOCOL

- “Bad news travel slow”:
- “Count to infinity”-Problem:



## View of X (about neighbor y and z's routing tables)



# INTERNET ROUTING PROTOCOL

- Das Internet als Netzwerk von Netzwerken wird von unterschiedlichen Organisationen betrieben (Netzwerke werden Autonomes System genannt).
- Daher sind Richtlinien beim Routing relevant!
- Jedes Netzwerk nutzt unabhängig von den anderen ein internes Routing-Protokoll ("Intradomain" oder "Interior Gateway Protocol").
- Alle Netzwerke müssen ein gemeinsames ("Exterior-") Interdomain-Protokoll unterstützen.
- Ist dabei ein Link-State- oder ein Distanz-Vektor-Protokoll besser?

# INTERNET ROUTING PROTOCOL

- Das Internet als Netzwerk von Netzwerken wird von unterschiedlichen Organisationen betrieben (Netzwerke werden Autonomes System genannt).
- Daher sind Richtlinien beim Routing relevant!
- Jedes Netzwerk nutzt unabhängig von den anderen ein internes Routing-Protokoll ("Intradomain" oder "Interior Gateway Protocol").
- Alle Netzwerke müssen ein gemeinsames ("Exterior-") Interdomain-Protokoll unterstützen.
- Ist dabei ein Link-State- oder ein Distanz-Vektor-Protokoll besser?
- Link-State?
  - Jeder Router muss die gesamte Internet-Topologie kennen.
  - Keine Möglichkeit für Richtlinien!



# INTERNET ROUTING PROTOCOL

- Das Internet als Netzwerk von Netzwerken wird von unterschiedlichen Organisationen betrieben (Netzwerke werden Autonomes System genannt).
- Daher sind Richtlinien beim Routing relevant!
- Jedes Netzwerk nutzt unabhängig von den anderen ein internes Routing-Protokoll ("Intradomain" oder "Interior Gateway Protocol").
- Alle Netzwerke müssen ein gemeinsames ("Exterior-") Interdomain-Protokoll unterstützen.
- Ist dabei ein Link-State- oder ein Distanz-Vektor-Protokoll besser?
- Distanz-Vektor?
  - Dynamische Verteilung des kürzesten/günstigsten Pfades.
  - Auch keine Möglichkeit für Richtlinien!

- Geschichte des Internets
  - ARPANet
  - NSFNet
  - „Das Internet“
- Internet Protocol
- Wie Alice und Bob über das Internet kommunizieren
- Netzwerk von Netzwerken (Autonome Systeme)
- Netzwerk-Routing Protokolle
- **Border Gateway Protocol**

# BORDER GATEWAY PROTOCOL (BGP)

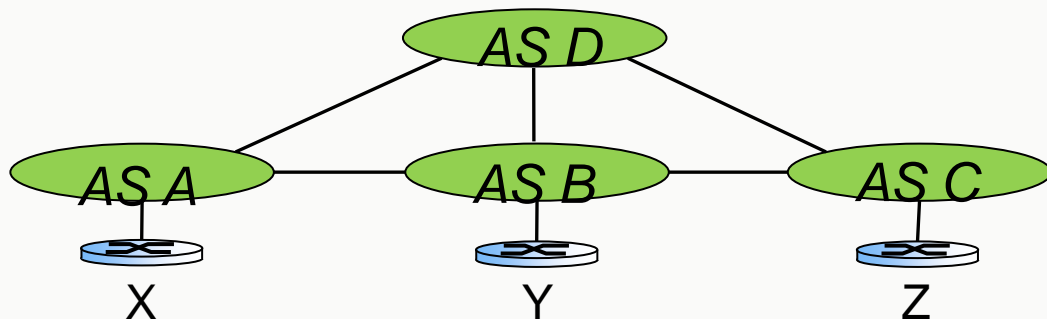
BGP is the Internet's de-facto standard interdomain routing protocol.

Allows policy base routing decisions such as:

- Do not carry commercial traffic on educational networks.
- Use the cheaper upstream provider even if the performance is bad.
- For VoIP and Video stream use the expensive provider instead.
- Do not route over the US to reach a partner also located in Germany! (US have Not Specified Adversaries, thank you Edward Snowden)

