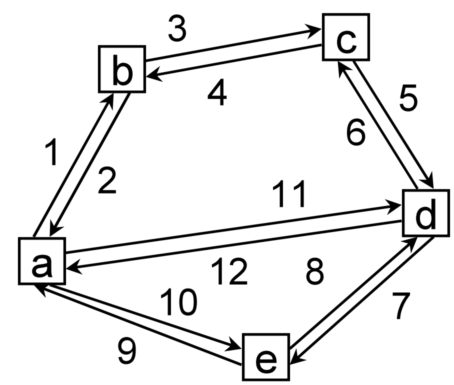
12. Traffic engineering

Contents

- Topology
- Traffic matrix
- Traffic engineering
- Load balancing

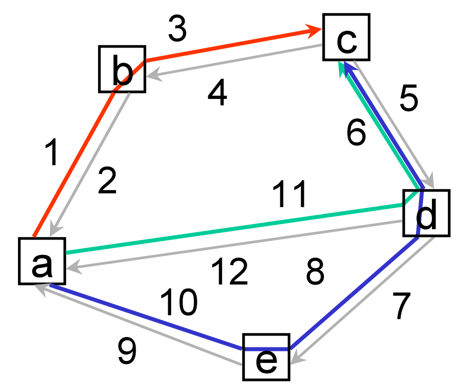
Topology

- A telecommunication network consists of nodes and links
  - Let N denote the set of nodes indexed with n
  - Let J denote the set of nodes indexed with j
- Example:
  - $N = \{\mathrm{a,b,c,d,e}\}$
  - $J = \{1,2,3,\ldots,12\}$
  - link 1 from node a to node b
  - link 2 from node b to node a
- Let c_j denote the capacity of link j (bps)



Paths

- We define a **path** (= route) as a
  - set of consecutive links connecting two nodes
  - Let P denote the set of paths indexed with p
- Example:
  - three paths from node a to node c:
    - red path consisting of links 1 and 3
    - green path consisting of links 11 and 6
    - blue path consisting of links 10, 8 and 6



Path matrix

- Each path consists of a set of links
- This connection is described by the **path matrix** $\mathbf{A}$,for which
  - element $a_{jp} = 1$ if $j \in p$, that is, link j belongs to path p
  - otherwise $a_{jp} = 0$
- Example:
  - three columns of a path matrix

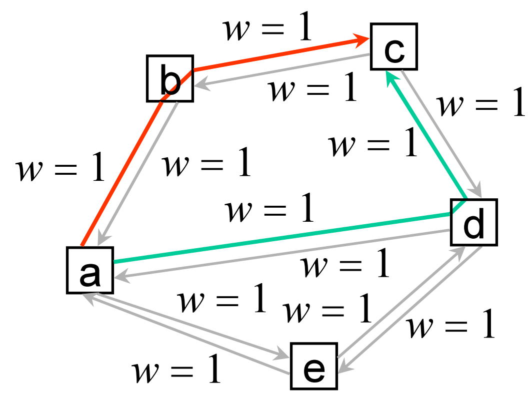
	ac1	ac2	ac3
1	1	0	0
2	0	0	0
3	1	0	0
4	0	0	0
5	0	0	0
6	0	1	1
7	0	0	0
8	0	0	1
9	0	0	0
10	0	0	1
11	0	1	0
12	0	0	0

Shortest paths

- If each link j is associated with a correponding weight w_j, the length l_p of path p is given by

