CSS1051: ADVANCED COMPUTING LAB-1

Instructor

Dr. Debasis Mitra
Associate Professor
Department of Computer Science and Engineering
National Institute of Technology Durgapur

Disclaimer

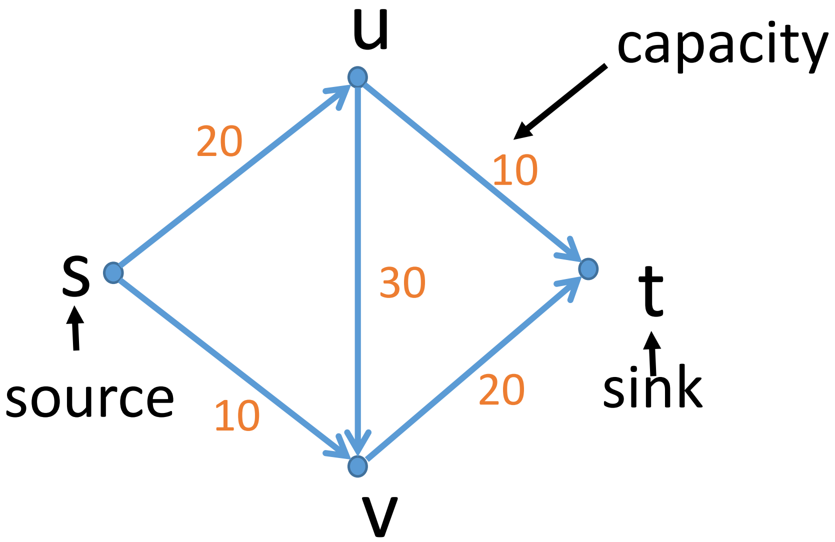
- The study materials/presentations used in this course are solely meant for academic purposes
- The study materials presented/ distributed can be reused, reproduced, modified, and distributed by others for academic purposes only with proper acknowledgements

Max Flow problem

Max Flow problem

- Flow network and flow:
  - A flow network is a tuple $G = (V, E, s, t, c)$
    - Digraph (V, E) with source $s \in V$ and sink $t \in V$
    - Capacity $c(e) \geq 0$ for each $e \in E$

Intuition: Material flowing through a transportation network; material originates at source and is sent to sink



