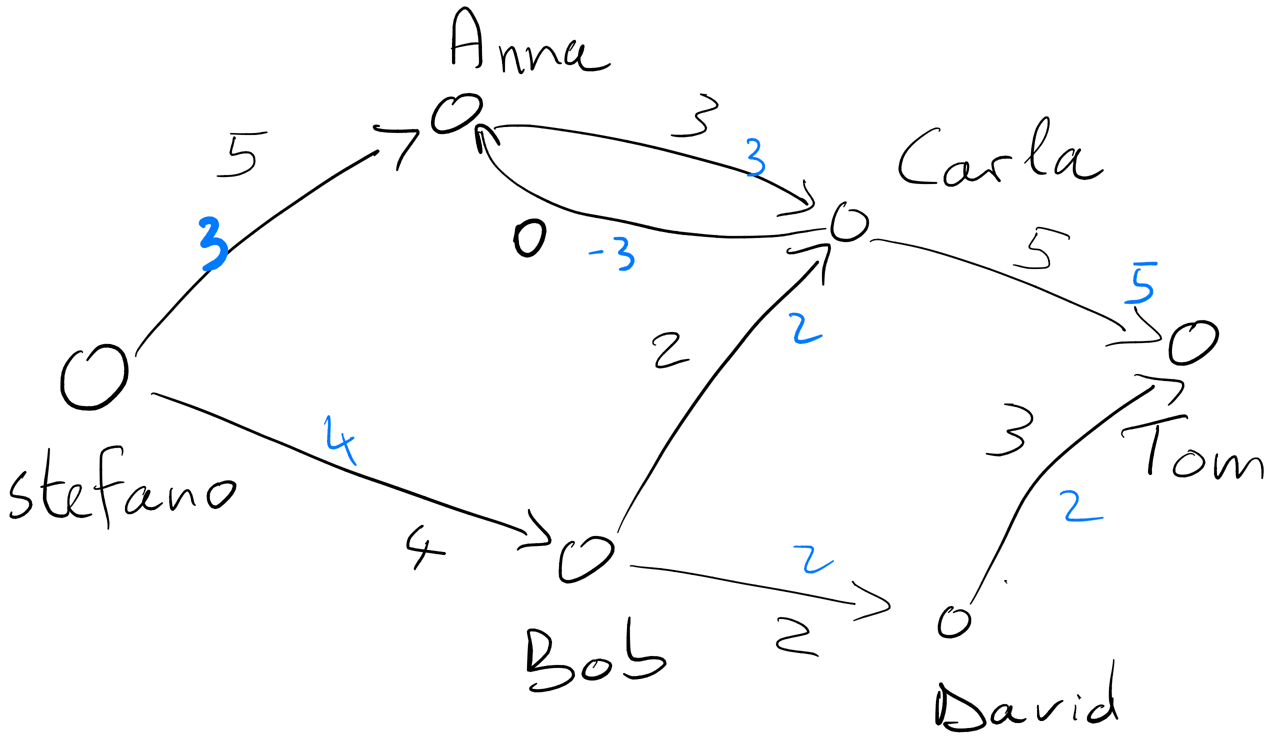
CS II Lecture 7

Flows in Networks

Peer-to-peer Lending



Flow

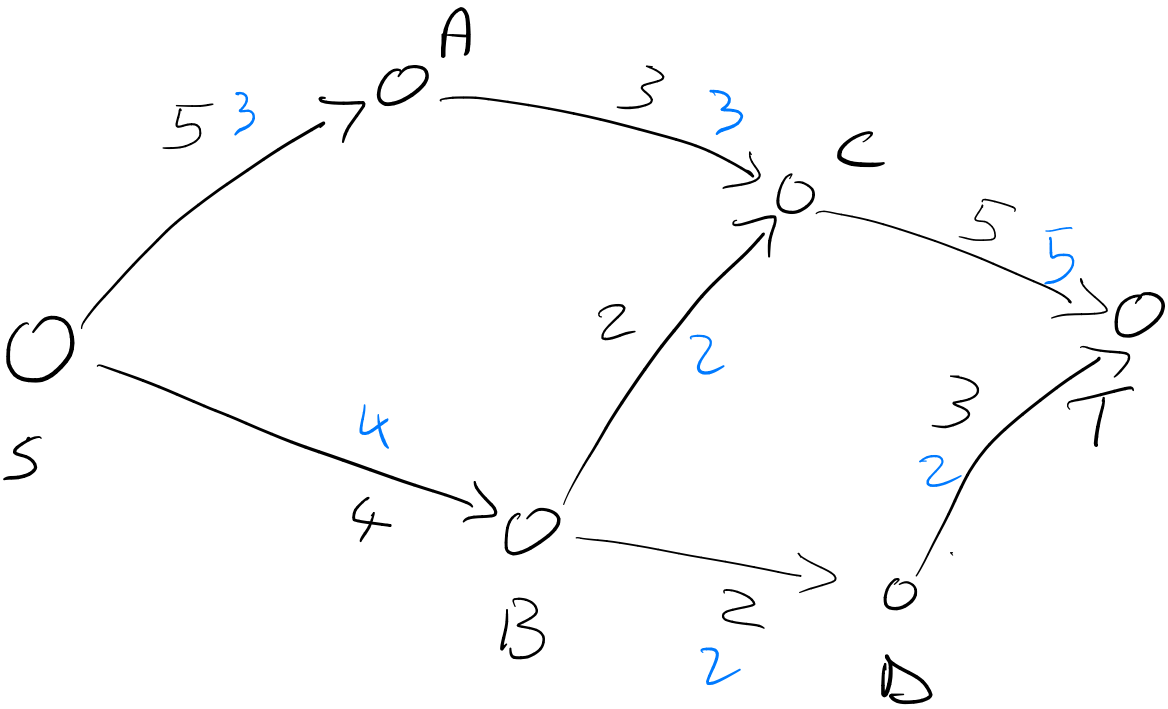
Assignment $f_{u,v}$ to each pairs of vertices such that (u,v) and/or (v,u) is an edge, such that