$$l_p = \sum_{j \in p} w_j$$

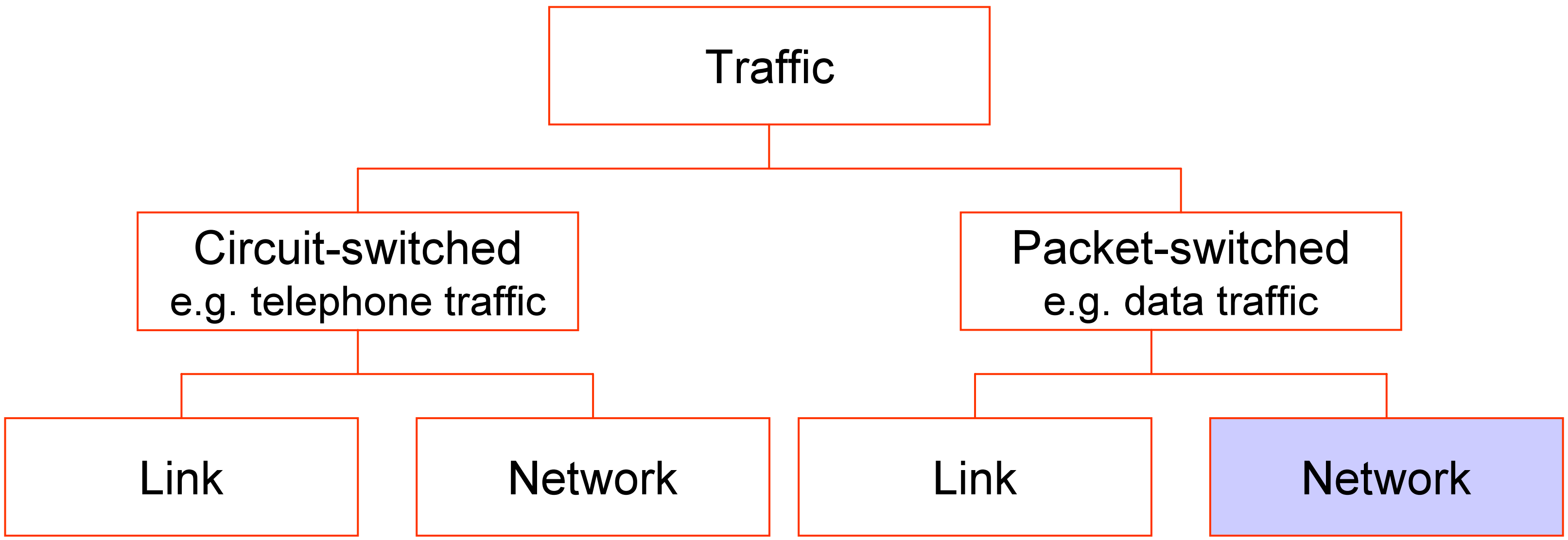
  - With unit link weights $w_j = 1$, path length = hop count
- Example:
  - two shortest paths (of length 2) from node a to node c



Contents

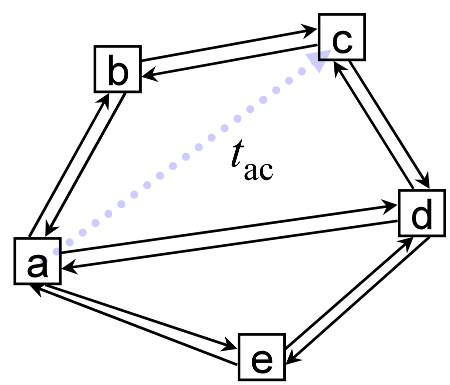
- Topology
- Traffic matrix
- Traffic engineering
- Load balancing

Traffic characterisation



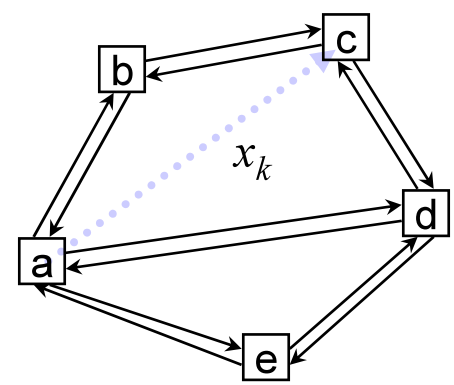
Traffic matrix (1)

- Traffic in a network is described by the **traffic matrix** $\mathbf{T}$, for which
  - element t_{nm} tells the **traffic demand** (bps) from origin node n to destination node m
  - Aggregated traffic of all flows with the same origin and destination
  - Aggregated traffic during a time interval, e.g. busy hour or ”typical 5-minute interval”
- Example:
  - Traffic demand from origin a to destination c is t_{ac} (bps)



Traffic matrix (2)

- Below we present the traffic demands in a vector form
  - Let K denote the set of origin-destination pairs (**OD-pairs**) indexed with k
- Traffic demands constitute a vector $\mathbf{x}$, for which
  - element x_k tells the traffic demand of OD-pair k
- Example:
  - if OD-pair $(\mathrm{a,c})$ is indexed with k, then $x_k = t_{\mathrm{ac}}$



