Routing Principles

Elisa Jasinska & Robert Scheffler

What to expect?

- Networking Introduction
- Link State Routing
  - Dijkstra
  - OSPF
- Distance Vector Routing
  - Bellman-Ford
  - RIP
  - BGP
  - DSDV

What to expect?

- Networking Introduction
- Link State Routing
  - Dijkstra
  - OSPF
- Distance Vector Routing
  - Bellman-Ford
  - RIP
  - BGP
  - DSDV

Why routing?

Networking

- Different networks...
  - Circuit switched (phone network)
  - Packet switched (internet)
- ... all need routing

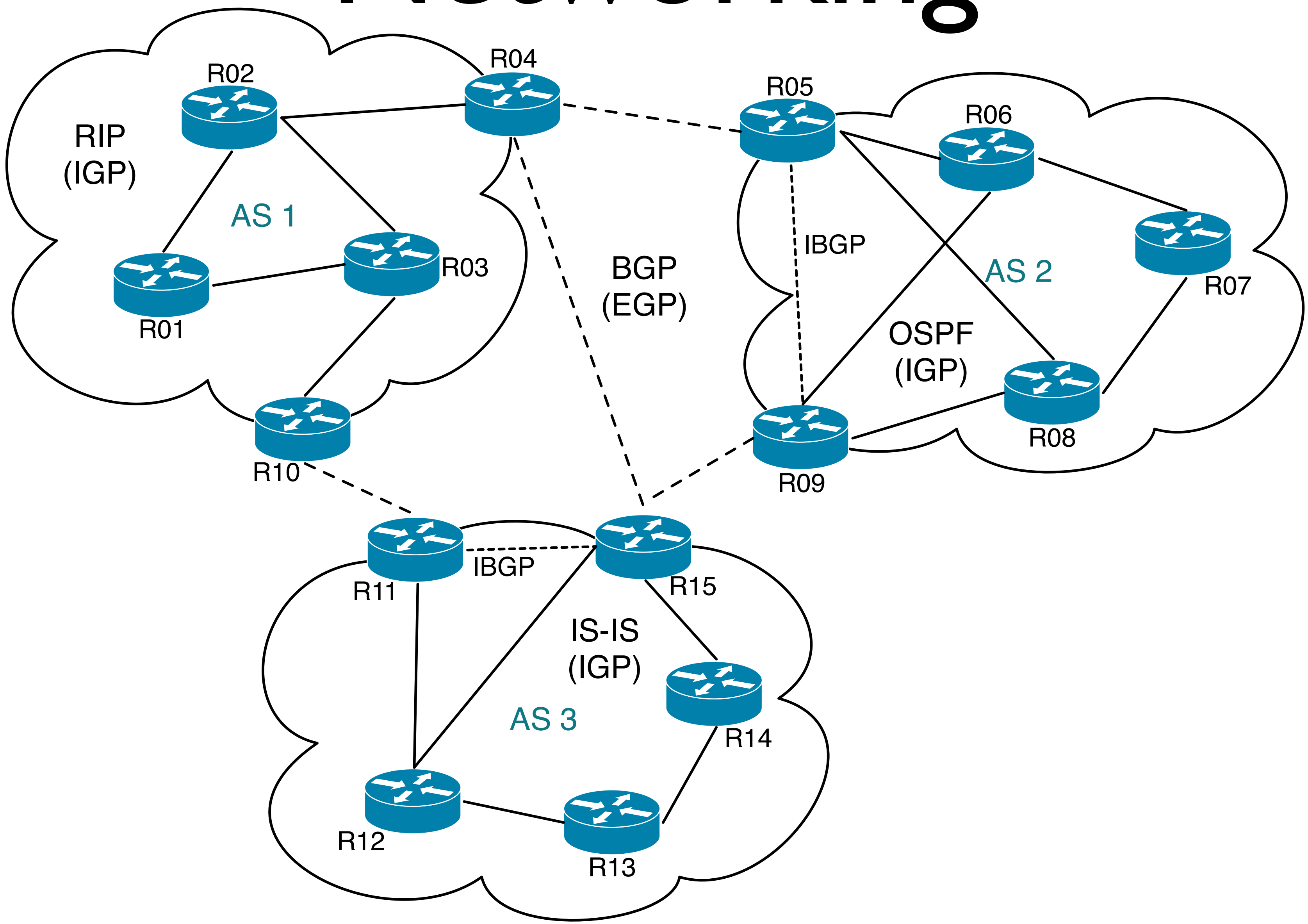
Networking

- Topologies may change
- ... thus we need dynamic routing ...
  - Routing algorithms
  - Routing tables

Classification

Global information	Decentralized information
All routers have complete topology Link-State Algorithms	Router only knows its connected neighbors Distance-Vector Algorithms

Networking


R02
R04
R05
R06
RIP
(IGP)
AS 1
R03
R01
BGP
(EGP)
IBGP
AS 2
R07
OSPF
(IGP)
R08
R10
R09
R11
IBGP
R15
IS-IS
(IGP)
AS 3
R14
R12
R13

Networking

- Exterior Gateway Protocols (EGP)
  - BGP
  - Inter domain routing
- Interior Gateway Protocol (IGP)
  - OSPF, IS-IS, RIP
  - Networks under your control

What to expect?

- Networking Introduction
- Link State Routing
  - Dijkstra
  - OSPF
- Distance Vector Routing
  - Bellman-Ford
  - RIP
  - BGP
  - DSDV

Link State Routing

- Invented in 1978 by John McQuillan
- First use in the ARPANET
- Link-state routing protocols: OSPF and IS-IS

Link State Routing

- Global information
- Topology and link costs known to all hosts

  ➡ Distributed map

- Computes least cost paths to all other nodes (Dijkstra)