$$\boxed{\begin{array}{l} - \ f_{u,v} = -f_{v,u} \\ - \ \sum_v f_{u,v} = 0 \quad \text{for all } u \neq s,t \\ \qquad \text{conservation constraints} \\ - \ f_{u,v} \leq c_{u,v} \quad \text{capacity constraints} \end{array}}$$

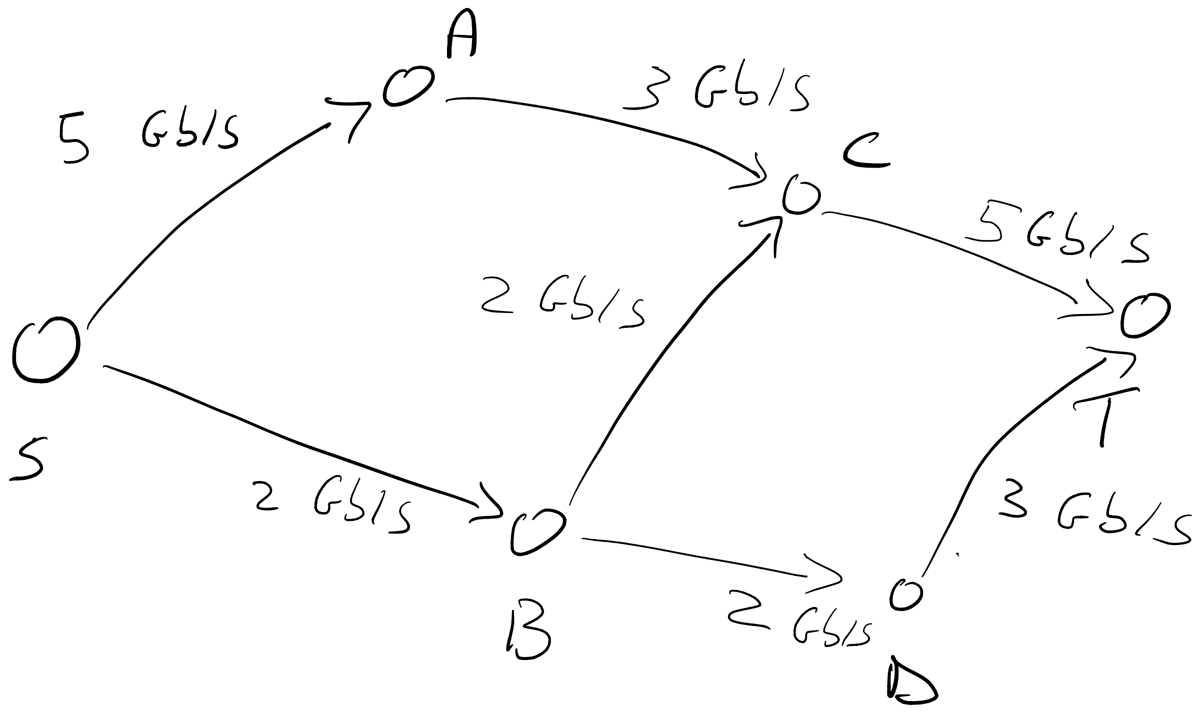
Value of flow

$$val(f) = \sum_v f_{s,v} = \sum_w f_{w,t}$$

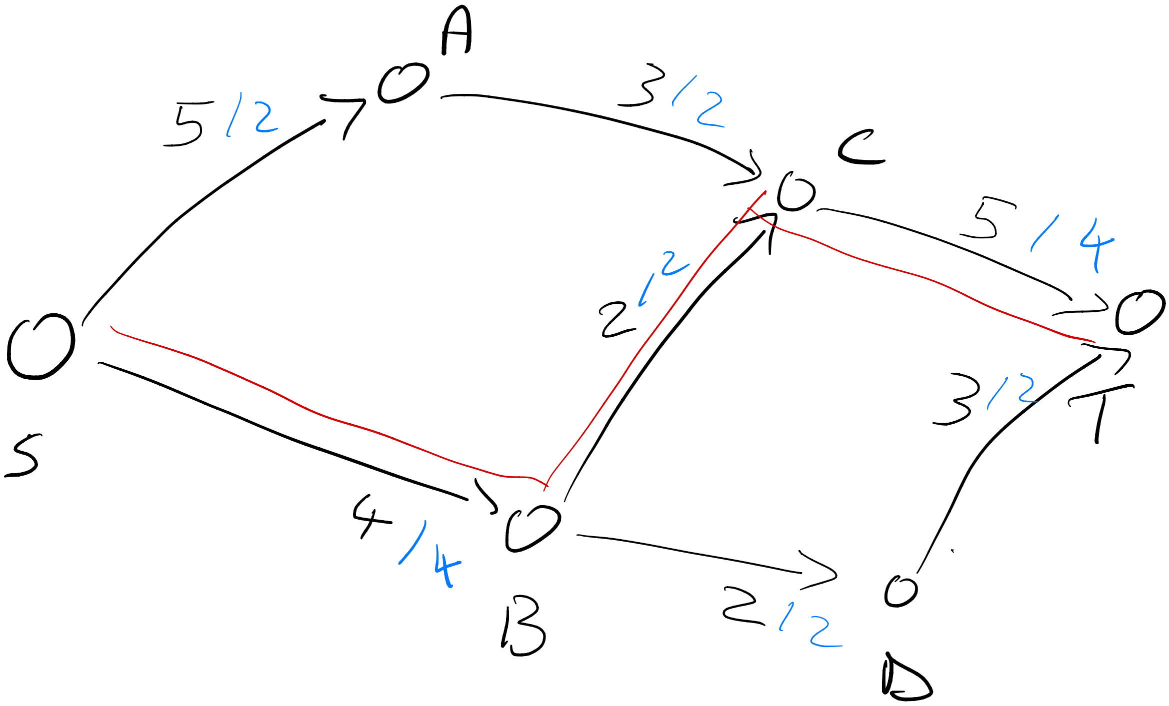
Logistics



Packet Routing in Comm. Network