Contents

- Topology
- Traffic matrix
- Traffic engineering
- Load balancing

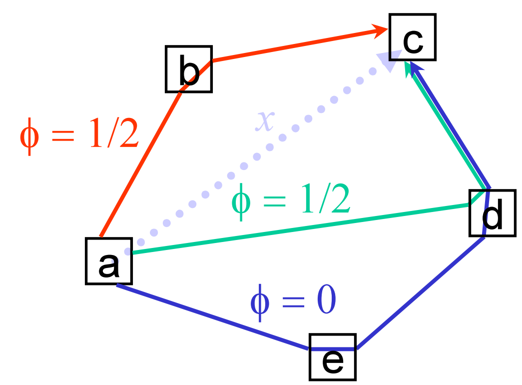
Traffic engineering and network design

- **Traffic engineering** = ”Engineer the traffic to fit the topology”
  - Given a fixed topology and a traffic matrix, how to **route** these traffic demands?
- **Network design** = ”Engineer the topology to fit the traffic”

Effect of routing on load distribution

- Routing algorithm determines how the traffic load is distributed to the links
  - Internet routing protocols (RIP, OSPF, BGP) apply the shortest path algorithms (Bellman-Ford, Dijkstra)
  - In MPLS networks, other algorithms are also possible
- More precisely: routing algorithm determines the proportions (**splitting ratios**) ϕ_{pk} of traffic demands x_k allocated to paths p,

$$\sum_{p \in P} \phi_{pk} = 1 \quad \text{for all } k$$



Link counts

- Traffic on a path p between OD-pair k is thus

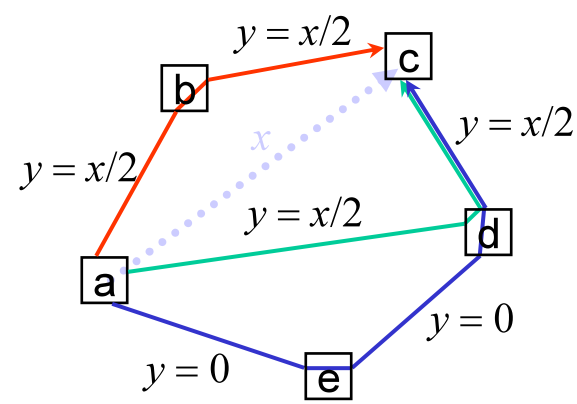
$$\phi_{pk} x_k$$

- **Link counts** y_j are determined by traffic demands x_k and splitting ratios ϕ_{pk}:

$$y_j = \sum_{p \in P} \sum_{k \in K} a_{jp} \phi_{pk} x_k$$

- The same in matrix form:

$$\mathbf{y} = \mathbf{A} \phi \mathbf{x}$$



MPLS

- **MPLS** (Multiprotocol Label Switching) supports traffic load distribution to parallel paths between OD-pairs
  - In MPLS networks, there can be any number of parallel Label Switched Paths (LSP) between OD-pairs
  - These paths do not need to belong to the set of shortest paths
  - Each LSP is associated with a label and each MPLS packet is tagged with such a label
- MPLS packets are routed through the network via these LSP's (according to their label)
- Traffic load distribution can be affected **directly** by changing the splitting ratios ϕ_{pk} at the origin nodes

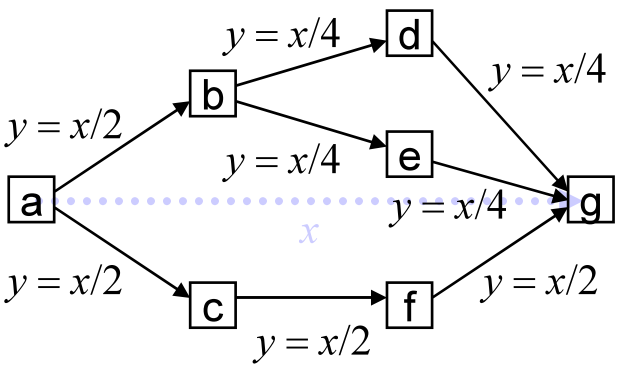
OSPF (1)

- **OSPF** (Open Shortest Path First) is an intradomain routing protocol in IP networks
- Link State Protocol
  - each node tells the other nodes the distance to its neighbouring nodes
  - these distances are the link weights for the shortest path algorithm
  - based on this information, each node is aware of the whole topology of the domain
  - the shortest paths are derived from this topology using Dijkstra's algorithm
- IP packets are routed through the network via these shortest paths