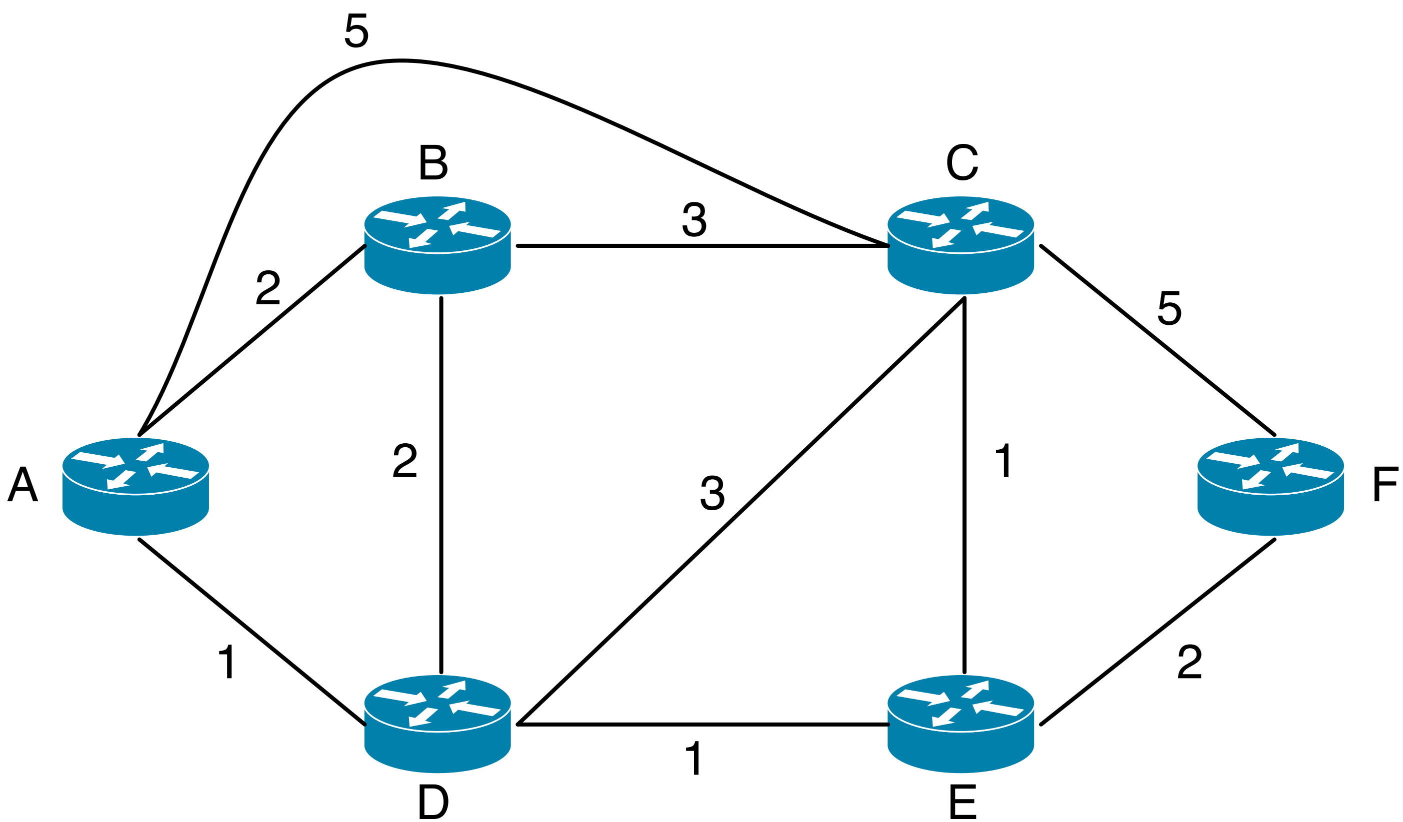
  ➡ Routing table

Link State Advertisements

- Each node ...
  - Checks for its directly connected neighbors
  - Sends link-state advertisements through the network
  - Computes a topology out of the received advertisements