Augmenting Path



If we find path

$$v_0 = s \to v_1 \to v_2 \to \dots \to v_k = t$$

such that all edges (v_i, v_{i+1}) have

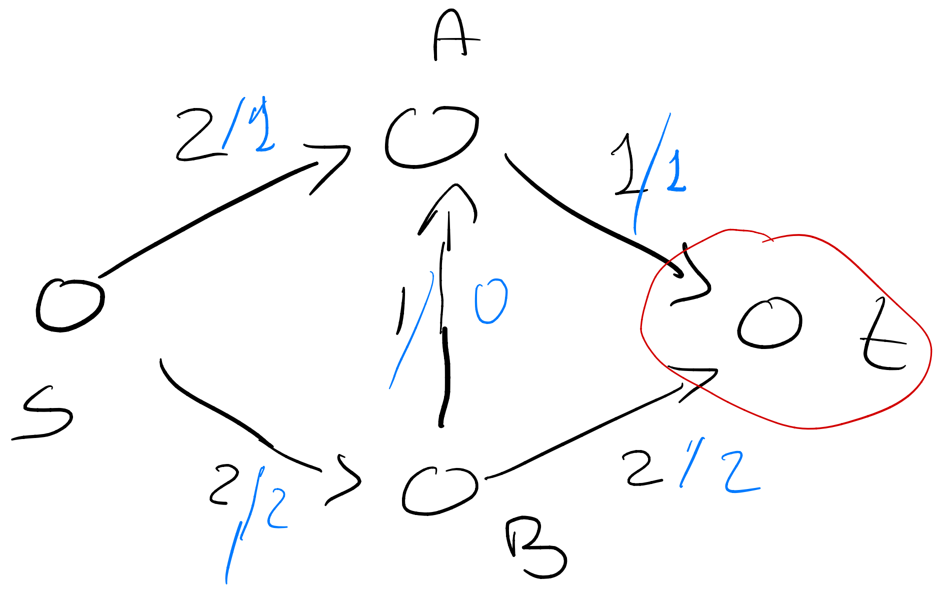
$$f_{v_i, v_{i+1}} < c_{v_i, v_{i+1}}$$

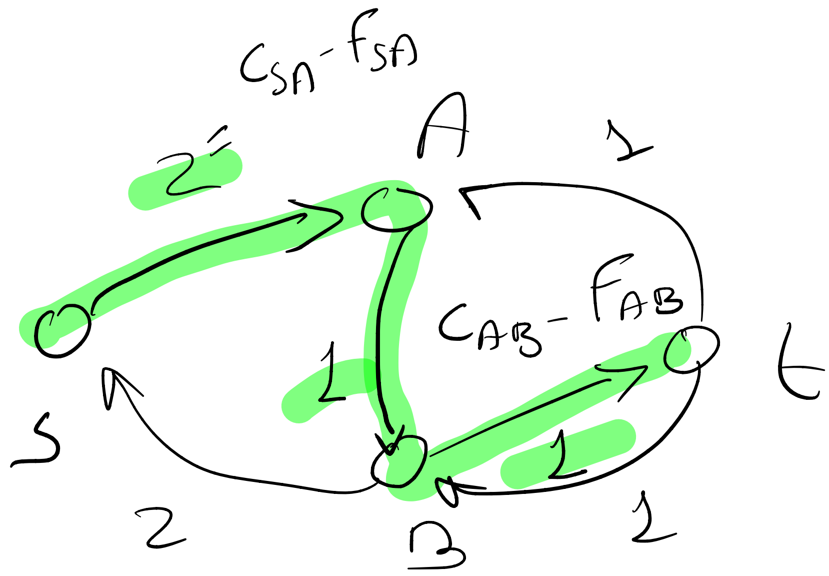
call $r = \min_{i=0,\dots,k-1} -f_{v_i, v_{i+1}} + c_{v_i, v_{i+1}}$

increase flow by r along path

Ford-Fulkerson Algorithm

1st attempt