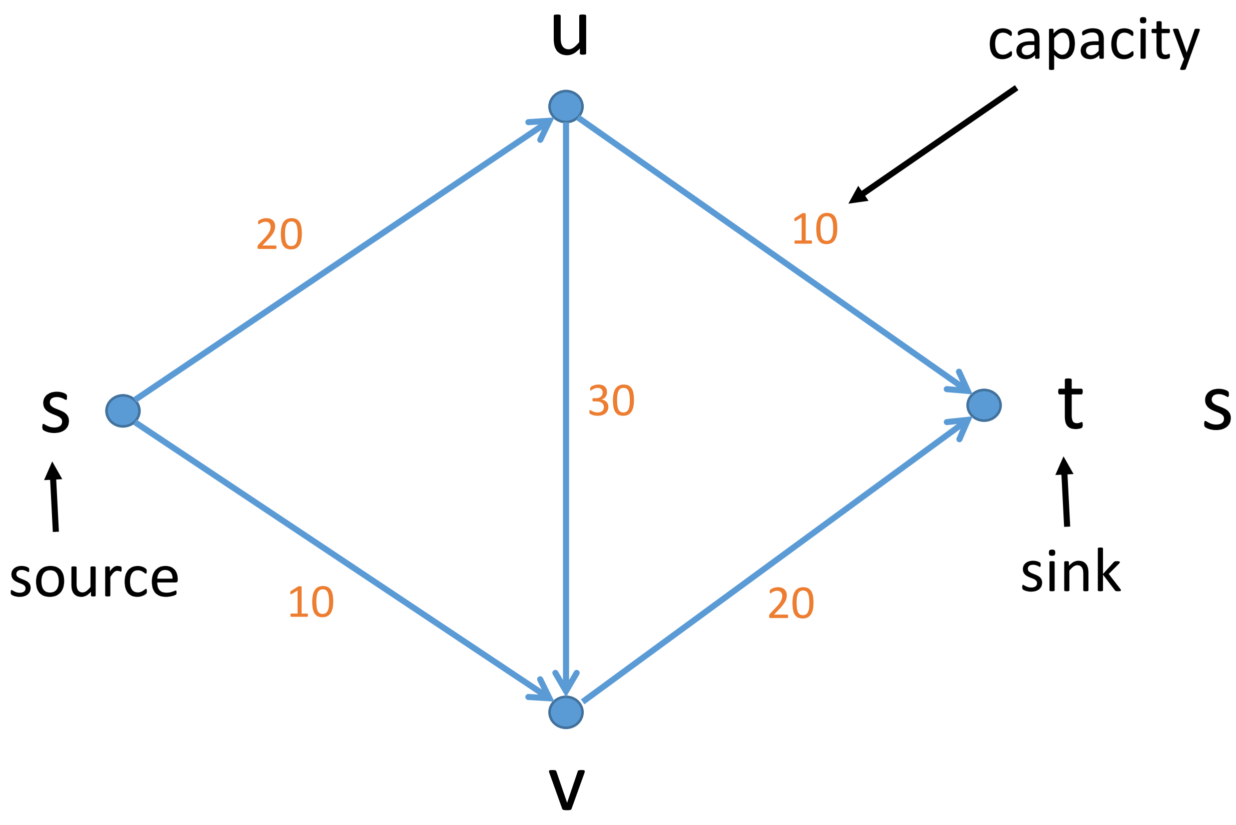
Max Flow problem

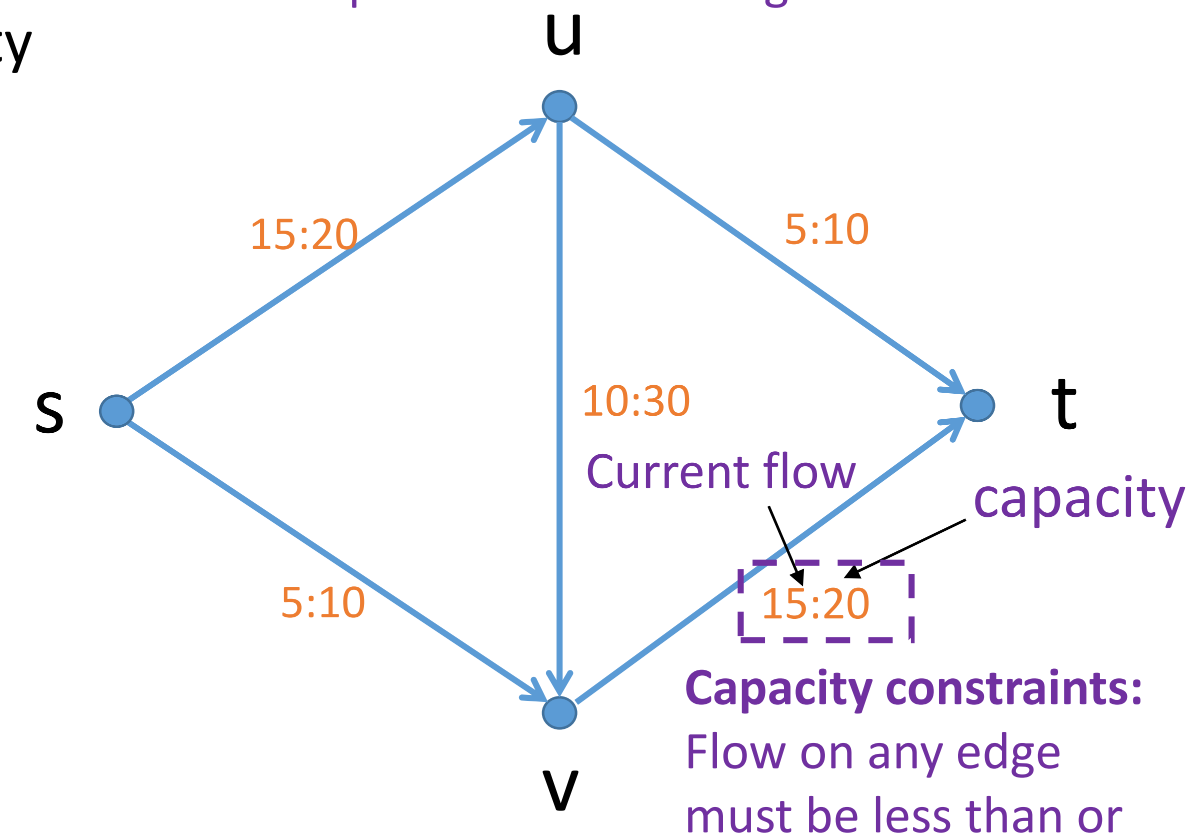
- Flow network and flow:
  - An *s*-*t* flow (flow) *f* is a function that satisfies:
    - For each $e \in E$: $0 \leq f(e) \leq c(e)$ [capacity constraint]
    - For each $v \in V - \{s, t\}$: $\sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]
  - The value of a flow *f* is : $val(f) = \sum_{e \text{ out of } s} f(e) - \sum_{e \text{ in to } s} f(e)$

Max Flow problem

- Flow network and flow:

Flow conservation: total positive flow entering a vertex (except source and sink) must be equal to total positive flow leaving that vertex