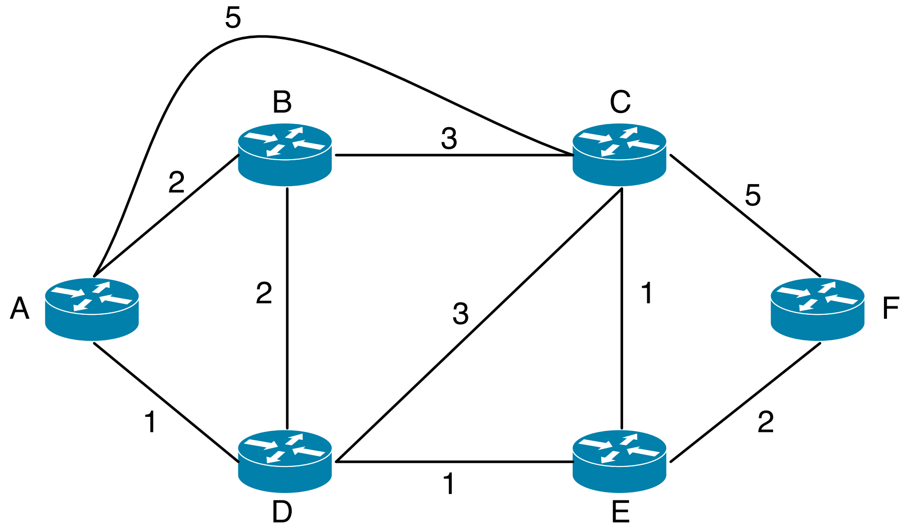
Dijkstra's Algorithm

- Calculating the routing table

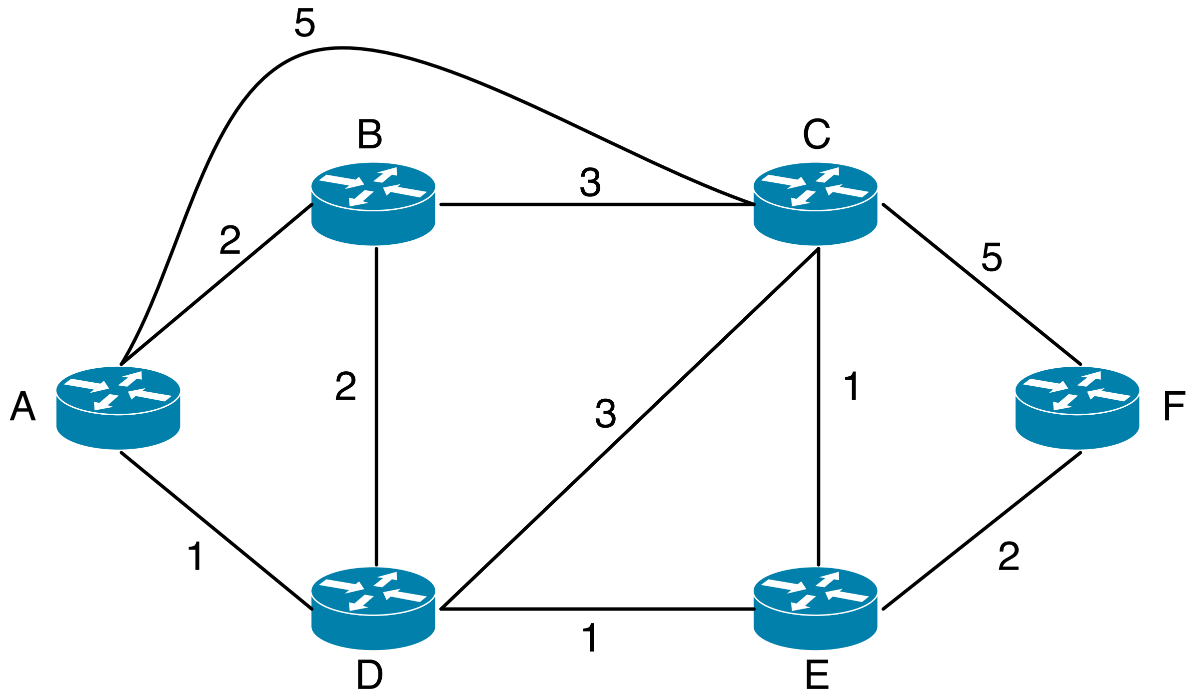


Dijkstra's Algorithm

```
Initialization for A:
  N' = {A}
  for all nodes v
    if v adjacent to A
      then D(v) = c(A,v)
      else D(v) = ∞
```



Dijkstra's Algorithm



N'	D(B)	D(C)	D(D)	D(E)	D(F)
A	2	5	1	∞	∞

