Planning and Optimizing Large Scale Telecommunication Networks

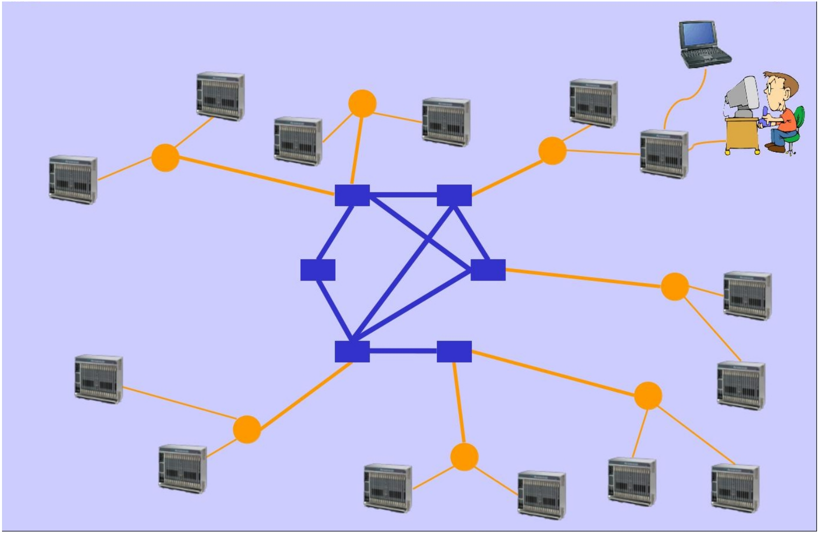
CO@Work

October 5, 2009

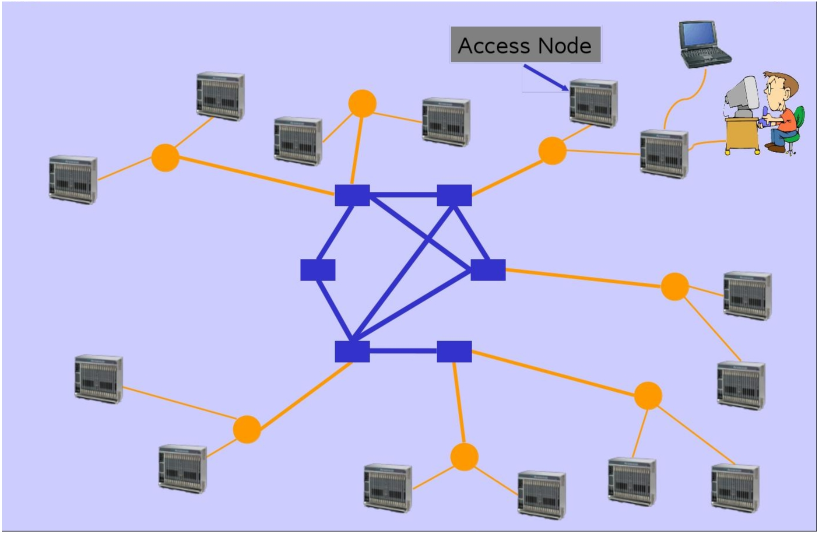
Maren Martens
Andreas Bley, Christian Raack

Contents

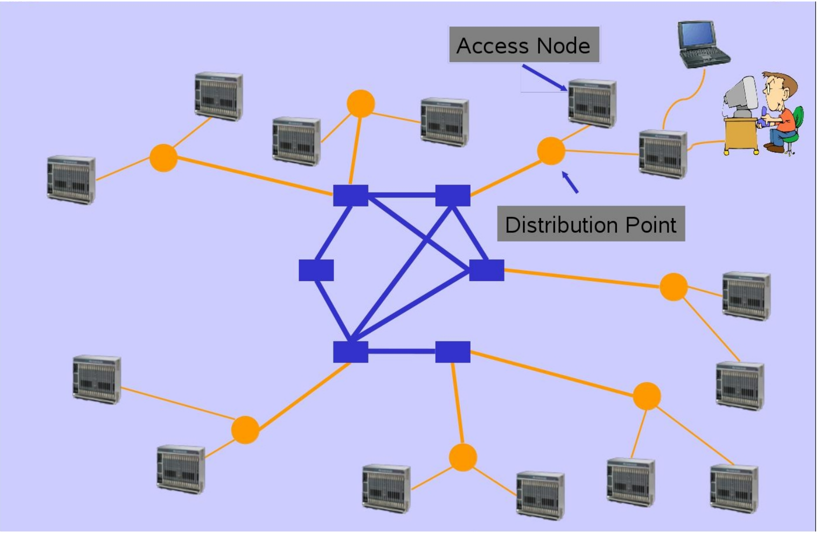
A Telecommunication Network



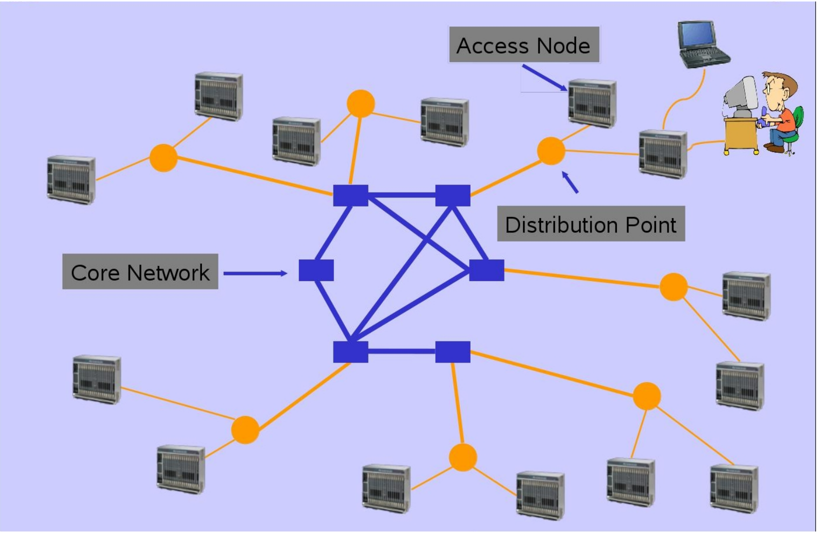
A Telecommunication Network



A Telecommunication Network



A Telecommunication Network



Planning Problems

Access Networks:

- How to connect customers to given routers?

Core Networks:

- Which links to use between routers?

Survivability:

- How to avoid interruptions?

Demands:

- Given customer demands, how to route them?