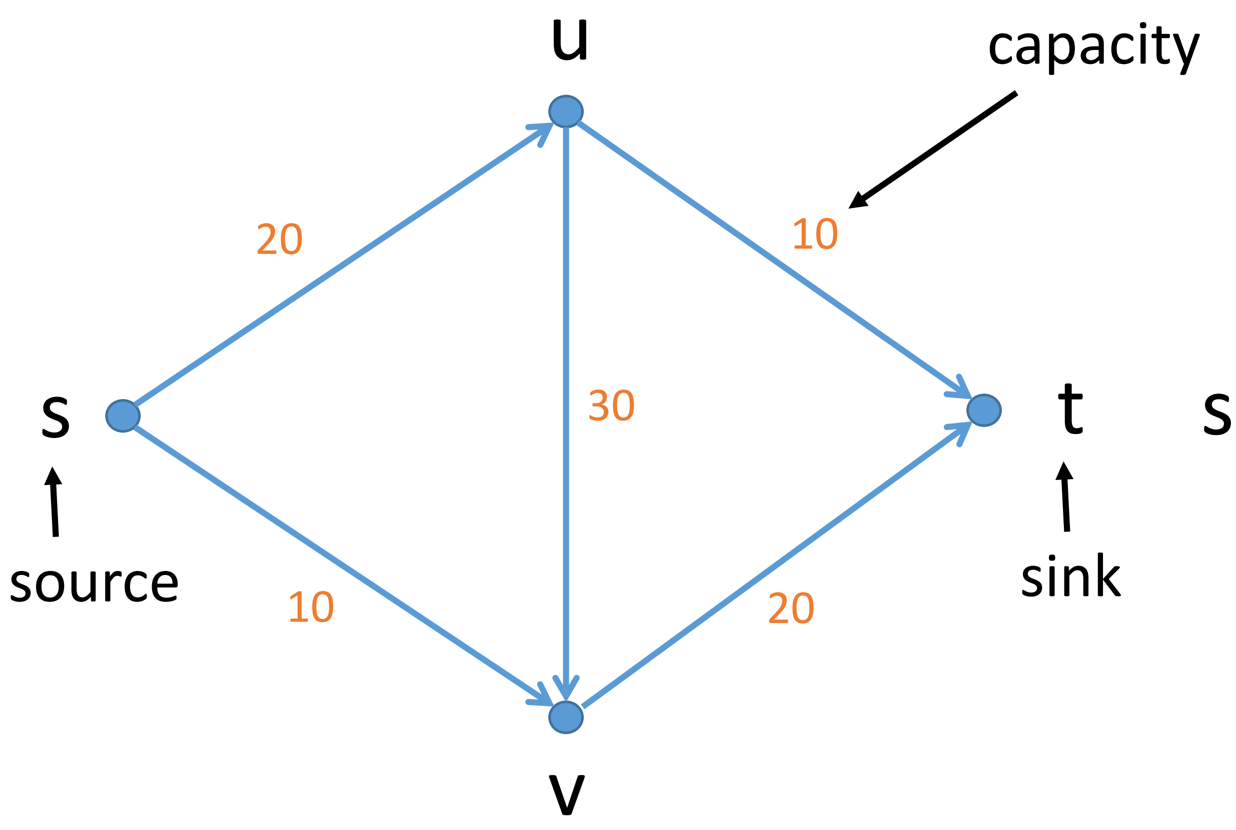


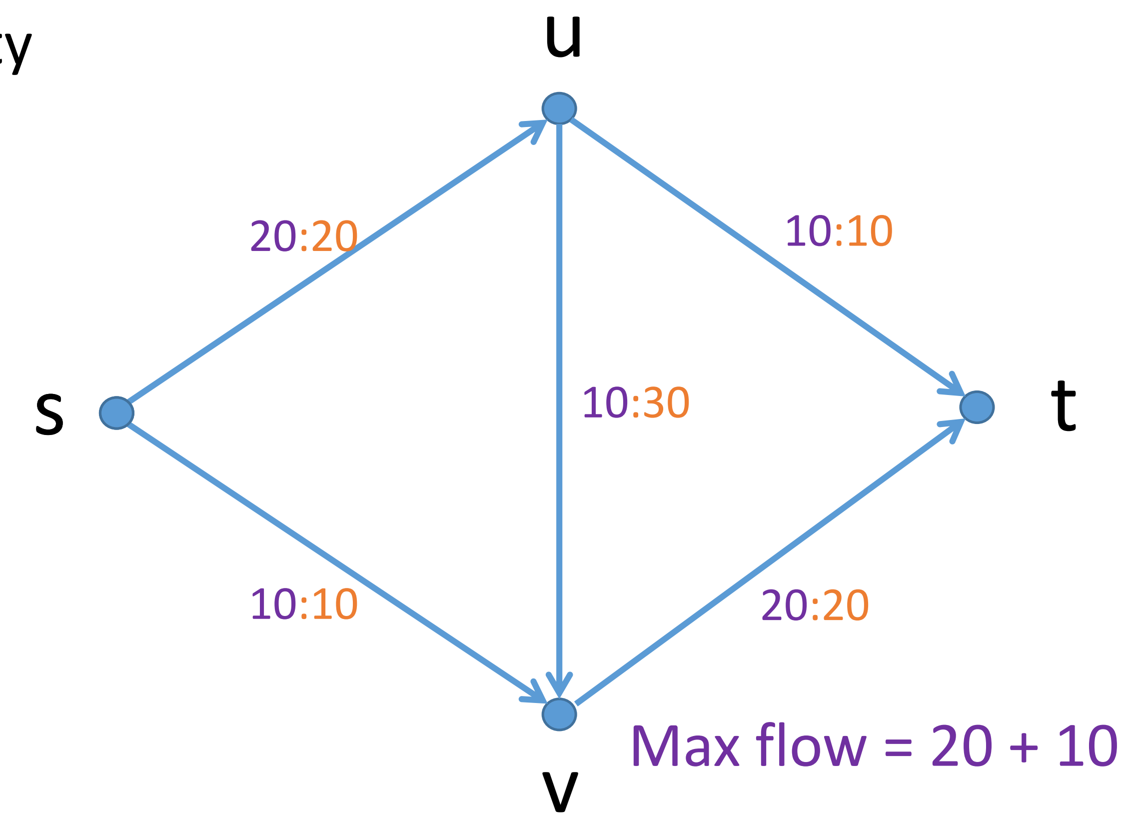


Capacity constraints: Flow on any edge must be less than or equal to its capacity