Dijkstra’s Algorithm

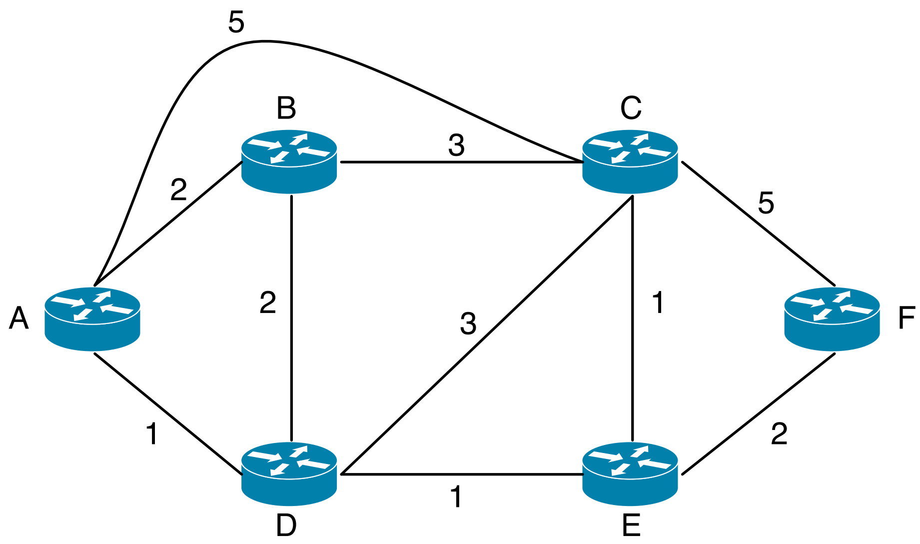
```
Loop
  find w not in N’ such that D(w) is a min
  add w to N’
  update D(v) for all v adjacent to w
   and not in N’:
    D(v) = min( D(v), D(w) + c(w,v) )

   /* new cost to v is either old cost to v
   or known shortest path cost to w plus
   cost from w to v*/

until all nodes in N’
```