Access Networks

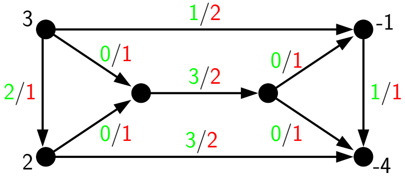
Access Networks

The Minimum Cost Flow Problem (MCFP)

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

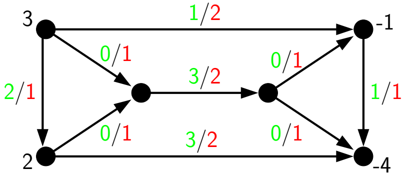
The Minimum Cost Flow Problem (MCFP)

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$



The Minimum Cost Flow Problem (MCFP)

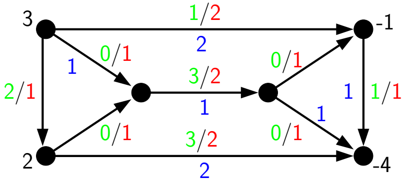
Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$



Find: Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands

The Minimum Cost Flow Problem (MCFP)

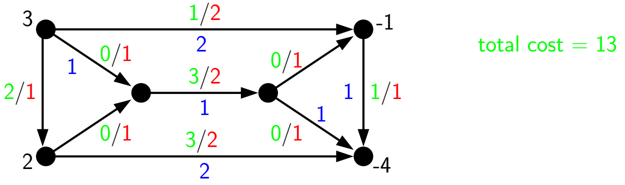
Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$



Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

The Minimum Cost Flow Problem (MCFP)

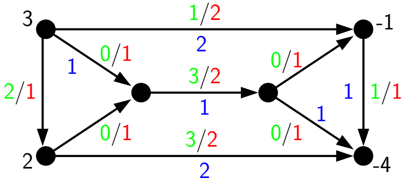
Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$



Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

The Minimum Cost Flow Problem (MCFP)

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$



total cost $= 13$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

$$\min \sum_{a \in A} c_a f(a)$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v \qquad \forall\, v \in V$$

$$0 \le f(a) \le u_a \qquad \forall\, a \in A$$

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

cost = length

capacities $= \infty$

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

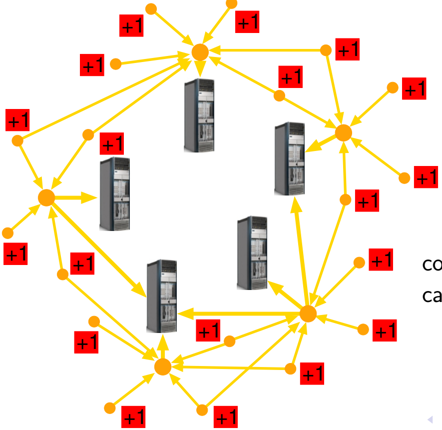
cost = length

capacities $= \infty$

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands



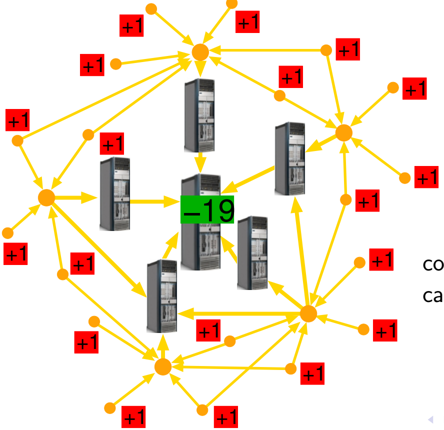
cost = length

capacities = ∞

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{Z}_0^+$ $(v \in V)$
with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands



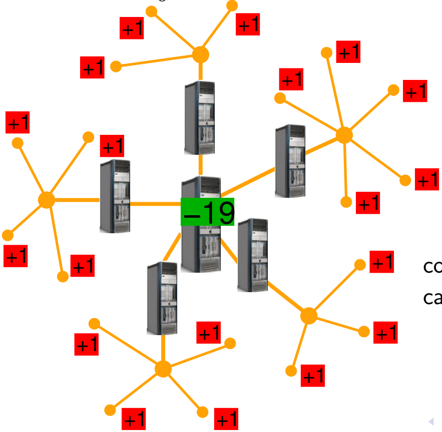
cost = length

capacities = ∞

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{Z}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

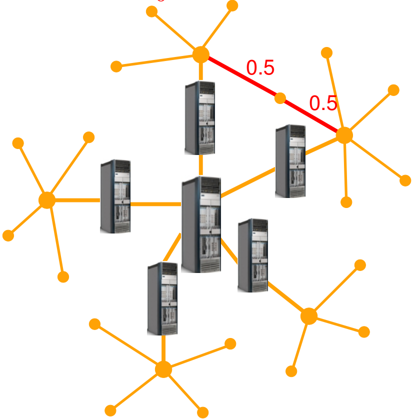


cost = length
capacities = ∞

Access Networks and the MCFP

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{Z}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost c_a and capacities u_a $(a \in A)$

Find: Flow $f : A \to \mathbb{Z}_0^+$ of minimum cost, satisfying all demands



$\to f_e$ integral!

Algorithms for MCFP

$$\begin{aligned} \min \quad & \sum_{a \in A} c_a f(a) & \\ \text{s.t.} \quad & \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v & \forall\, v \in V \\ & 0 \le f(a) \le u_a & \forall\, a \in A \end{aligned}$$

$\rightarrow$ Solvable in polynomial time for $f(a) \in \mathbb{R}_0^+$!

Algorithms for MCFP

$$
\begin{aligned}
\min \quad & \sum_{a \in A} c_a f(a) & \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v & \forall\, v \in V \\
& 0 \leq f(a) \leq u_a & \forall\, a \in A
\end{aligned}
$$

$\rightarrow$ Solvable in polynomial time for $f(a) \in \mathbb{R}_0^+$!

How do we quickly find integral solutions?

Algorithms for MCFP

$$\begin{aligned} \min \quad & \sum_{a \in A} c_a f(a) & \\ \text{s.t.} \quad & \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v & \forall\, v \in V \\ & 0 \leq f(a) \leq u_a & \forall\, a \in A \end{aligned}$$

$\rightarrow$ Solvable in polynomial time for $f(a) \in \mathbb{R}_0^+$!

How do we quickly find integral solutions?

Polynomial combinatorial algorithms:

- Minimum mean cycle cancelling algorithm
- Successive shortest path algorithm

Algorithms for MCFP

$$
\begin{aligned}
\min \quad & \sum_{a \in A} c_a f(a) & \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v & \forall\, v \in V \\
& 0 \le f(a) \le u_a & \forall\, a \in A
\end{aligned}
$$

$\rightarrow$ Solvable in polynomial time for $f(a) \in \mathbb{R}_0^+$!

How do we quickly find integral solutions?