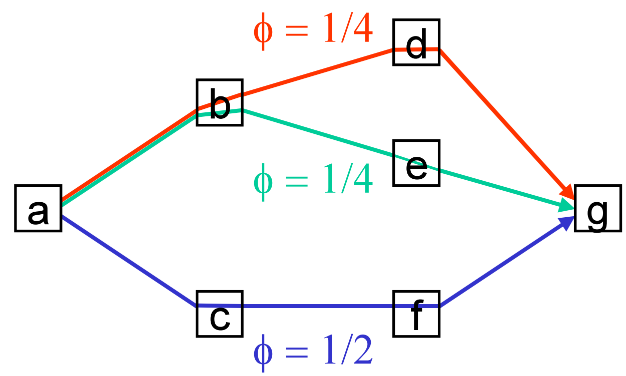
OSPF (2)

- Routers in OSPF networks typically apply **ECMP** (Equal Cost Multipath)
  - If there are multiple shortest paths from node n to node m, then node n tries to split the traffic uniformly to those outgoing links that belong to at least one of these shortest paths
  - However, this does **not** imply that the traffic load is distributed uniformly to all shortest paths! See the example on next slide.
- Traffic load distribution can be affected only **indirectly** by changing the link weights
  - splitting ratios ϕ_{pk} can not directly be changed
  - due to ECMP, the desired splitting ratios ϕ_{pk} may be out of reach

ECMP

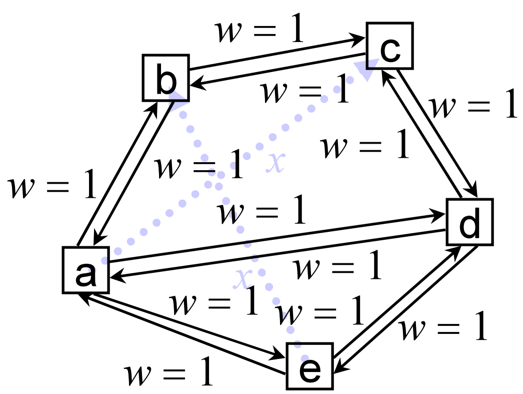


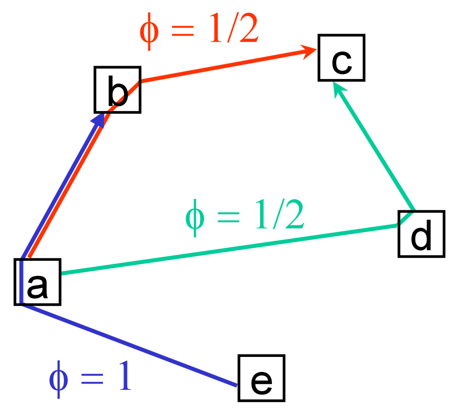
$y = x/4$
d
$y = x/4$
b
$y = x/2$
$y = x/4$
e
a
$y = x/4$
g
x
$y = x/2$
c
f
$y = x/2$
$y = x/2$