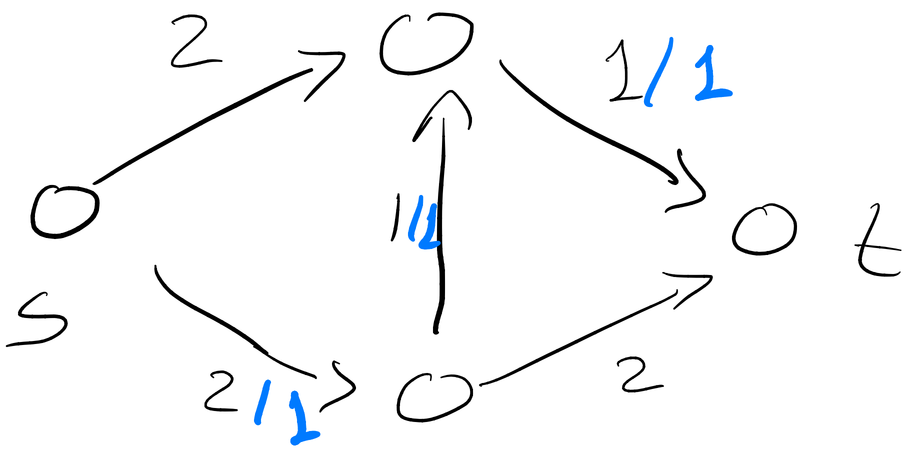


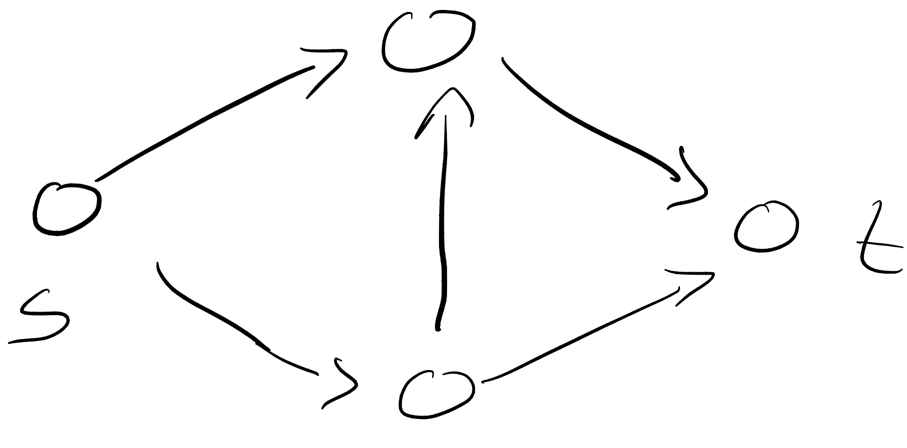


$c_{AB}=0$

$f_{AB}=-1$

Residual network





F. F. Algorithm

Input: Network $N = G, s, t, c$
where $G = (V, E)$

For each $(u,v) \in E$:

$f_{u,v} = f_{v,u} = 0$

$r_{u,v} = c_{u,v}$

while there is an $s \to t$ path P
in which all edges have $r_{u,v} > 0$:

$r = \min_{(u,v) \in P} r_{u,v}$

For each $(u,v) \in P$:

$f_{u,v} = f_{u,v} + r$

$f_{v,u} = f_{v,u} - r$

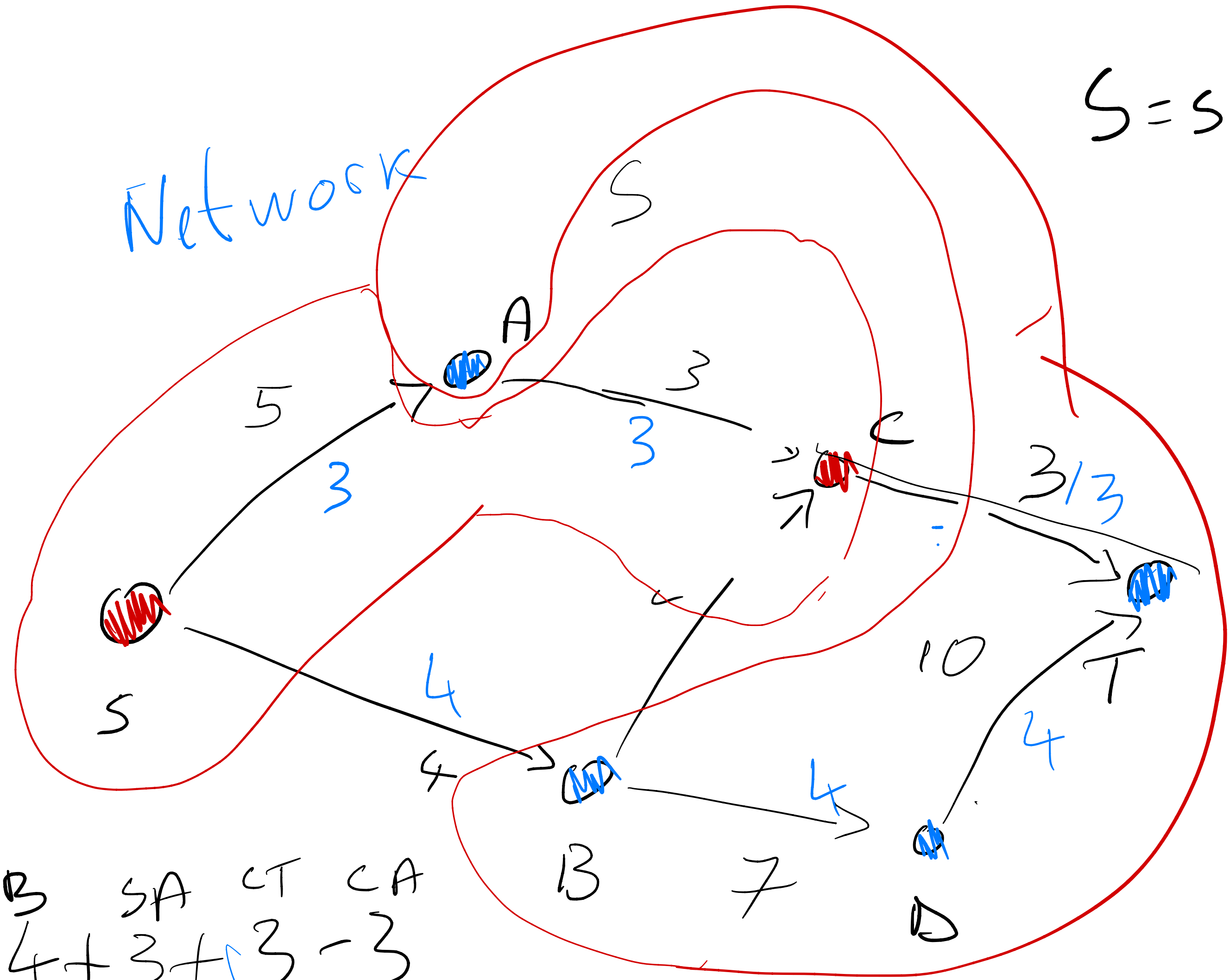
$r_{u,v} = c_{u,v} - f_{u,v}$

$r_{v,u} = c_{v,u} - f_{v,u}$

return f