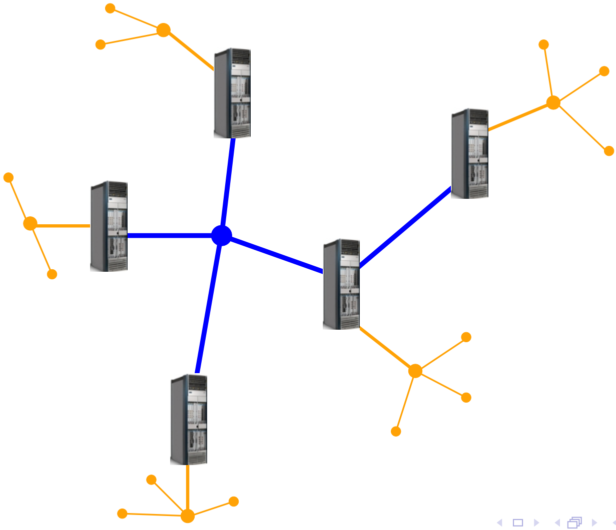
Polynomial combinatorial algorithms:

- Minimum mean cycle cancelling algorithm
- Successive shortest path algorithm

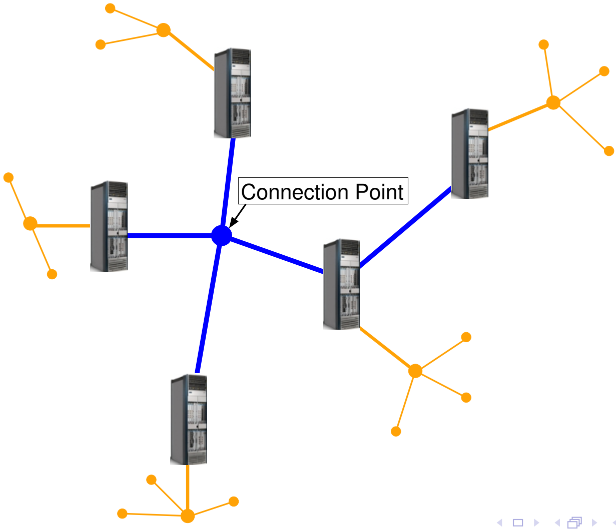
Both give integral optimal solution, if b and u are integral!

Core Networks

A Core Network



A Core Network



Steiner Trees

Definition

Given a graph $G = (V, E)$ with *terminals* $T \subseteq V$, a *Steiner tree* is a tree $S \subseteq E$ that connects all terminals in T.

Steiner Trees

Definition

Given a graph $G = (V, E)$ with *terminals* $T \subseteq V$, a *Steiner tree* is a tree $S \subseteq E$ that connects all terminals in T.

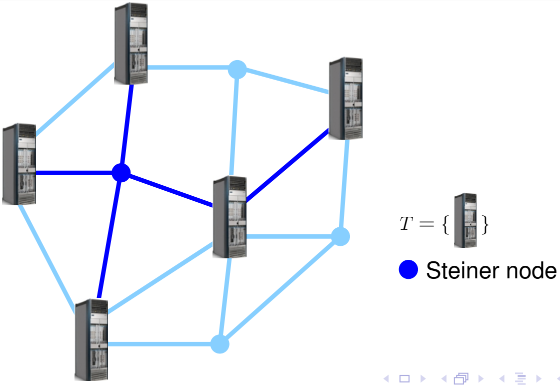
$T = \{$  $\}$

Steiner node

Steiner Trees

Definition

Given a graph $G = (V, E)$ with *terminals* $T \subseteq V$, a *Steiner tree* is a tree $S \subseteq E$ that connects all terminals in T.



The Steiner Tree Problem

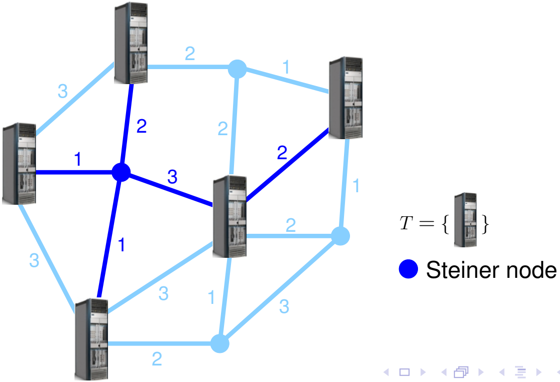
Definition

Given: $G = (V, E)$, terminals $T \subseteq V$, edge weights c_e $(e \in E)$
Find: Steiner tree $S \subseteq E$ of minimum weight $c(S) = \sum_{e \in S} c_e$

The Steiner Tree Problem

Definition

Given: $G = (V, E)$, terminals $T \subseteq V$, edge weights c_e $(e \in E)$
Find: Steiner tree $S \subseteq E$ of minimum weight $c(S) = \sum_{e \in S} c_e$



The Steiner Tree Problem

Definition

Given: $G = (V, E)$, terminals $T \subseteq V$, edge weights c_e $(e \in E)$
Find: Steiner tree $S \subseteq E$ of minimum weight $c(S) = \sum_{e \in S} c_e$

Theorem (Garey & Johnson, 1979)

The Steiner Tree Problem is NP-complete.

The Steiner Tree Problem

Definition

Given: $G = (V, E)$, terminals $T \subseteq V$, edge weights c_e $(e \in E)$
Find: Steiner tree $S \subseteq E$ of minimum weight $c(S) = \sum_{e \in S} c_e$

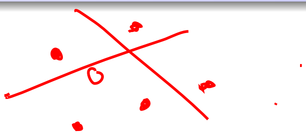
Theorem (Garey & Johnson, 1979)

The Steiner Tree Problem is NP-complete.

Simple special cases:

- $|T| = 2$: Shortest Path Problem
- $T = V$: Minimum Spanning Tree Problem

Formulation via Cuts (undirected)



Aneja 1980:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & x(\delta(U)) \geq 1 && \forall\, U \subset V:\ \emptyset \neq U \cap T \neq T \\ & x_e \in \mathbb{Z}_0^+ && \forall\, e \in E \end{aligned}$$

Formulation via Cuts (undirected)

Aneja 1980:

$$
\begin{array}{lll}
\min & c^T x & \\
\text{s.t.} & x(\delta(U)) \geq 1 & \forall\, U \subset V:\ \emptyset \neq U \cap T \neq T \\
& x_e \in \mathbb{Z}_0^+ & \forall\, e \in E
\end{array}
$$

- This formulation has $|E|$ variables and $O(2^{|V|})$ constraints.

Formulation via Cuts (undirected)

Aneja 1980:

$$
\begin{aligned}
\min\ & c^T x & & \\
\text{s.t. }\ & x(\delta(U)) \geq 1 & & \forall\, U \subset V:\ \emptyset \neq U \cap T \neq T \\
& x_e \in \mathbb{Z}_0^+ & & \forall\, e \in E
\end{aligned}
$$

- This formulation has $|E|$ variables and $O(2^{|V|})$ constraints.
- The separation problem is a minimum cut problem and can therefore be solved in polynomial time.