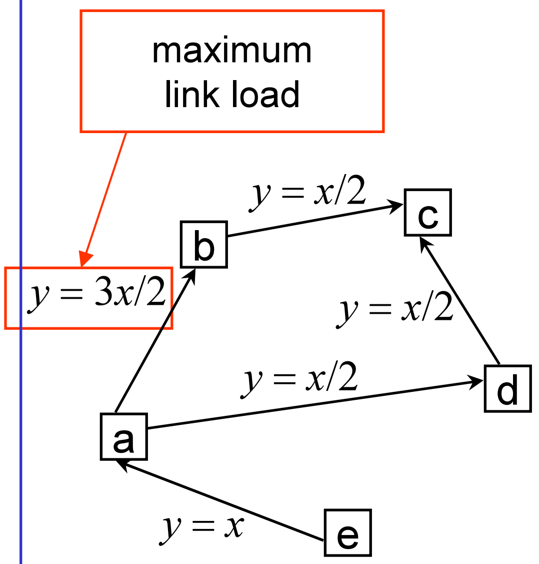


$\phi = 1/4$
d
b
$\phi = 1/4$
e
a
g
c
f
$\phi = 1/2$

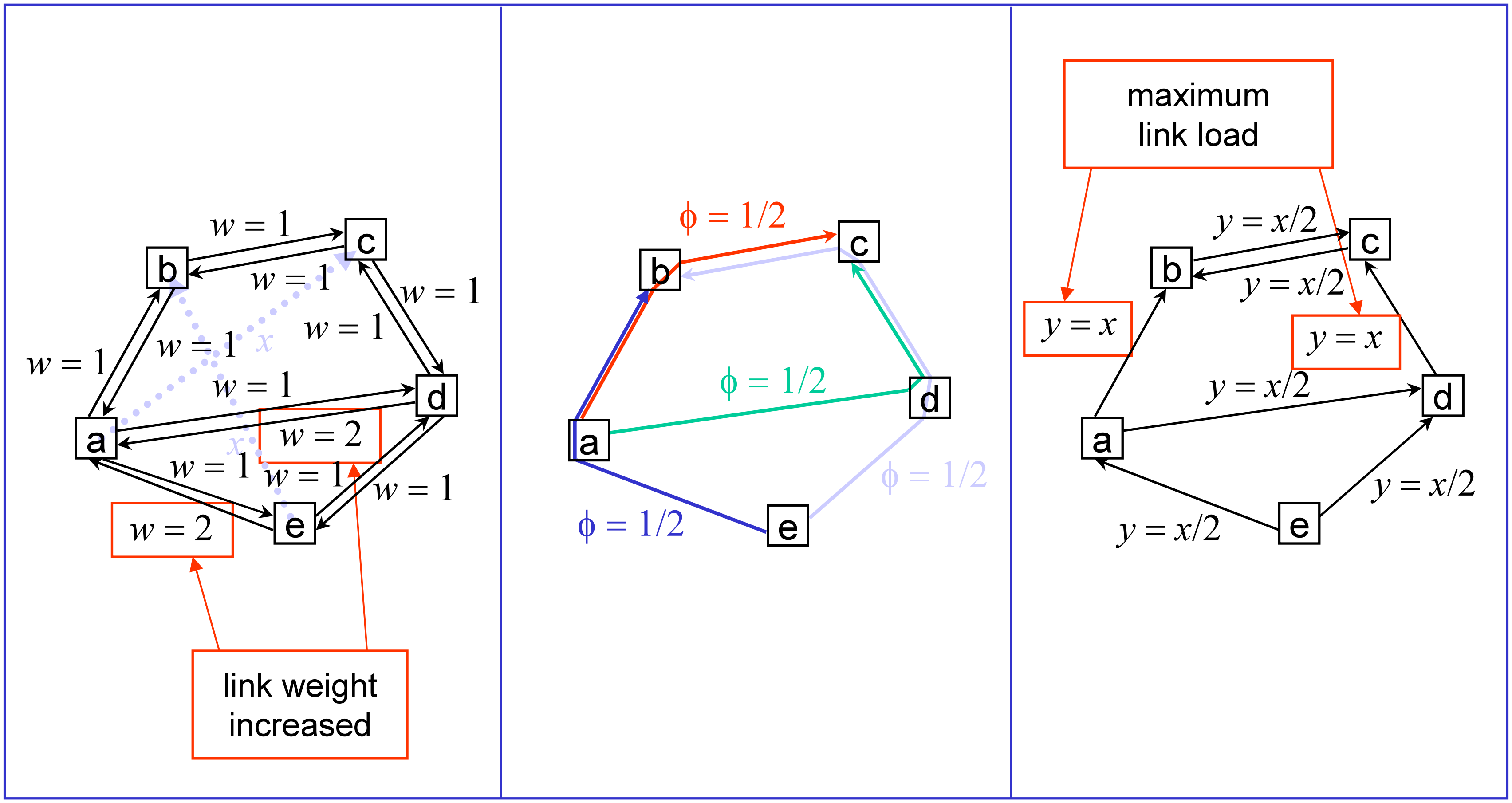
Effect of link weights on load distribution (1)







Effect of link weights on load distribution (2)

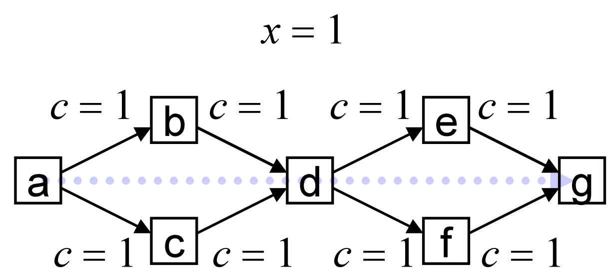


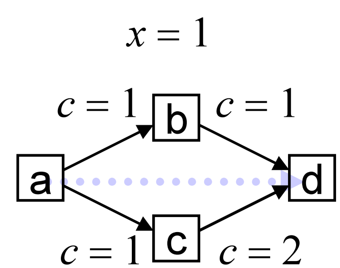
Contents

- Topology
- Traffic matrix
- Traffic engineering
- Load balancing

Load balancing problem (1)