$S = s, c$

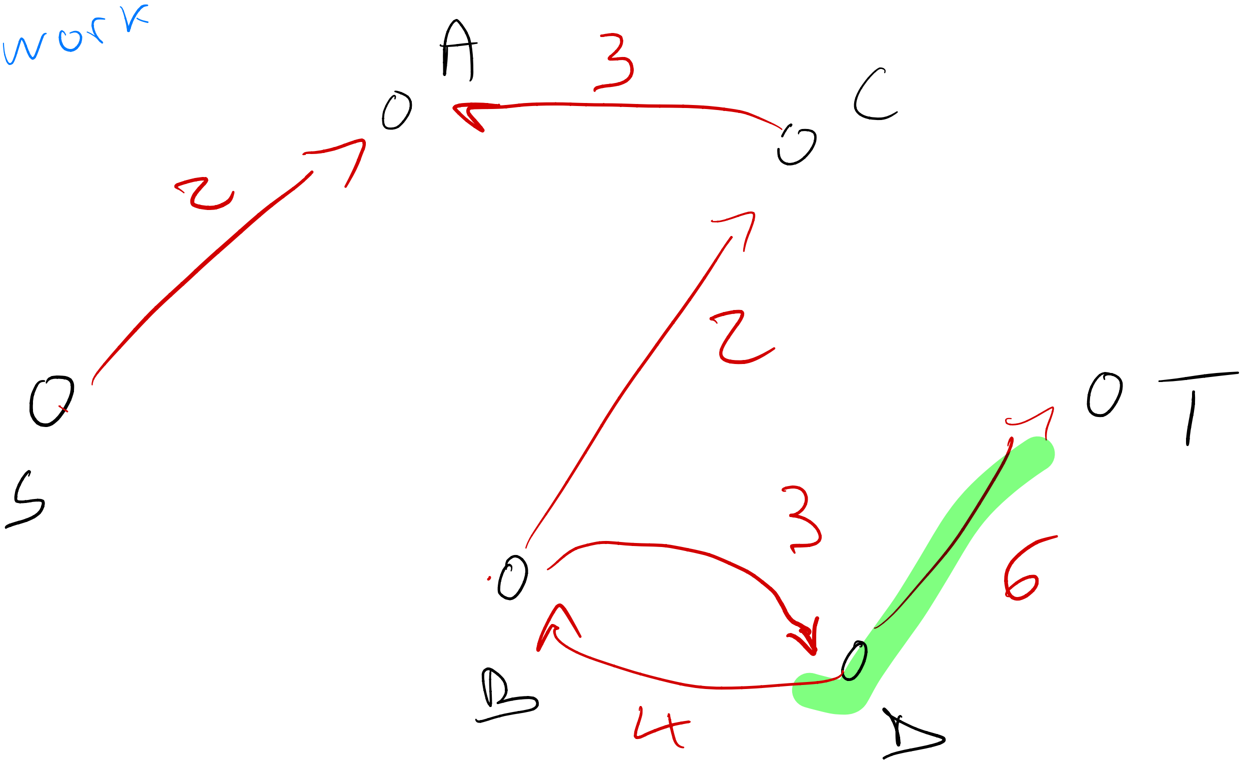
Network



SB SA CT CA

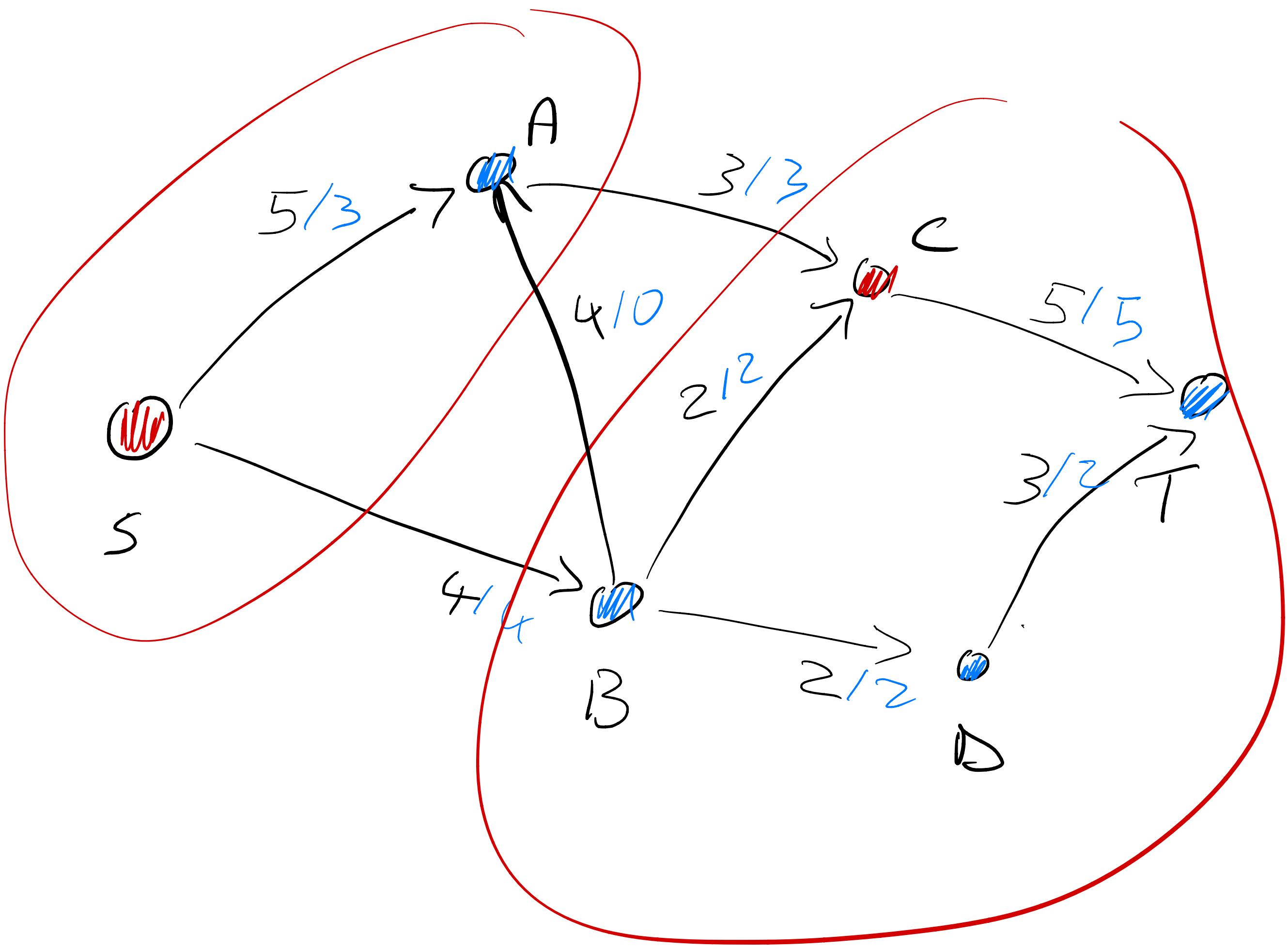
$4 + 3 + 3 - 3$

Residual network



Analysis of F.F.