Dijkstra's Algorithm

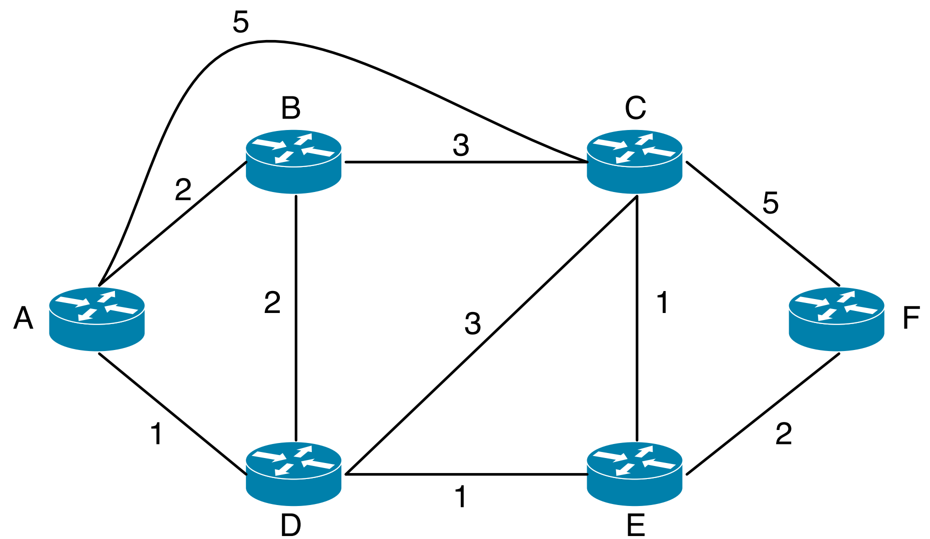
- Select D
- Check D's neighbors



N'	D(B)	D(C)	D(D)	D(E)	D(F)
A	2	5	1	∞	∞
AD	2	4		2	∞

Dijkstra's Algorithm

- Select E
- Check E's neighbors

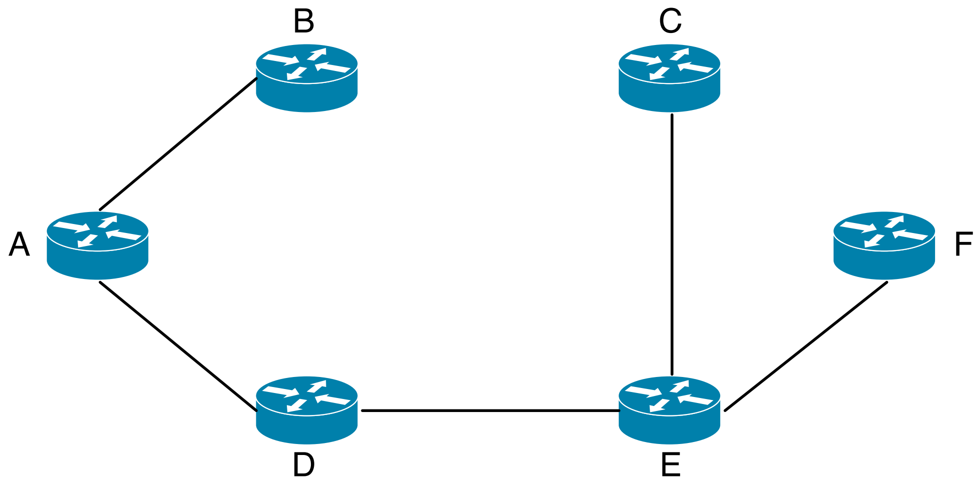


N'	D(B)	D(C)	D(D)	D(E)	D(F)
A	2	5	1	∞	∞
AD	2	4		2	∞
ADE	2	3			4

Dijkstra's Algorithm

N'	D(B)	D(C)	D(D)	D(E)	D(F)
A	2	5	1	∞	∞
AD	2	4		2	∞
ADE	2	3			4
ADEB		3			4
ADEBC					4
ADEBCF					

Shortest Path Tree Routing Table



Destination	Next-Hop
B	B
D	D
E	D
C	D
F	D

Dijkstra's Algorithm

- Complexity
  - Each iteration checks all nodes in N
  - $O(n^2 + m)$
  - Better implementations $O(n * \log n + m)$
- Loops if maps are not identical

OSPF

- Interior Gateway Protocol
- Link-State algorithm (Dijkstra)
- Path cost as routing metric (bandwidth...)
- Incremental updates, after sending the whole routing table

OSPF

- Checks link state by sending Hello packet periodically
- If no response, flood network with updates to a multicast address
- Updates must be acknowledged to ensure identical maps at each node

What to expect?

- Networking Introduction
- Link State Routing
  - Dijkstra
  - OSPF
- Distance Vector Routing
  - Bellman-Ford
  - RIP
  - BGP
  - DSDV

Distance Vector Routing

- Main routing principle for decades
- Based on Bellman-Ford algorithm
- First implementation 1969 (ARPANET)

Bellman-Ford

- Weighted digraph
- Initialize distances with infinity
- n times: iterate through all edges and update destination vertex (distance and predecessor)
- Check for negative cycles

Bellman-Ford

- Worse running time than Dijkstra O(n*m)
- Works for negative edge weights