Formulation via Cuts (undirected)

Aneja 1980:

$$\begin{aligned} \min \quad & c^T x & \\ \text{s.t.} \quad & x(\delta(U)) \geq 1 & \forall\, U \subset V:\ \emptyset \neq U \cap T \neq T \\ & x_e \in \mathbb{Z}_0^+ & \forall\, e \in E \end{aligned}$$

- This formulation has $|E|$ variables and $O(2^{|V|})$ constraints.
- The separation problem is a minimum cut problem and can therefore be solved in polynomial time.

$\longrightarrow$ The LP relaxation can also be solved in polynomial time. (separation time $\approx$ optimization time, [GLS1988])

Formulation via Directed Cuts

Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights c_e:

- Build digraph $D = (V, A)$ with $A := \{(i, j), (j, i) | \{i, j\} \in E\}$.
- Choose arc weights $c_{(i,j)} := c_{\{i,j\}}$ for all $(i, j) \in A$.

Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights c_e:

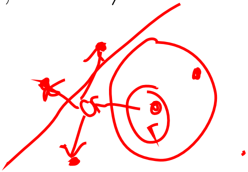
- Build digraph $D = (V, A)$ with $A := \{(i, j), (j, i) | \{i, j\} \in E\}$.
- Choose arc weights $c_{(i,j)} := c_{\{i,j\}}$ for all $(i, j) \in A$.
- Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$.

Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights c_e:

- Build digraph $D = (V, A)$ with $A := \{(i,j), (j,i) | \{i,j\} \in E\}$.
- Choose arc weights $c_{(i,j)} := c_{\{i,j\}}$ for all $(i,j) \in A$.
- Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$.

$$
\begin{array}{lll}
\min & c^T x & \\
\text{s.t.} & y(\delta^+(U)) \geq 1 & \forall\, U \subset V :\ r \in U, U \cap T' \neq T' \\
& y_{(i,j)} + y_{(j,i)} \leq x_{\{i,j\}} & \forall\, \{i,j\} \in E \\
& y_{(i,j)} \in \{0,1\} & \forall\, (i,j) \in A \\
& x_{\{i,j\}} \in \mathbb{Z}_0^+ & \forall\, \{i,j\} \in E
\end{array}
$$



Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights c_e:

- Build digraph $D = (V, A)$ with $A := \{(i,j),(j,i) | \{i,j\} \in E\}$.
- Choose arc weights $c_{(i,j)} := c_{\{i,j\}}$ for all $(i,j) \in A$.
- Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$.

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & y(\delta^+(U)) \geq 1 && \forall\, U \subset V : \; r \in U, U \cap T' \neq T' \\
& y_{(i,j)} + y_{(j,i)} \leq x_{\{i,j\}} && \forall\, \{i,j\} \in E \\
& y_{(i,j)} \geq 0 && \forall\, (i,j) \in A \\
& x_{\{i,j\}} \in \mathbb{Z}_0^+ && \forall\, \{i,j\} \in E
\end{aligned}
$$

Directed vs. Undirected

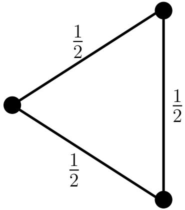
- The directed cut formulation reflects all feasible Steiner trees.

Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

Directed vs. Undirected

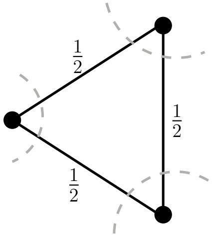
- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



$T = V$

Directed vs. Undirected

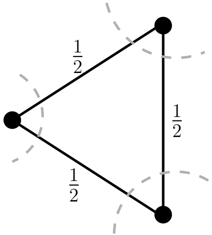
- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



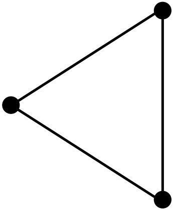
$T = V$

Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

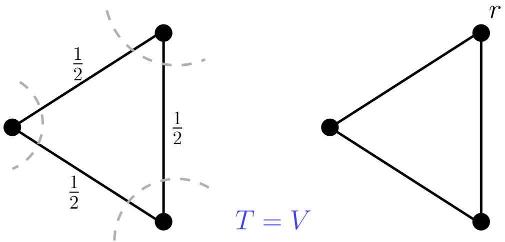


$T = V$



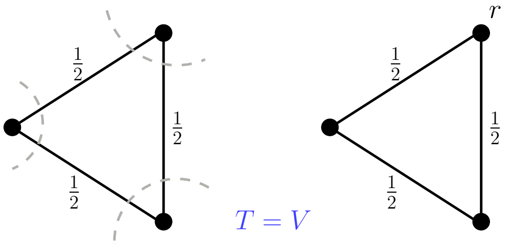
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



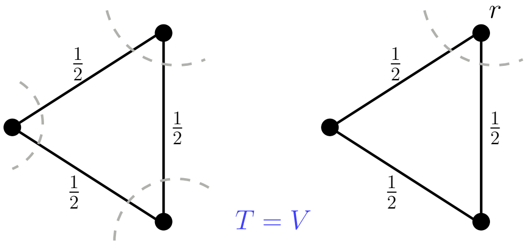
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



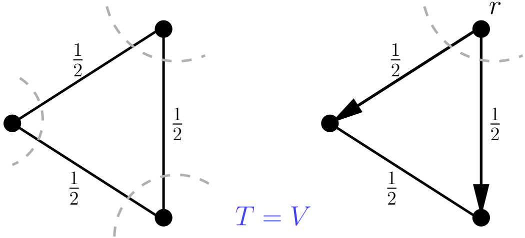
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



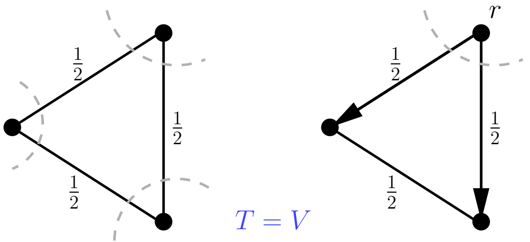
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.



Theorem

The directed cut formulation is stronger than the undirected cut formulation.

Formulation via Flows

Problem with cuts: There are too many!

Formulation via Flows

Problem with cuts: There are too many!

On digraph $D = (V, A)$ with root $r \in T$ and $T' := T \setminus \{r\}$:

- $\forall t \in T', a \in A$: flow variables $f_t(a) = \begin{cases} 1, & a \text{ on } r\text{-}t\text{-path} \\ 0, & \text{otherwise} \end{cases}$

Formulation via Flows

Problem with cuts: There are too many!