- Given a fixed topology and a traffic matrix, how to **optimally route** these traffic demands?
- One approach is to equalize the relative load of different links, $\rho_j = y_j/c_j$
  - Sometimes this can be done in multiple ways (upper figure)
  - Sometimes it is not possible at all (lower figure)
  - In this case, we may, however, try to get as close as possible, e.g. by minimizing the maximum relative link load (called: load balancing problem)





Load balancing problem (2)

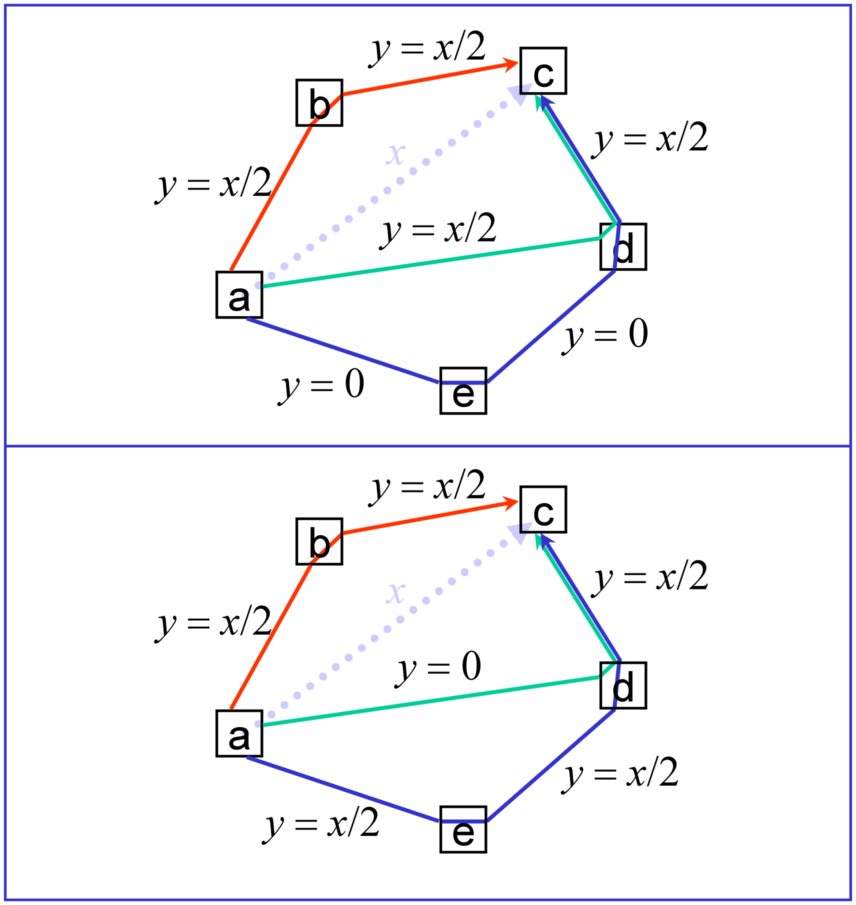
- **Load Balancing Problem**:
  - Consider a network with topology (N,J), link capacities c_j, and traffic demands x_k. Determine the splitting ratios ϕ_{pk} so that the maximum relative link load is minimized

$$\text{Minimize} \quad \max_{j \in J} \frac{y_j}{c_j}$$

$$\text{subject to} \quad \begin{cases} y_j = \sum_{p \in P} \sum_{k \in K} A_{jp} \phi_{pk} x_k & \forall j \in J \\ \sum_{p \in P} \phi_{pk} = 1 & \forall k \in K \\ \phi_{pk} \geq 0 & \forall p \in P, k \in K \end{cases}$$

Load balancing problem (3)

- Load Balancing Problem has always a solution but this might not be unique
- Example:
  - the same maximum link load is achieved with routes of different length
  - the upper routes are better due to smaller capacity consumption
- A reasonable unique solution is achieved by associating a negligible cost with all the hops along the paths used