Max Flow problem

- Max flow: Maximum flow that can be pushed through the network without violating capacity constraints and flow conservation

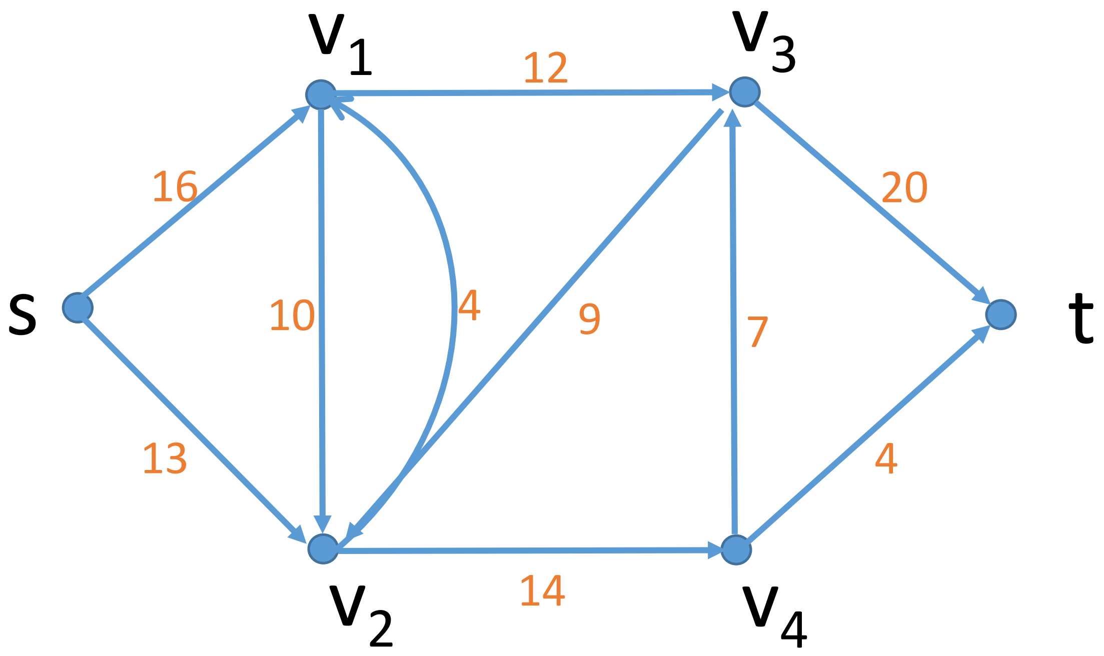




Max flow = 20 + 10 = 30

Max Flow problem

- Max flow: Maximum flow that can be pushed through the network without violating capacity constraints and flow conservation



Upper bound?

- Minimum of the following:
  - Maximum capacity of edges emanating from s
  - Maximum capacity of edges entering t
- **Any other way to get tighter bound?**

Max flow using Ford-Fulkerson method

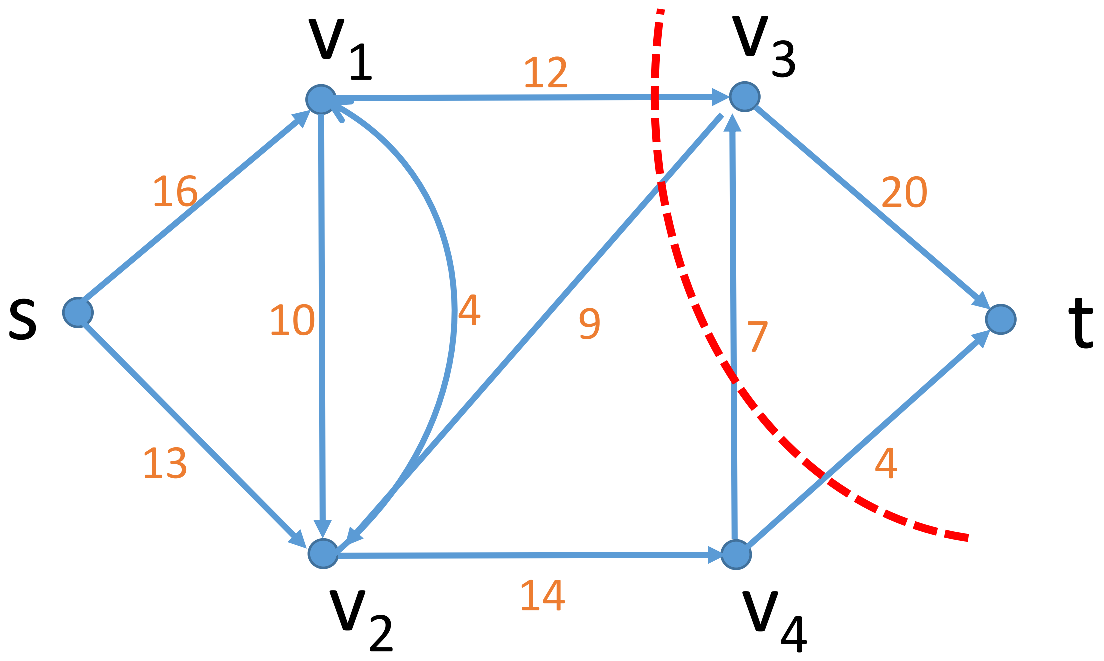
- **Max-flow min cut theorem:**
  - ✓**Value of a max flow = capacity of a min cut**