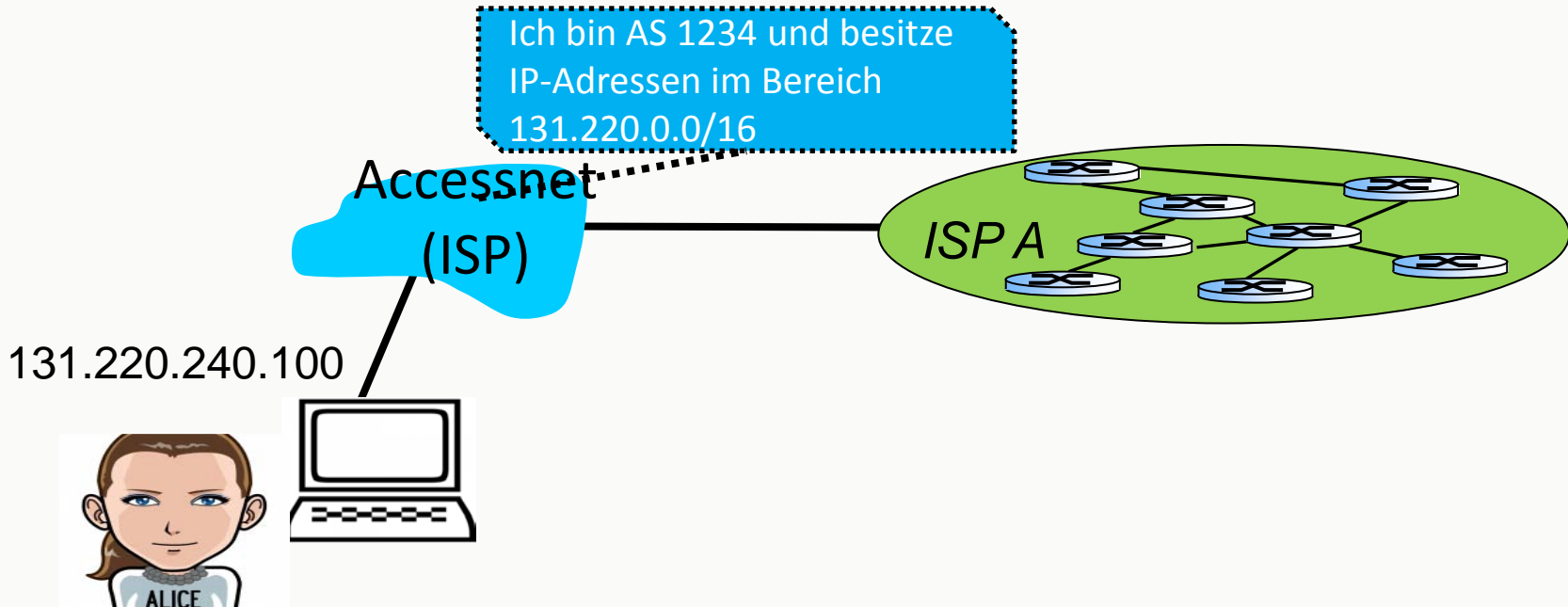
Policies allow a hierarchical routing system having providers, customers and peers.

# BORDER GATEWAY PROTOCOL (BGP)



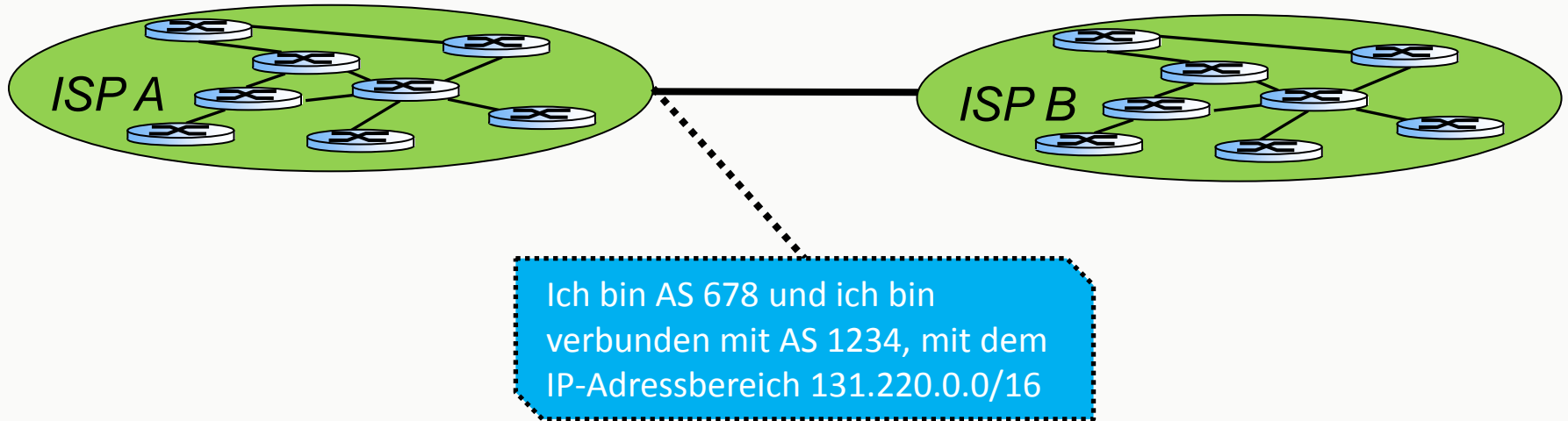
# INTERNETROUTING MIT BGP

- Autonome Systeme annoncieren Erreichbarkeitsinformationen.