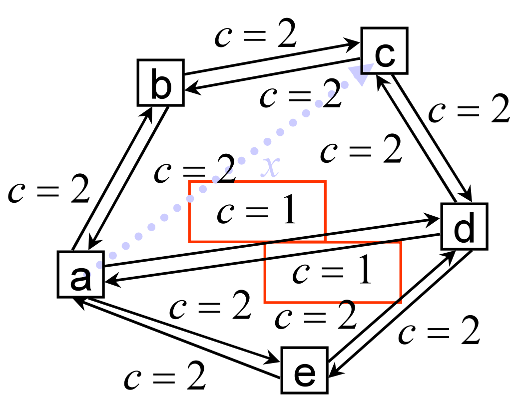
Load balancing problem (4)

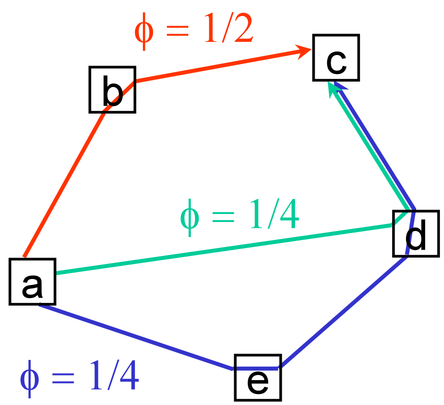
- **Load Balancing Problem** with a reasonable and unique solution:
  - Consider a network with topology (N,J), link capacities c_j, and traffic demands x_k. Determine the splitting ratios ϕ_{pk} so that the maximum relative link load is minimized with the smallest amount of required capacity

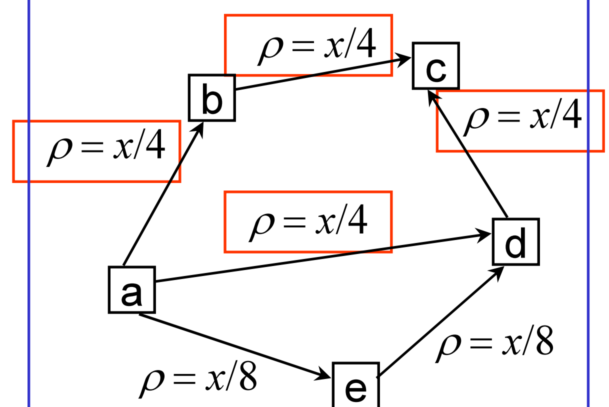
$$\text{Minimize} \quad \max_{j \in J} \frac{y_j}{c_j} + \varepsilon \sum_{j' \in J} y_{j'}$$

$$\text{subject to} \quad \begin{cases} y_j = \sum_{p \in P} \sum_{k \in K} A_{jp} \phi_{pk} x_k & \forall j \in J \\ \sum_{p \in P} \phi_{pk} = 1 & \forall k \in K \\ \phi_{pk} \geq 0 & \forall p \in P, k \in K \end{cases}$$