- **Cut:**
  - ➢ A cut (S, T) or s-t cut of flow networks is a partition of V into two disjoint sets S and T such that $s \in S$ and $t \in T$.
    - ➢Net flow of across a cut (including positive and negative flows) is called flow of the cut
      - At any point of time, flow across any cut of the network is same
    - ➢**Capacity of a cut** is computed by adding the capacities of edges going from S to T only (not from T to S)

      A minimum cut of a network is a cut whose capacity is minimum over all cuts of the network

Max Flow problem

- **Bottleneck:** Min cut (Minimum of the sum of capacities of all forward edges of any s-t cut)

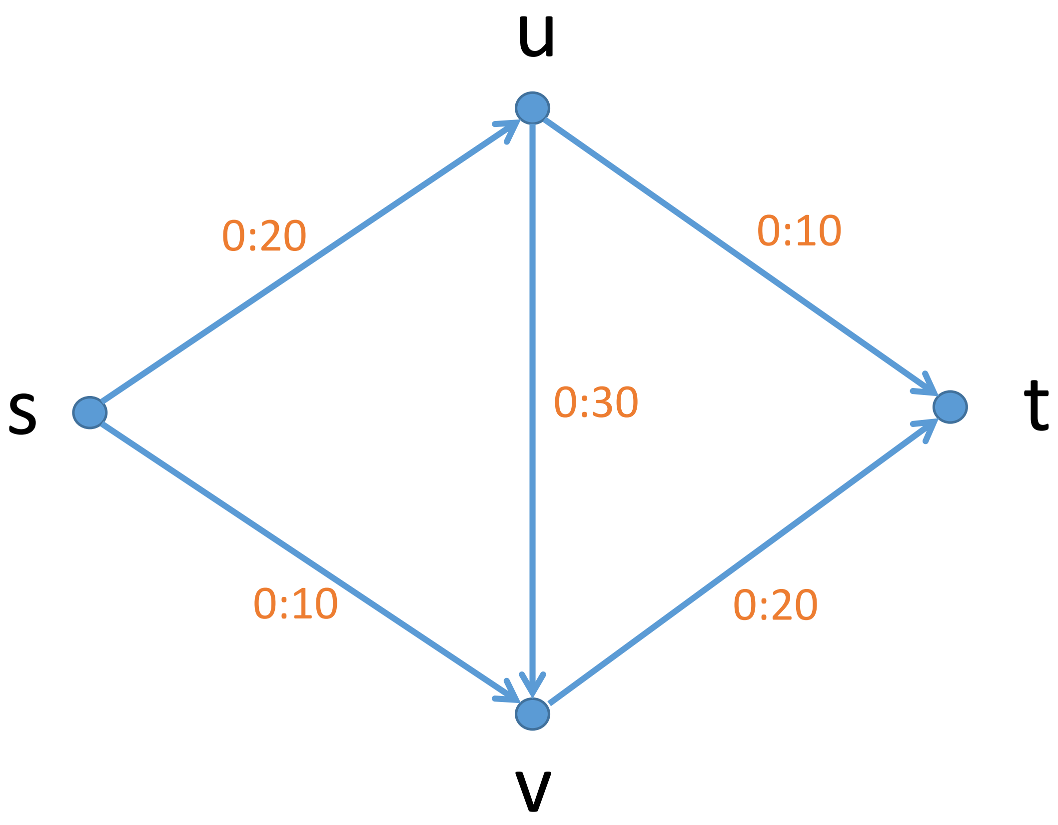


Max-flow min-cut theorem*

* To be discussed latter

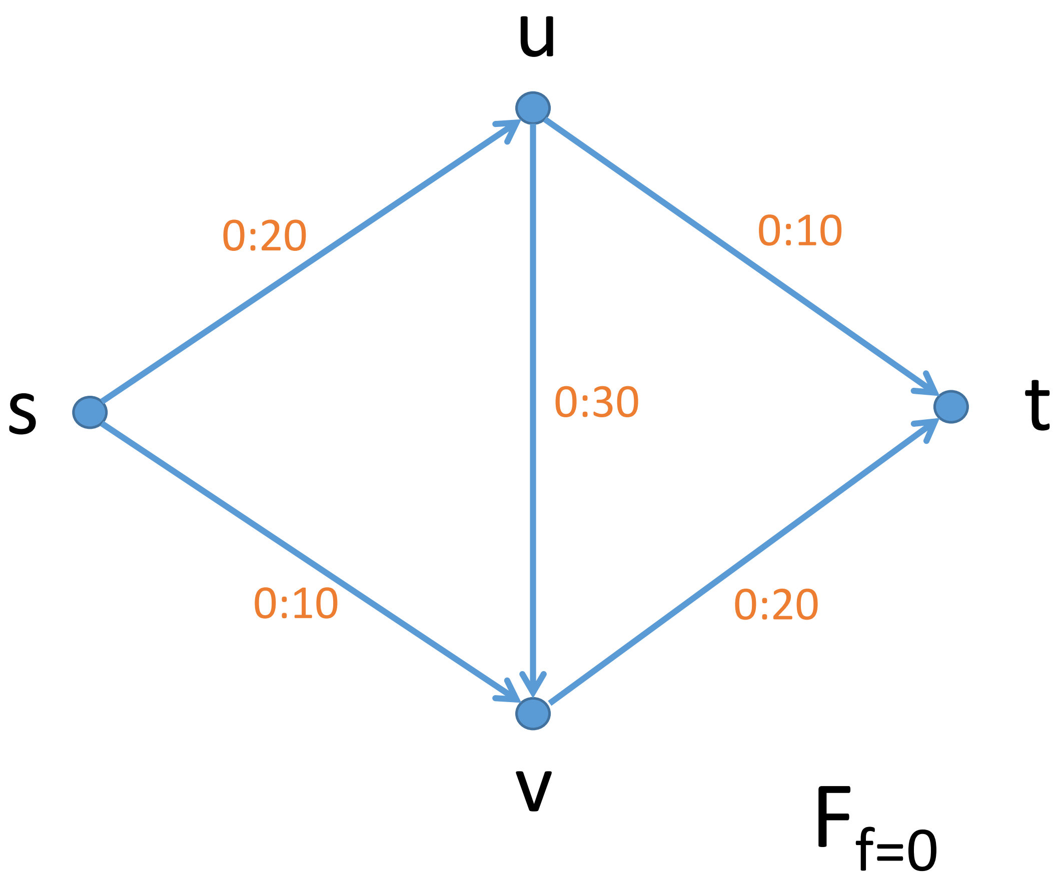
Max Flow problem

- Algorithm Design:



Max Flow problem

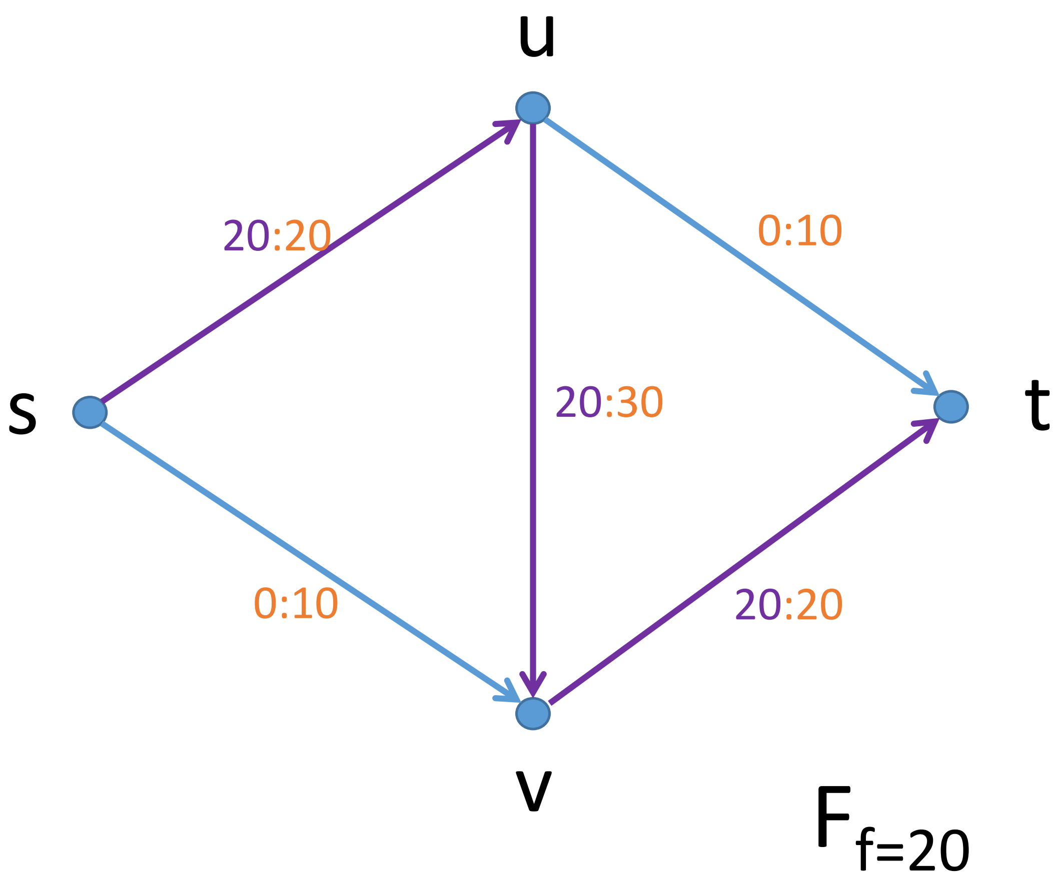
- Algorithm Design:



- Start with 0 flow
- Increase the flow by adding streams (s-t paths) of flow one by one
  - Choose the edges with highest residual capacity while selecting the path [Greedy approach]

Max Flow problem

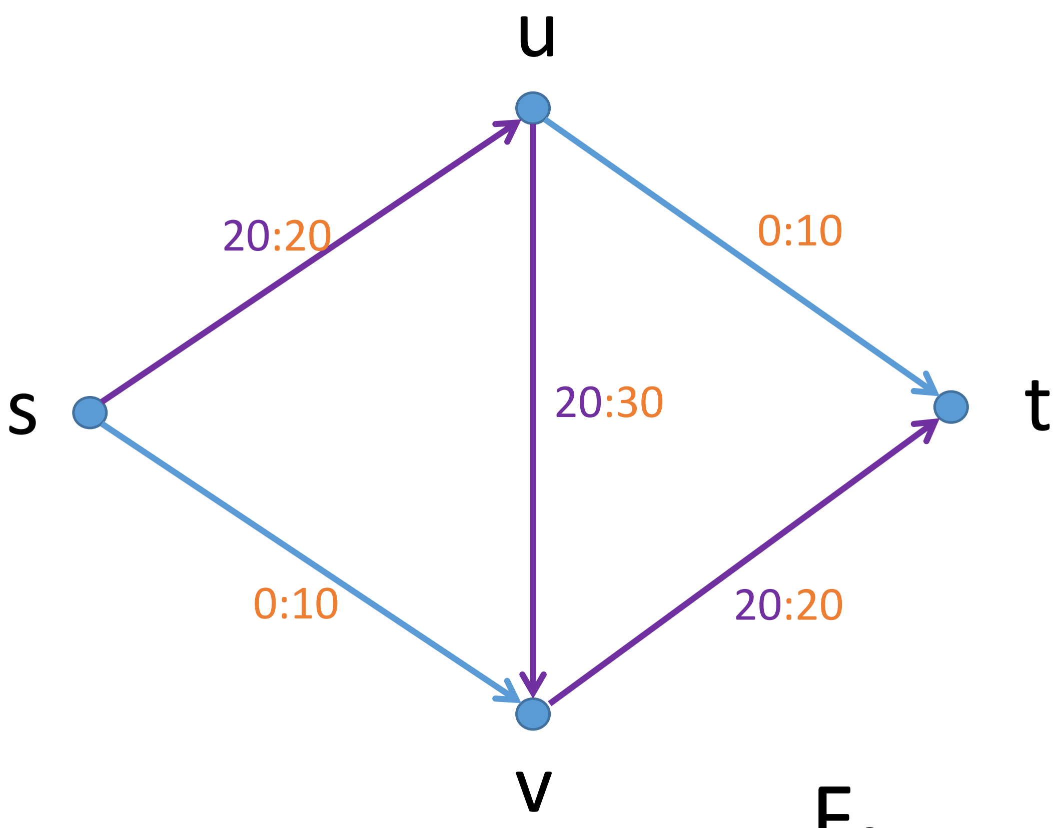
- Algorithm Design:



- Start with 0 flow
- Increase the flow by adding streams (s-t paths) of flow one by one
  - Choose the edges with highest residual capacity while selecting the path [Greedy approach]

Max Flow problem

- Algorithm Design:



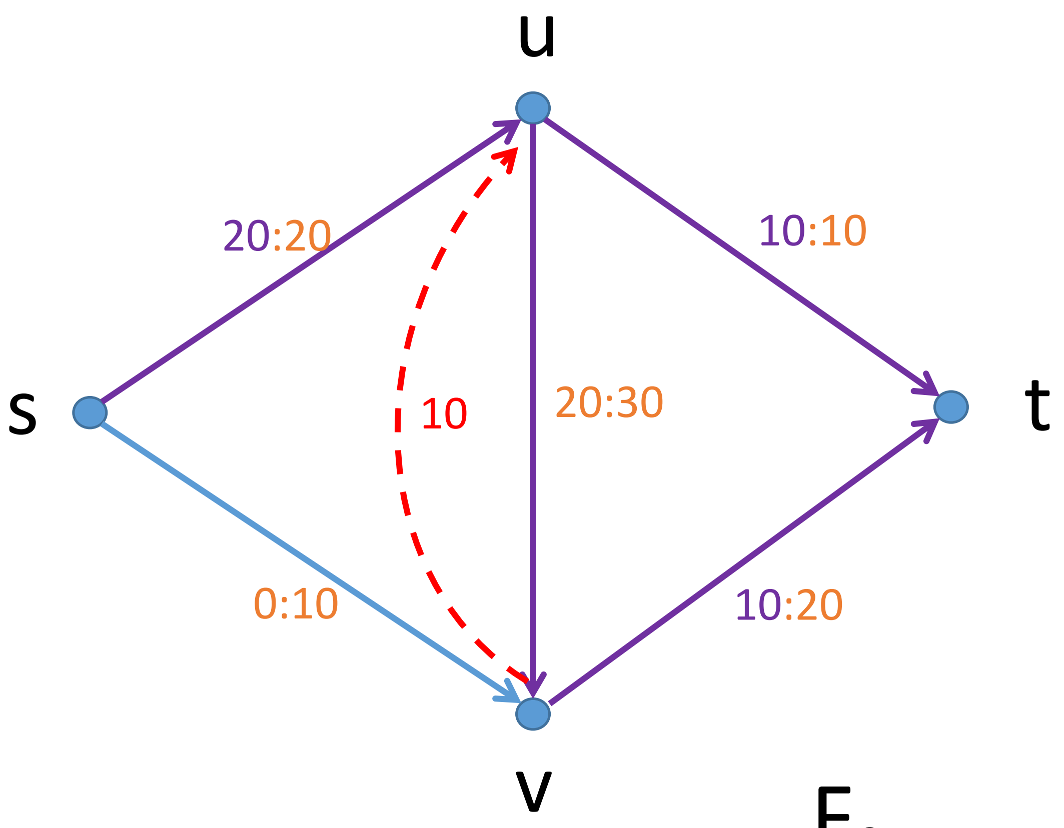
$F_{f=20}$

- Start with 0 flow
- Increase the flow by adding streams (s-t paths) of flow one by one
  - Choose the edges with highest residual capacity while selecting the path [Greedy approach]

- Can we add more flow?
  - Can we find another path?
- Is it (f = 20) max flow?

Max Flow problem

- Algorithm Design:



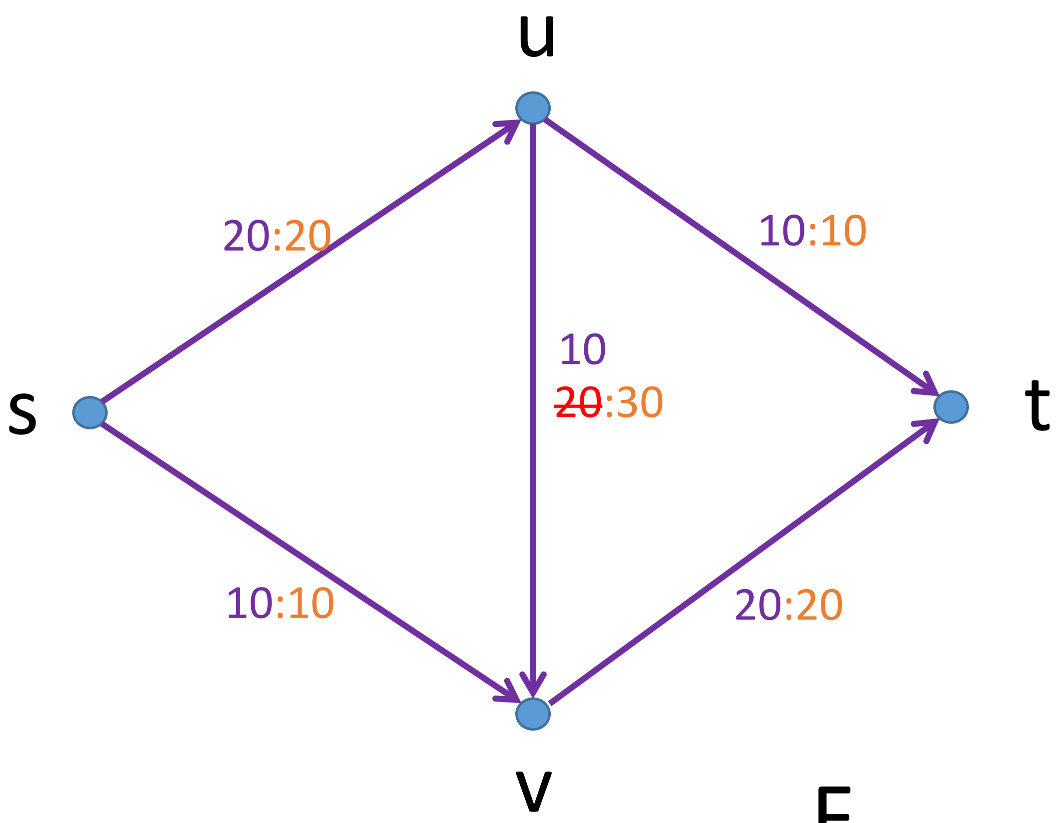
$F_{f=20}$

- Start with 0 flow
- Increase the flow by adding streams (s-t paths) of flow one by one
  - Choose the edges with highest residual capacity while selecting the path [Greedy approach]

- Can we add more flow?
  - Yes…
  - If we can **undo** (push back) 10 units of flow along u-v and divert that along u-t

Max Flow problem

- Algorithm Design:



$F_{f=20}$

- Start with 0 flow
- Increase the flow by adding streams (s-t paths) of flow one by one
  - Choose the edges with highest residual capacity while selecting the path [Greedy approach]

- Can we add more flow?
  - Yes…
  - If we can **undo** 10 units of flow along u-v and push that along u-t
  - Then we can add one more stream of flow 10 along s-v-t