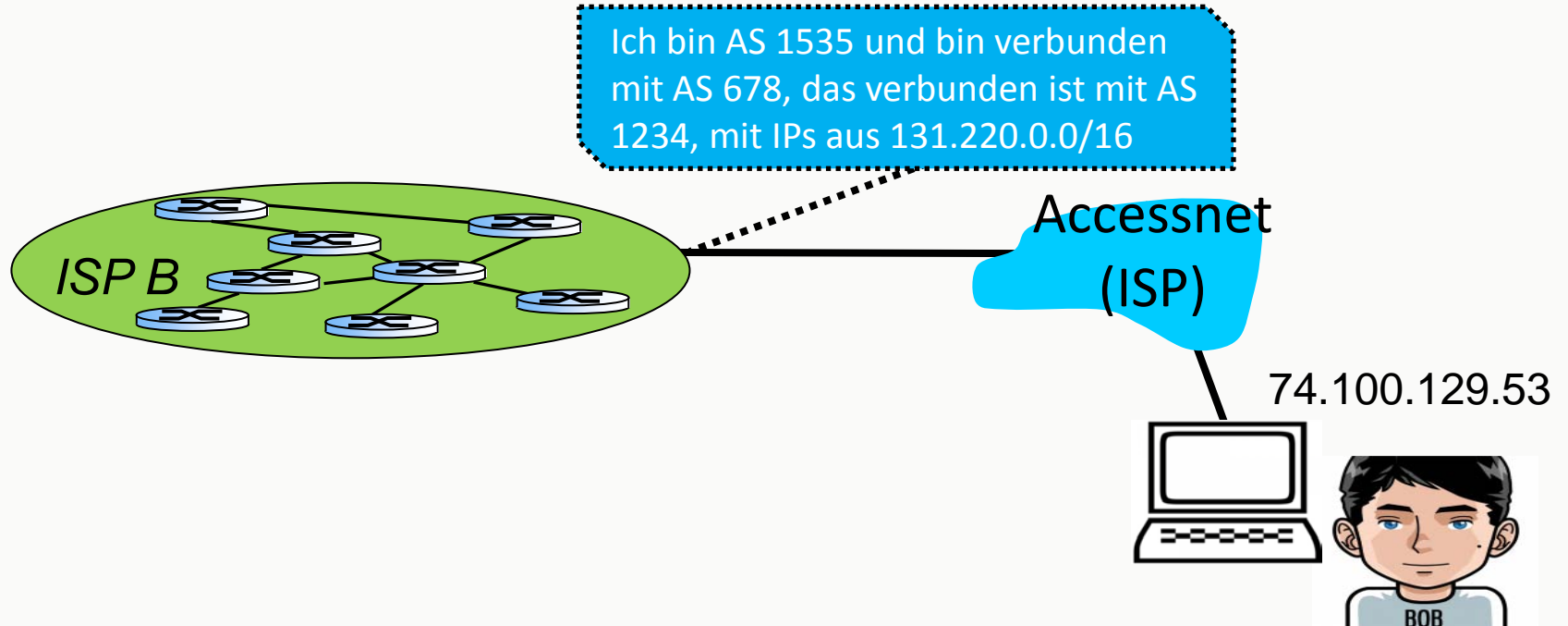
# INTERNETROUTING MIT BGP

- Autonome Systeme annoncieren Erreichbarkeitsinformationen.