On digraph $D = (V, A)$ with root $r \in T$ and $T' := T \setminus \{r\}$:

- $\forall t \in T', a \in A$: flow variables $f_t(a) = \begin{cases} 1, & a \text{ on } r\text{-}t\text{-path} \\ 0, & \text{otherwise} \end{cases}$

$$\min \; c^T x$$

$$\begin{array}{lll} \text{s.t.} & \displaystyle\sum_{a \in \delta^+(v)} f_t(a) - \sum_{a \in \delta^-(v)} f_t(a) = \begin{cases} 1, & v = r \\ 0, & \text{else} \end{cases} & \forall\, t \in T', v \neq t \\ & f_s(i,j) + f_t(j,i) \leq x_{\{i,j\}} & \forall\, s,t \in T', \{i,j\} \in E \\ & f_t(a) \in \{0,1\} & \forall\, t \in T', a \in A \\ & x_e \in \mathbb{Z}_0^+ & \forall\, e \in E \end{array}$$

Formulation via Flows

Problem with cuts: There are too many!

On digraph $D = (V, A)$ with root $r \in T$ and $T' := T \setminus \{r\}$:

- $\forall t \in T', a \in A$: flow variables $f_t(a) = \begin{cases} 1, & a \text{ on } r\text{-}t\text{-path} \\ 0, & \text{otherwise} \end{cases}$

$$\min \ c^T x$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f_t(a) - \sum_{a \in \delta^-(v)} f_t(a) = \begin{cases} 1, & v = r \\ 0, & \text{else} \end{cases} \qquad \forall\, t \in T', v \neq t$$

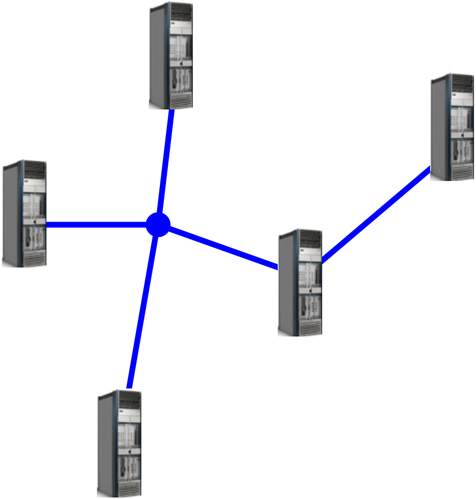
$$f_s(i,j) + f_t(j,i) \leq x_{\{i,j\}} \qquad \forall\, s,t \in T', \{i,j\} \in E$$

$$f_t(a) \geq 0 \qquad \forall\, t \in T', a \in A$$

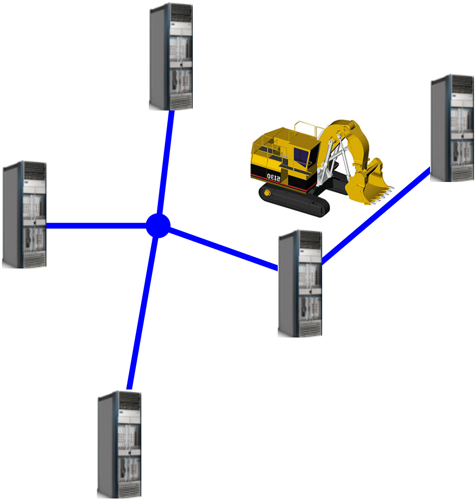
$$x_e \in \mathbb{Z}_0^+ \qquad \forall\, e \in E$$

Survivability

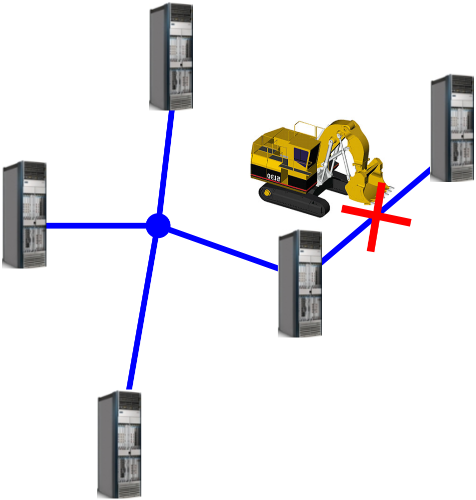
Failures happen...



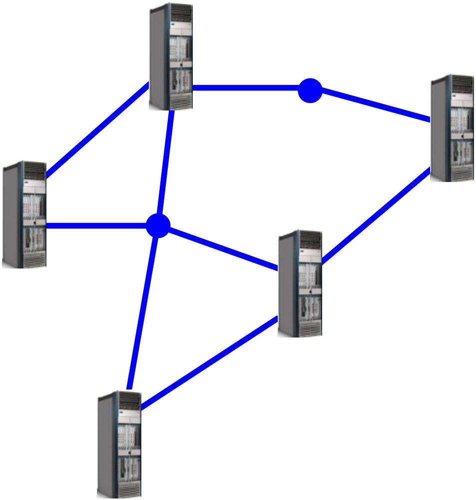
Failures happen...



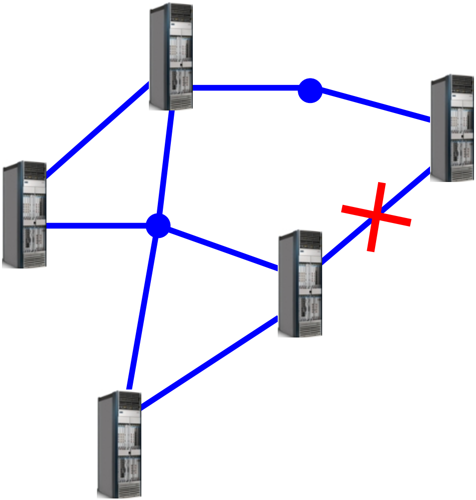
Failures happen...



Failures happen...



Failures happen...



2-Connectivity

Given: $G = (V, E)$, edge cost c_e $(e \in E)$

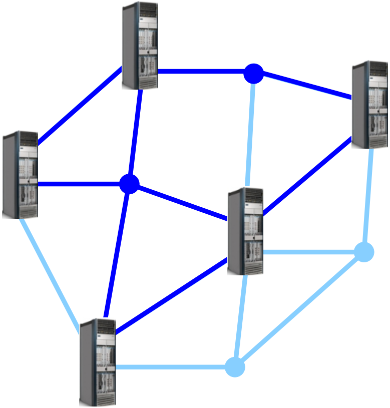
Find: Minimum cost subgraph containing all $v \in V$ and having 2 node/edge-disjoint s-t-paths for all $s, t \in V$

2-Connectivity

Given: $G = (V, E)$, edge cost c_e $(e \in E)$
Find: Minimum cost subgraph containing all $v \in V$ and having 2 node/edge-disjoint s-t-paths for all $s, t \in V$

2-Connectivity

Given: $G = (V, E)$, edge cost c_e $(e \in E)$
Find: Minimum cost subgraph containing all $v \in V$ and having 2 node/edge-disjoint s-t-paths for all $s, t \in V$