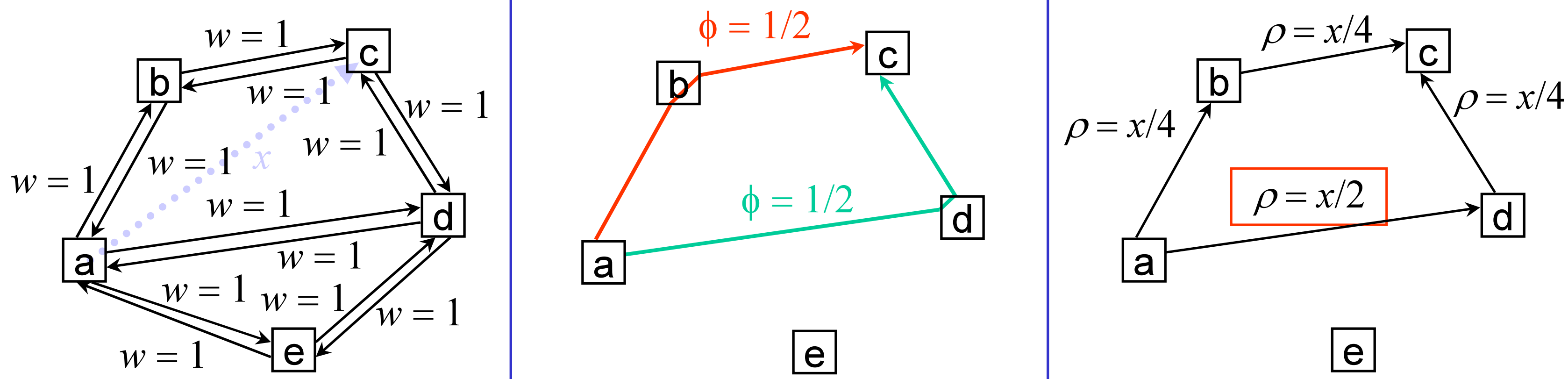
Example (1): optimal solution



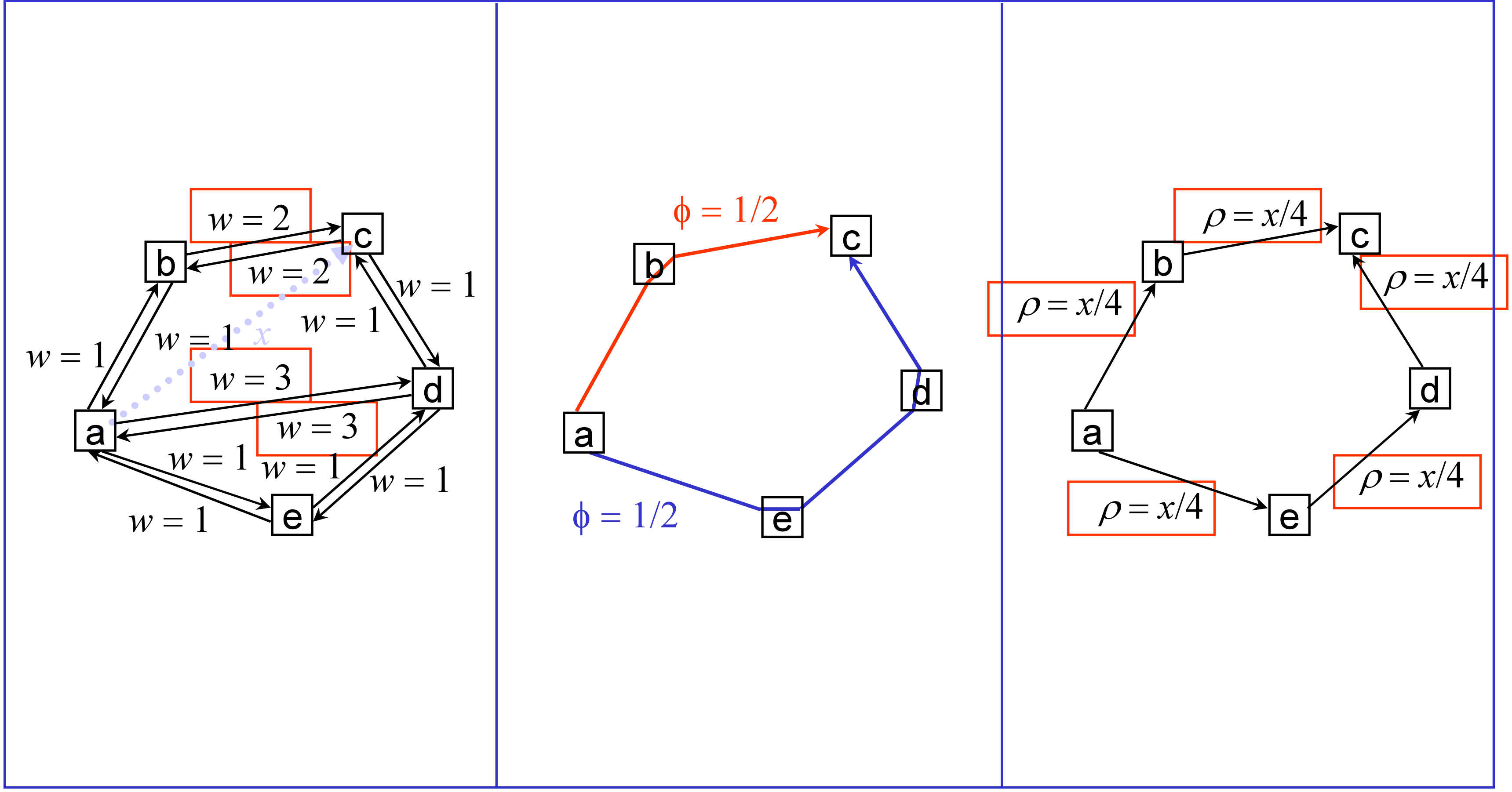




Example (2): link weights w = 1



Example (3): optimal link weights



THE END