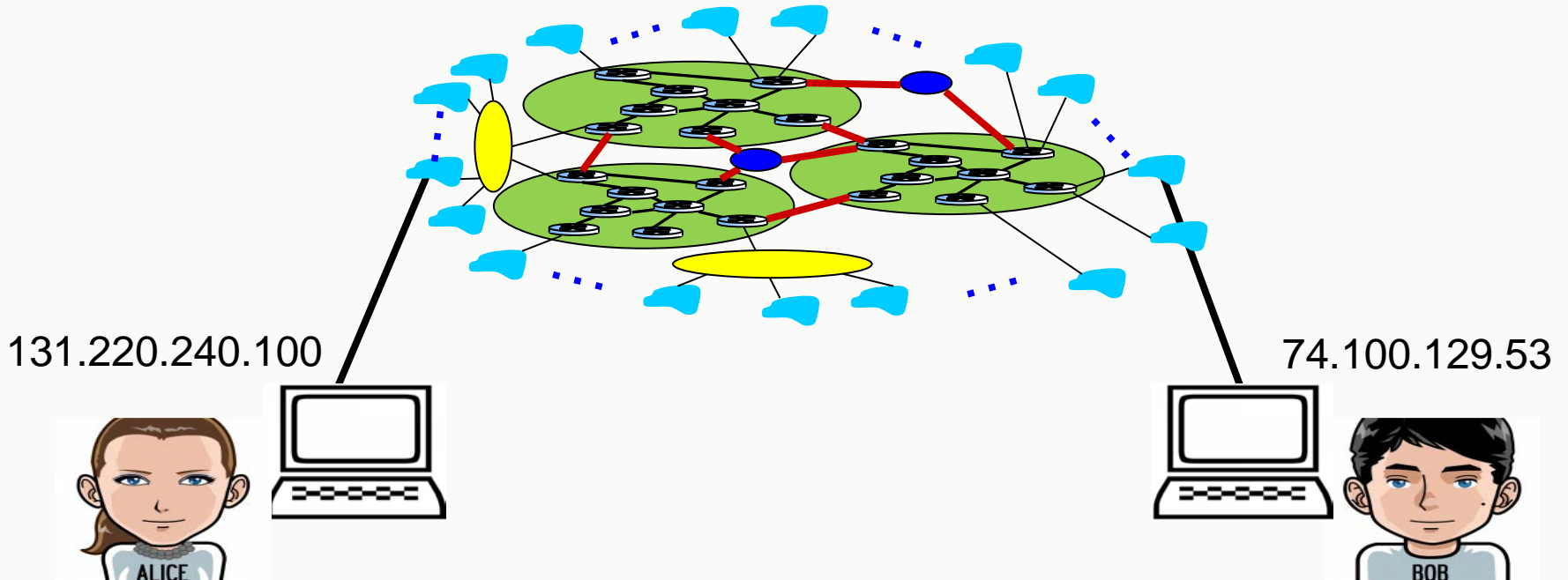
# INTERNETROUTING MIT BGP

- Autonome Systeme annoncieren Erreichbarkeitsinformationen.



# INTERNETROUTING MIT BGP

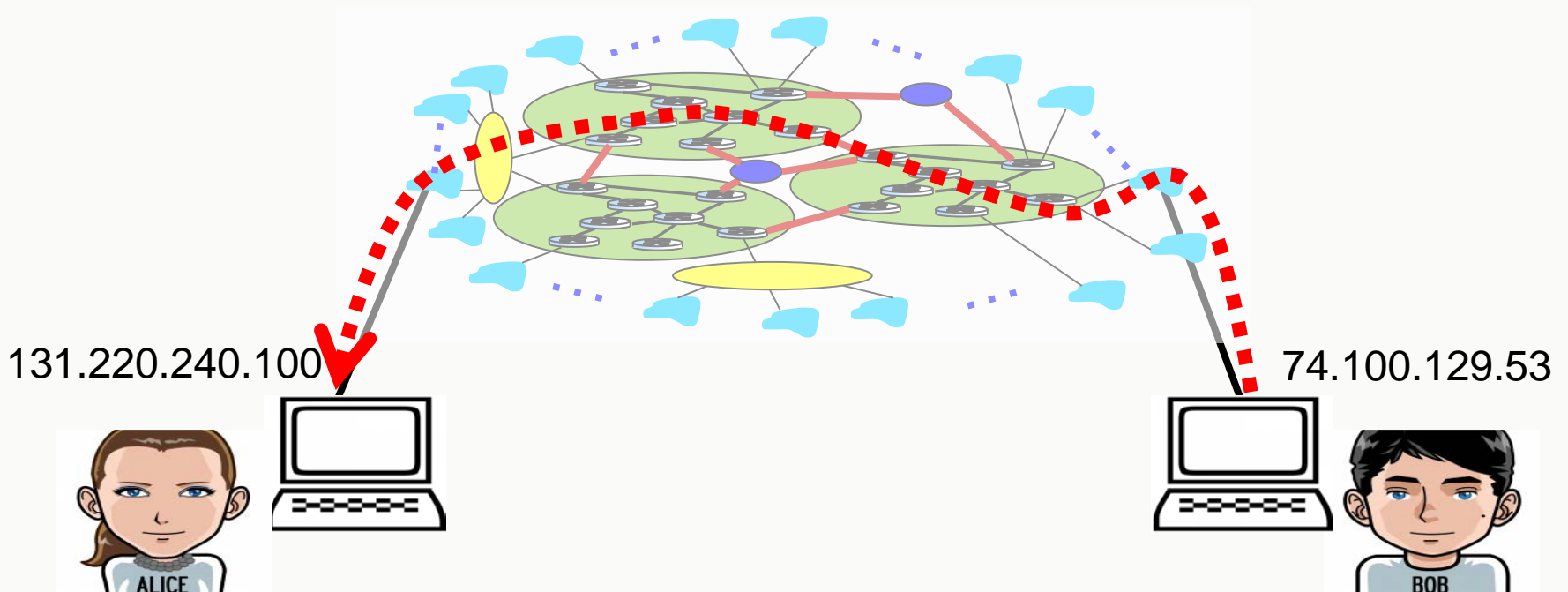
- Also wie kommt jetzt die Nachricht von Bob zu Alice?