{0,1,2}-Survivable Network Design ({0,1,2}-SND)

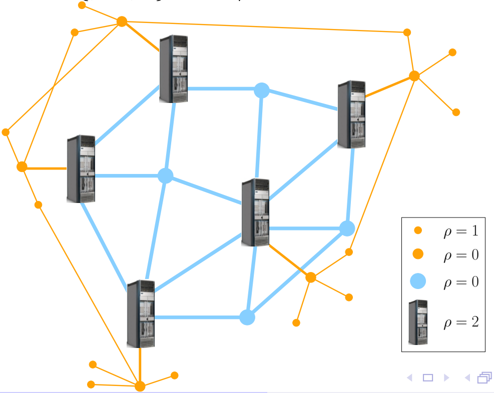
Given: $G = (V, E)$, connectivity request $\rho_v \in \{0, 1, 2\}$ $(v \in V)$, edge cost c_e $(e \in E)$

Find: Minimum cost subgraph containing all $v \in V$ with $\rho_v > 0$, having $\min\{\rho_s, \rho_t\}$ node/edge-disjoint s-t-paths $(s, t \in V)$

{0,1,2}-Survivable Network Design ({0,1,2}-SND)

Given: $G = (V, E)$, connectivity request $\rho_v \in \{0, 1, 2\}$ $(v \in V)$, edge cost c_e $(e \in E)$

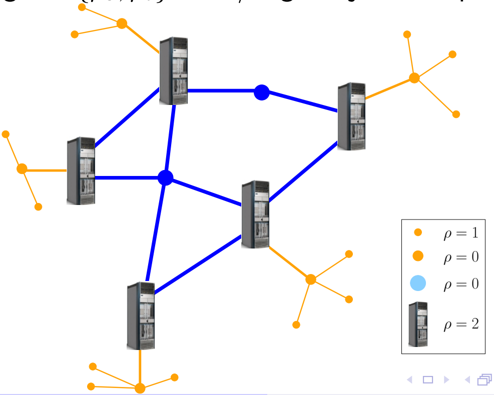
Find: Minimum cost subgraph containing all $v \in V$ with $\rho_v > 0$, having $\min\{\rho_s, \rho_t\}$ node/edge-disjoint s-t-paths $(s, t \in V)$



{0,1,2}-Survivable Network Design ({0,1,2}-SND)

Given: $G = (V, E)$, connectivity request $\rho_v \in \{0, 1, 2\}$ $(v \in V)$, edge cost c_e $(e \in E)$

Find: Minimum cost subgraph containing all $v \in V$ with $\rho_v > 0$, having $\min\{\rho_s, \rho_t\}$ node/edge-disjoint s-t-paths $(s, t \in V)$



Algorithms for {0,1,2}-SND

Theorem

{0,1,2}-SND is NP-hard.

Algorithms for {0,1,2}-SND

Theorem

{0,1,2}-SND is NP-hard.

IP formulations:

- via undirected cuts (Grötschel, Monma, Stoer 1989)
- via directed cuts (Chimani, Kandyba, Ljubic, Mutzel 2007)

Algorithms for {0,1,2}-SND

Theorem

{0,1,2}-SND is NP-hard.

IP formulations:

- via undirected cuts (Grötschel, Monma, Stoer 1989)
- via directed cuts (Chimani, Kandyba, Ljubic, Mutzel 2007)

Algorithms approximating the minimum cost:

- Factor 2-approximation for edge-connectivity (Jain 1998)
- Node-connectivity is NP-hard to approximate within factor $2^{\log^{1-\epsilon}|V|}$ (Kortsarz, Krauthgamer, Lee 2003)

Algorithms for {0,1,2}-SND

Theorem

{0,1,2}-SND is NP-hard.

IP formulations:

- via undirected cuts (Grötschel, Monma, Stoer 1989)
- via directed cuts (Chimani, Kandyba, Ljubic, Mutzel 2007)

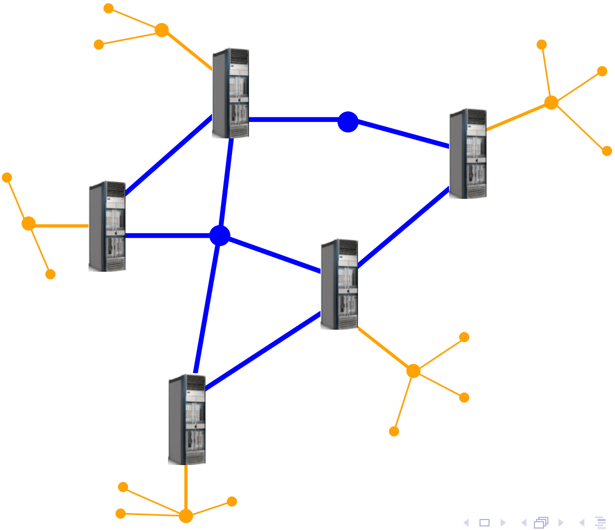
Algorithms approximating the minimum cost:

- Factor 2-approximation for edge-connectivity (Jain 1998)
- Node-connectivity is NP-hard to approximate within factor $2^{\log^{1-\epsilon}|V|}$ (Kortsarz, Krauthgamer, Lee 2003)*

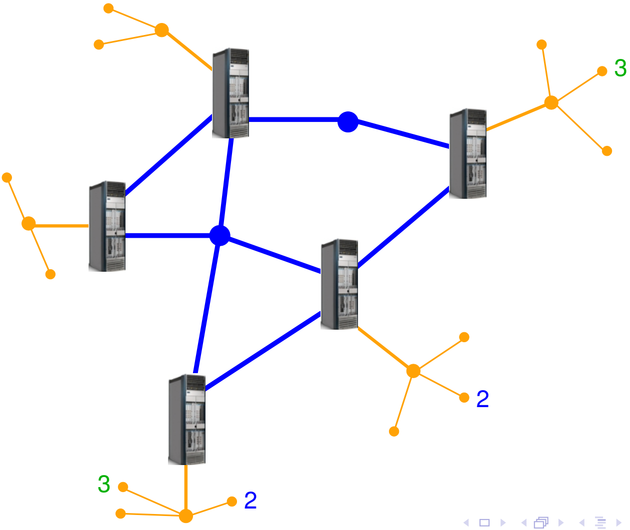
* 2-approximation for 2-node-connected subgraph (Khuller 1997)

Demands and Capacities

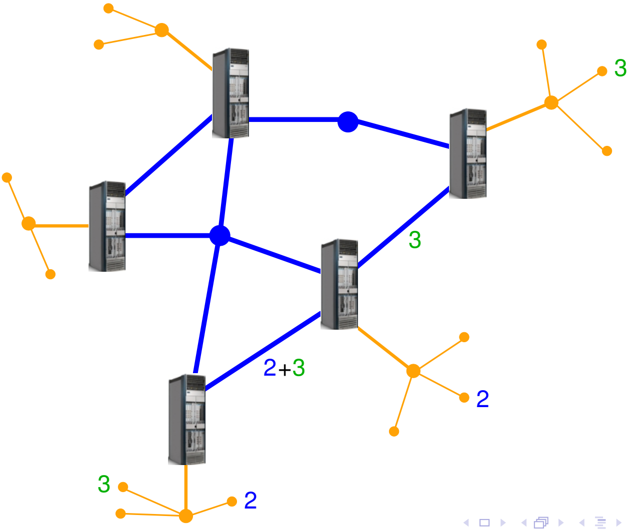
Are we done yet?