- Can be implemented decentralized

Routing Information Protocol

- Different implementations of Distance Vector Routing
- Proposed unification in RFC 1058 (1988)
- Later revisions: RIP2 (1994), RIPng (1997)

Routing Information Protocol

- Each node keeps routing entry for each destination in the network
- Send routing table to neighbors regularly
- Neighbors compare entries

Routing Information Protocol

- Routing table entry consists of...
  - Address: destination IP
  - Gateway: first hop to destination
  - Interface: hardware interface to use
  - Metric: number of hops
  - Timer: time of last update

Routing Information Protocol

- Timeout
- Counting to infinity
- Split horizon
- Poisoned reverse
- Triggered updates

Border Gateway Protocol

- De facto standard for inter-AS Routing (BGP v1: 1989, BGP v4: 1995)
- Path vector protocol
  - Routing entry contains whole path (loop detection)
  - Administrator can implement policies

Destination-Sequenced Distance-Vector Routing

- Distance Vector Routing for wireless ad-hoc networks
- Rapid topology changes
- Broadcast instead of point-to-point

Destination-Sequenced Distance-Vector Routing

- Idea: numbering of update packets to differentiate between new and stale routes
- Propagate sequence numbers through route
- Discard updates with older sequence numbers

Destination-Sequenced Distance-Vector Routing

- Timeout information uses odd numbers, “real” updates use even numbers
- No instant update for new routes

Thanks for listening!
Questions?

Elisa Jasinska - jasinska@informatik.hu-berlin.de
Robert Scheffler - scheffle@informatik.hu-berlin.de