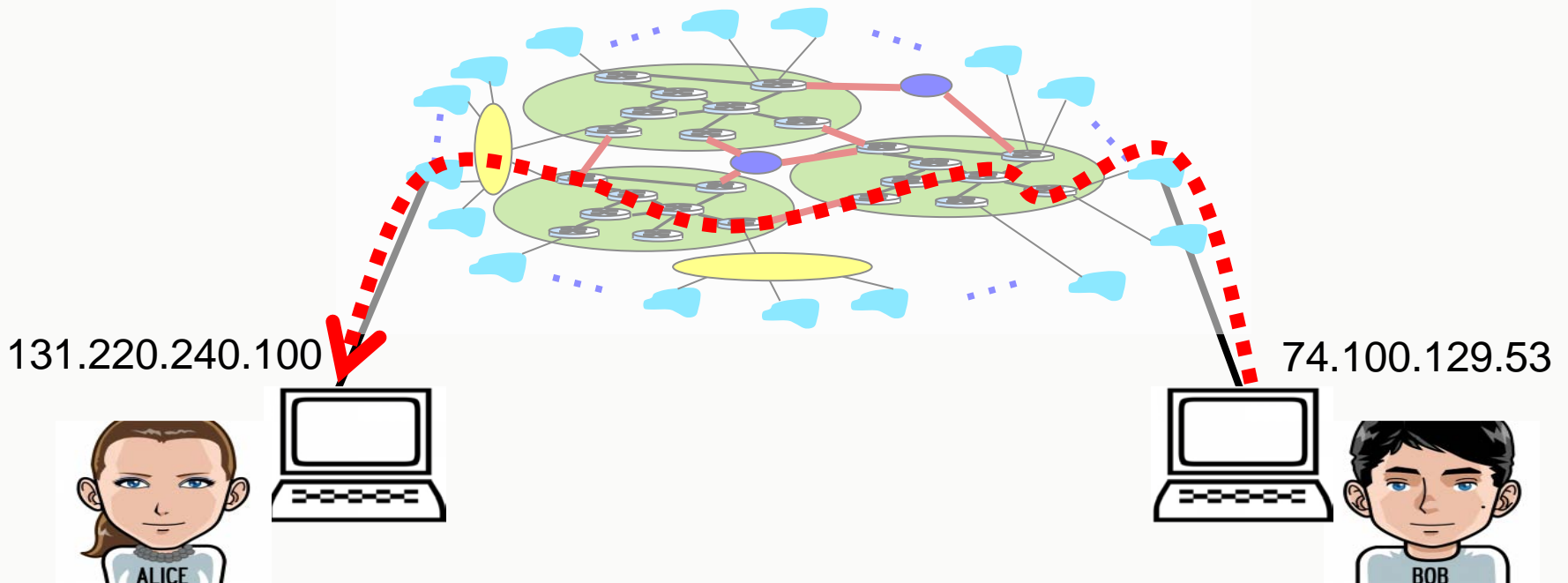
# INTERNETROUTING MIT BGP

- Es kann dieser Pfad sein, ....



# INTERNETROUTING MIT BGP

- ... oder dieser Pfad, ... oder irgendein anderer (kreisfreier) Pfad.



Vielen Dank für die Aufmerksamkeit!

Fragen?

Nächste Vorlesung:

- Montag, 22. Mai 2023

Nächste Übung:

- Dienstag, 16. Mai 2023 – 16 Uhr
- Abgabe des Übungszettels bis morgen – 16 Uhr