We need to prove that when algorithm stops, flow is optimal.

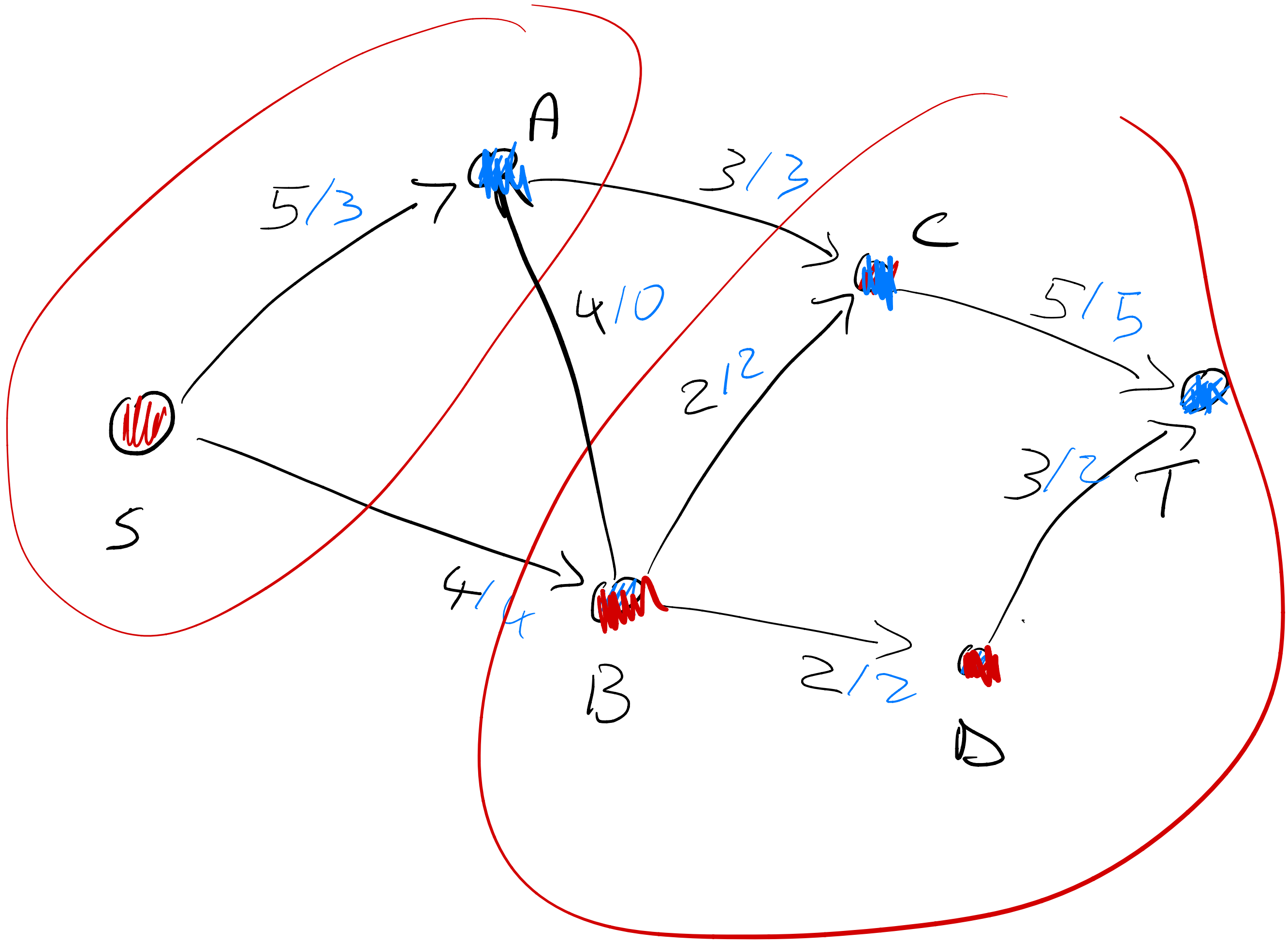


$$S = \{s, c\}$$

$$f_{SA} + f_{SB} + f_{CA} + f_{CT} + f_{CB}$$

$$3 + 4 - 3 + 5 - 2$$

$$= 7$$



4

$$f_{SA} + f_{BA} + f_{BC} + f_{DT}$$

3 0 2 2

Def: a "cut" in a network is a set $S \subseteq V$ such that

- $s \in S$
- $t \notin S$

Def: the "capacity" of a cut S is

$$cap(S) := \sum_{\substack{u \in S \\ v \notin S}} c_{u,v}$$

Lemma: if f is a flow and S is a cut

$$val(f) \leq cap(S)$$

Thm (Max Flow Min Cut Theorem) when FF Algorithm terminates, if f is flow at end of algorithm and S is set of vertices reachable from s in residual network:

$$val(f) = cap(S)$$

So f is maximum flow, and S is a minimum cut

Lemma: if f is a flow and S is a cut

$$\mathrm{val}(f) \le \mathrm{cap}(S)$$

We prove stronger fact

$$\mathrm{val}(f) = \sum_{\substack{u \in S \\ v \notin S}} f_{u,v}$$

$$\le \sum_{\substack{u \in S \\ v \notin S}} c_{u,v}$$

$$= \mathrm{cap}(S)$$

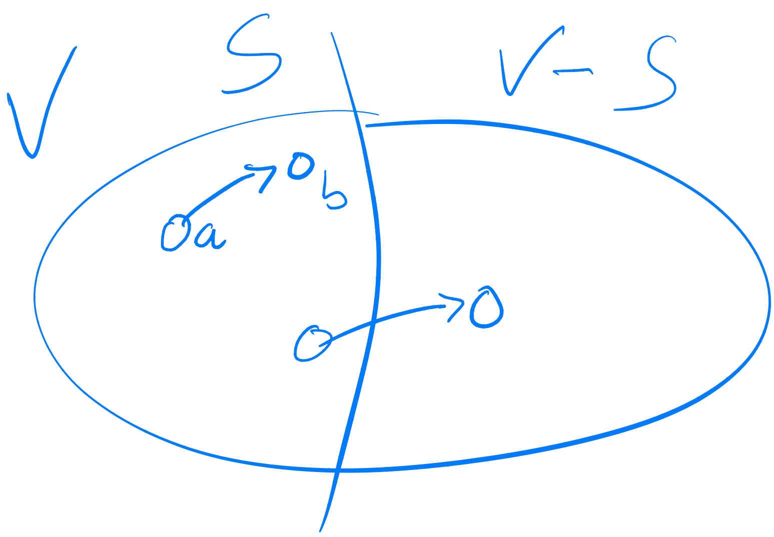
$$\mathrm{val}(f) \stackrel{\text{def}}{=} \sum_{v} f_{s,v}$$

$$= \sum_{v} f_{s,v} + \sum_{\substack{u \in S \\ u \neq s}} \sum_{v} f_{u,v}$$

[justified by $\sum_{v} f_{u,v} = 0$ if $u \neq s, u \neq t$]

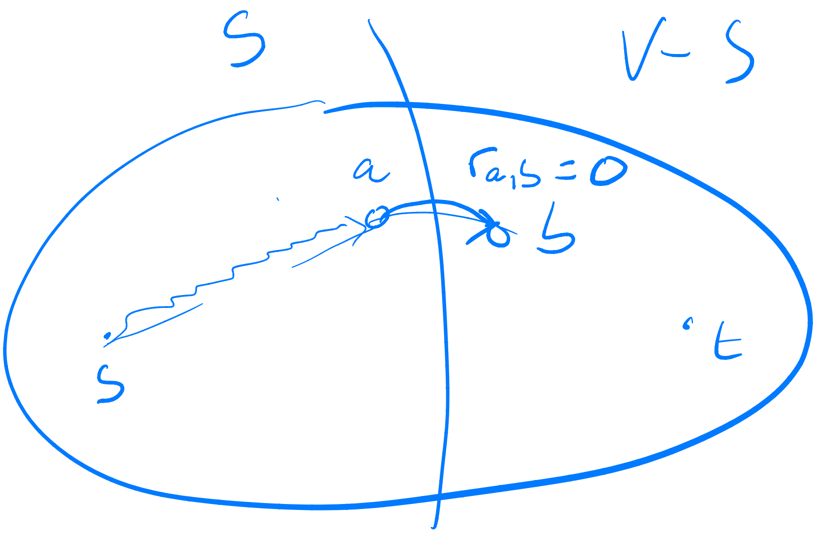
$$= \underbrace{\sum_{u \in S} \sum_{v \in V} f_{u,v}}$$

$$= \sum_{u \in S} \sum_{v \notin S} f_{u,v}$$



Thm (Max Flow Min Cut Theorem)
when FF Algorithm terminates, if
f is flow at end of algorithm
and S is set of vertices reachable
from s in residual network:

$$val(f) = cap(S)$$



S = set of vertices reachable
from s in residual
network

$r_{a,b} = c_{a,b} - f_{a,b}$ $\quad c_{a,b} = f_{a,b}$

$$val(f) = \sum_{a \in S} \sum_{b \notin S} f_{a,b} = \sum_{\substack{a \in S \\ b \notin S}} c_{a,b} = cap(S)$$