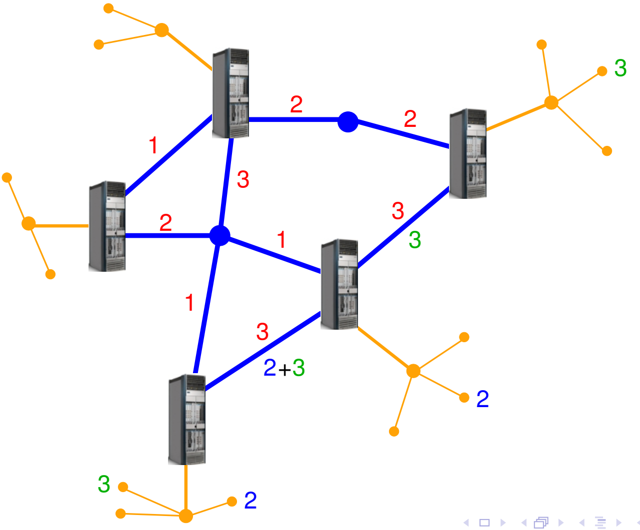
Are we done yet?



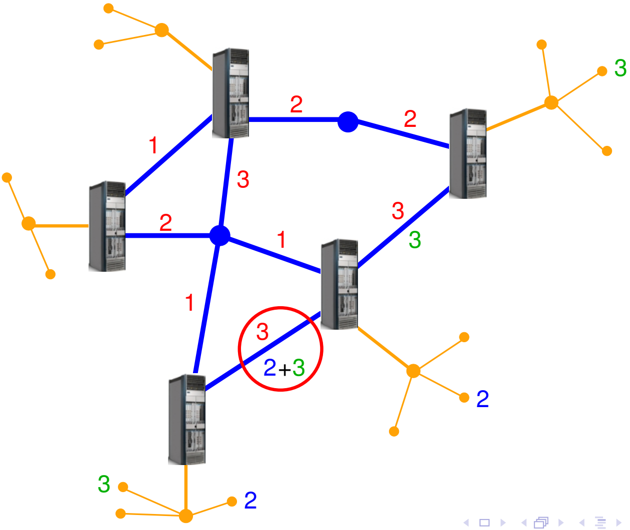
Are we done yet?



Are we done yet?



Are we done yet?



Connectivity and Flows

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Connectivity and Flows

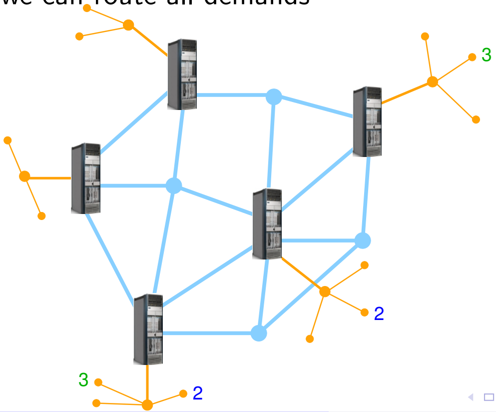
Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Find: Minimum cost subgraph with capacity installations such that we can route all demands

Connectivity and Flows

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

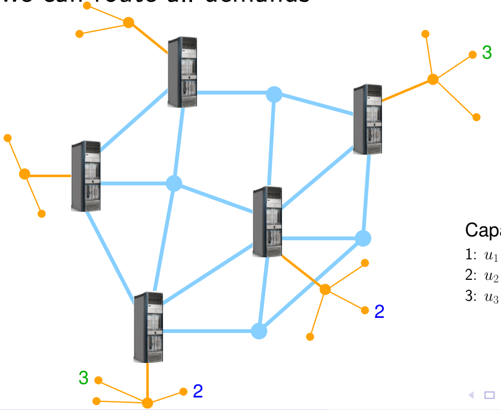
Find: Minimum cost subgraph with capacity installations such that we can route all demands



Connectivity and Flows

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Find: Minimum cost subgraph with capacity installations such that we can route all demands



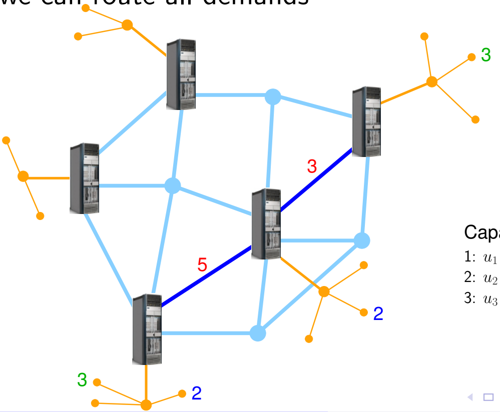
Capacity modules:

1: $u_1 = 2,\ c_1 = 20$
2: $u_2 = 3,\ c_2 = 25$
3: $u_3 = 5,\ c_3 = 30$

Connectivity and Flows

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Find: Minimum cost subgraph with capacity installations such that we can route all demands



Capacity modules:

1: $u_1 = 2,\ c_1 = 20$

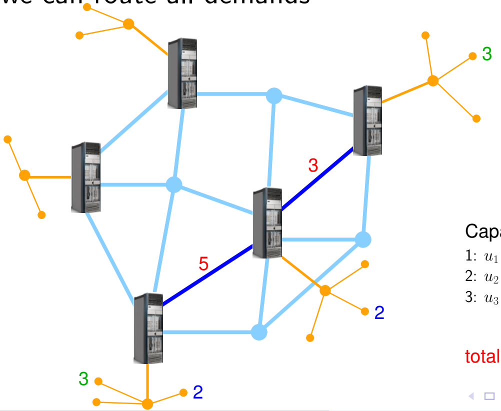
2: $u_2 = 3,\ c_2 = 25$

3: $u_3 = 5,\ c_3 = 30$

Connectivity and Flows

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Find: Minimum cost subgraph with capacity installations such that we can route all demands



Capacity modules:

1: $u_1 = 2$, $c_1 = 20$
2: $u_2 = 3$, $c_2 = 25$
3: $u_3 = 5$, $c_3 = 30$

total cost for core: 55

IP Formulation (Connectivity and Flows)

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

IP Formulation (Connectivity and Flows)

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i, j), (j, i) | \{i, j\} \in E\}$

Capacity decisions for edges: $y_k^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

IP Formulation (Connectivity and Flows)

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st}, capacity modules $k \in K$ with capacities u_k, cost c_k

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i, j), (j, i) | \{i, j\} \in E\}$
Capacity decisions for edges: $y_k^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

$$\min \sum_{e \in E} \sum_{k \in K} c_k y_k^e$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f_{st}(a) - \sum_{a \in \delta^-(v)} f_{st}(a) = \begin{cases} \overset{d_{st}}{\cancel{1}}, & v = s \\ 0, & \text{else} \end{cases} \qquad \forall\ (s,t), v \neq t$$

$$\sum_{(s,t)} \cancel{d_{st}}(f(a_e) + f(\overleftarrow{a_e})) \leq \sum_{k \in K} u_k y_k^e \qquad \forall\ e \in E$$

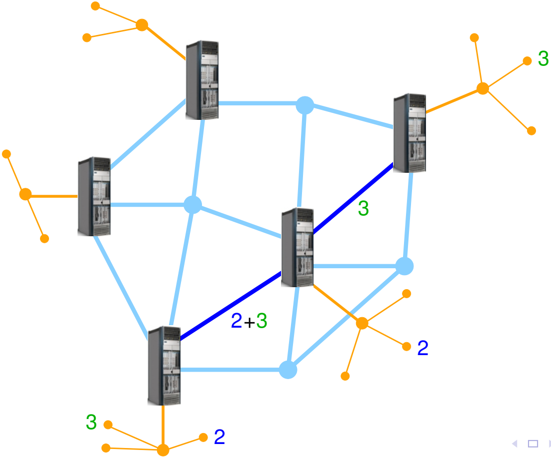
[for $e = \{i, j\} \in E$: directed arcs $a_e = (i, j)$ and $\overleftarrow{a_e} = (j, i)$]

How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.

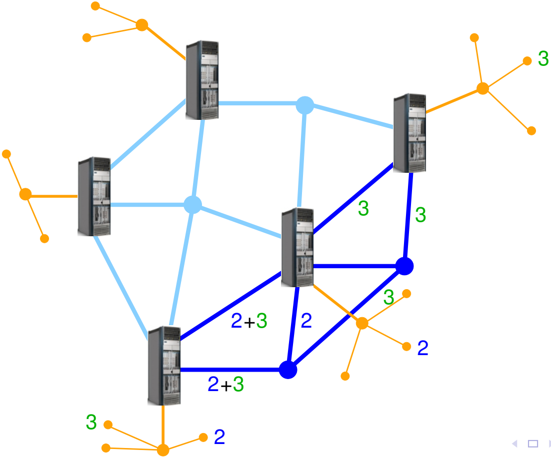
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.



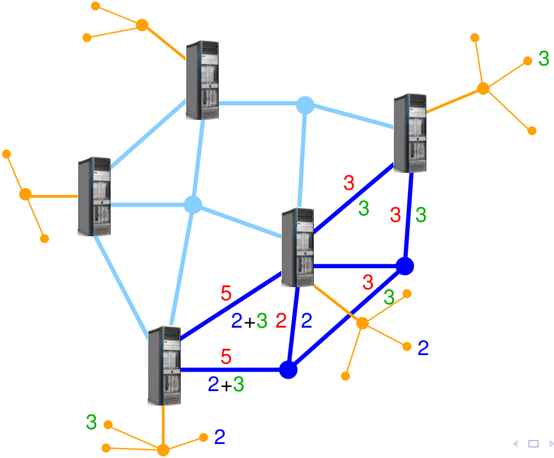
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.



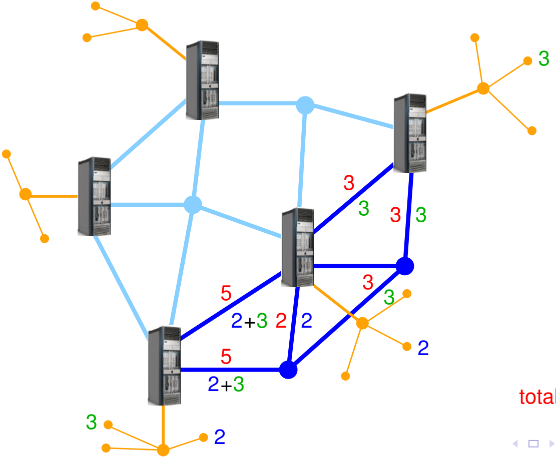
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.



How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.



total cost for core: 155

IP Formulation ((1+1)-Protection)

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st},
capacity modules $k \in K$ with capacities u_k, cost c_k

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i, j), (j, i) | \{i, j\} \in E\}$

Capacity decisions for edges: $y_k^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

IP Formulation ((1+1)-Protection)

Given: $G = (V, E)$, commodities $(s,t) \in V \times V$ with demands d_{st},
capacity modules $k \in K$ with capacities u_k, cost c_k

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i,j),(j,i) | \{i,j\} \in E\}$

Capacity decisions for edges: $y_k^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

$$\min \sum_{e \in E} \sum_{k \in K} c_k y_k^e$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f_{st}(a) - \sum_{a \in \delta^-(v)} f_{st}(a) = \begin{cases} 2, & v = s \\ 0, & \text{else} \end{cases} \qquad \forall\ (s,t), v \neq t$$

$$\sum_{(s,t)} d_{st}(f(a_e) + f(\overleftarrow{a_e})) \leq \sum_{k \in K} u_k y_k^e \qquad \forall\ e \in E$$

$$\sum_{a \in \delta^-(v)} f_{st}(a) \leq 1 \qquad \forall\ (s,t), v \neq t^*$$

IP Formulation ((1+1)-Protection)

Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands d_{st},
capacity modules $k \in K$ with capacities u_k, cost c_k

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i,j),(j,i) | \{i,j\} \in E\}$

Capacity decisions for edges: $y_k^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

$$\min \sum_{e \in E} \sum_{k \in K} c_k y_k^e$$

s.t.

$$\sum_{a \in \delta^+(v)} f_{st}(a) - \sum_{a \in \delta^-(v)} f_{st}(a) = \begin{cases} 2, & v = s \\ 0, & \text{else} \end{cases} \qquad \forall\ (s,t), v \neq t$$

$$\sum_{(s,t)} d_{st}(f(a_e) + f(\overleftarrow{a_e})) \leq \sum_{k \in K} u_k y_k^e \qquad \forall\ e \in E$$

$$\sum_{a \in \delta^-(v)} f_{st}(a) \leq 1 \qquad \forall\ (s,t), v \neq t^*$$

* for node-disjointness

The End

To be continued with Exercises at 2PM!

The End

To be continued with Exercises at 2PM!
Enjoy lunch!