Introduction to Network Coding, Bounds and Constructions

Tuvi Etzion

Lecture 2

Routing, Unicast, and Multicast
Routing Schemes

Outline

- Network flow
- Menger's Theorem and Edmonds Theorem
- Spanning trees and Steiner trees
- Basic concepts in routing and network coding
- Routing capacity
Network flow deals with modelling the flow of a commodity (water, electricity, packets, gas, cars, trains, money, etc.) in a network. The links in the network are capacitated and the commodity does not vanish in the network except at specified locations where we can either inject or extract some amount of commodity. The main question is how much commodity can be on the network sent in one iteration?
Let $G = (V, E)$ be an acyclic directed graph with two special vertices $s$ and $t$; $s$ is called the source and $t$ is called the sink. Each edge $e \in E$ has a positive integer capacity $c(e)$. A flow $f$ is an assignment of values to the edges such that:

- for every $e \in E$: $0 \leq f(e) \leq c(e)$;
- for every $v \in V \setminus \{s, t\}$: $\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e)$, where $\delta^+(v)$ is the set of edges leaving $v$ and $\delta^-(v)$ is the set of edges entering $v$.

The flow value is the net flow leaving $s$ which is equal to the net flow entering $t$. 
Network Flow

For a given set of vertices $S, S \subset V, s \in S, t \notin S$, we define the cut of $S$ as the set of edges $\delta^+(S) = \{(u, v) : (u, v) \in E, u \in S, v \in V \setminus S\}$. A cut is also called an $s-t$ cut. The capacity $C(S)$ of the cut $S$ is

$$C(S) = \sum_{e \in \delta^+(S)} c(e).$$

min-cut/max-flow Theorem

For any maximum flow problem for which a feasible flow exists, we have that the maximum $s-t$ flow value is equal to the minimum capacity of any $s-t$ cut.
Routing Schemes

A network will be an acyclic directed graph
\[ G = (V, E) \]
with \( h \) sources (transmitters)
\[ S_1, \ldots, S_h \in V \]
and \( N \) receivers (destinations)
\[ R_1, \ldots, R_N \in V \]. Each source has a set of messages and each receiver has a set of demands which is a set of messages from all the sources.

- **Unicast** - one source and one receiver
- **Multicast** - one source and many receivers
- **Broadcast** - one source, the rest receivers
**Routing Schemes**

- **Unicast** - one source and one receiver which demands all the messages.
- **Multicast** - one source and many receivers, each one demands all the messages.
- **Broadcast** - one source, the rest receivers, each one demands all the messages.
- **Multiple unicast** - many sources and many receivers. The messages of each source are demanded by exactly one receiver.
- **Multiple multicast** - many sources and many receivers. The messages of each source can be demanded by each receiver.
Network Flow

$G = (V, E)$

$s \in V$

t $\in V$, $T \subseteq V$

Capacity for unicast

$C_G(s, t)$ - the capacity of the min-cut

Capacity for multicast

$C_G(s, T)$ - the minimum capacity of $C_G(s, t)$, $t \in T$
Menger’s Theorem

Let $G = (V, E)$ be a unit capacity flow network. There are $k$ edge disjoint paths in $G$ from $s$ to $t$ if and only if the maximum value of an $s - t$ flow in $G'$ is at least $k$.

Proof

$\Rightarrow$ trivial.

$\Leftarrow$ by induction on $k$. At each step one path is removed from the graph and the induction hypothesis is used.
**Menger’s Theorem**

Let $G = (V, E)$ be a unit capacity flow network. There are $k$ edge disjoint paths in $G$ from $s$ to $t$ if and only if the maximum value of an $s-t$ flow in $G'$ is at least $k$.

**Corollary**

In every directed graph with vertices $s$ and $t$, the maximum number of edge disjoint $s-t$ paths is equal to the minimum number of edges whose removal separates $s$ from $t$.

**Proof**

By Menger’s Theorem the maximum number of paths is equal to the maximum flow which is equal by the min-cut/max-flow Theorem to the minimum capacity of a cut which is equal to the minimum number of edges whose removal separates $s$ from $t$. 
Edmonds Theorem

In a directed graph $G = (V, E)$ there are $k$ edge disjoint spanning trees rooted at $r \in V$ if and only if $k \leq C_G(r, V \setminus \{r\})$.

Proof

⇒ Trivial.

⇐ The proof is by induction on $k$. Let $D = \{U \subset V : r \in U\}$. By Menger's Theorem $|\delta^+(U)| \geq k$ for each $U \in D$. We have to find a spanning tree with set of edges $E'$ such that $|\delta^+(U) \setminus E'| \geq k - 1$ for each $U \in D$.
Edmonds Theorem

Proof
We start with a set $E' = \emptyset$ and $S = \{r\}$, where $E'$ is the set of edges in the current formed tree and $S$ is its set of vertices. We must always have that $|\delta^+(U) \setminus E'| \geq k - 1$, for each $U \subset S$. If $S = V$ then the inductive part is done.

A set $X \subset V$ is critical if

- $X \in D$ ;
- $X \cup S \neq V$ ;
- $|\delta^+(X) \setminus E'| = k - 1$ .

If there is no critical set then any edge of $\delta^+(S)$ can be used to augment $E'$. If $X$ is a critical set then no edge of $\delta^+(X)$ can be used to augment $E'$. 
We will now prove that if $X$, $Y$, where $X \cup Y \neq V$, are critical, then $X \cap Y$ and $X \cup Y$ are critical. If $G' = (V, E \setminus E')$ then since $X$ and $Y$ are critical $|\delta_{G'}^+(X)| = |\delta_{G'}^+(Y)| = k - 1$. Now,

$$2(k - 1) = |\delta_{G'}^+(X)| + |\delta_{G'}^+(Y)| \geq |\delta_{G'}^+(X \cup Y)| + |\delta_{G'}^+(X \cap Y)|.$$

Only edges from $X$ to $Y$ and from $Y$ to $X$, but not from $X \cap Y$ are not counted in both sides of the equation (counted only in the left side).

Since $r \in X \cup Y$, $r \in X \cap Y$, $X \cup Y \neq V$, it follows that $|\delta_{G'}^+(X \cup Y)| \geq k - 1$ and $|\delta_{G'}^+(X \cap Y)| \geq k - 1$. Thus, $|\delta_{G'}^+(X \cup Y)| = k - 1$ and $|\delta_{G'}^+(X \cap Y)| = k - 1$ which implies that $X \cap Y$ and $X \cup Y$ are critical.
Proof

Let $X$ be a maximal critical set. Since the capacity of the min-cut is at least $k$ and $|\delta^+_{G \setminus E'}(X)| = k - 1$, it follows that there exists an edge $(u, v) \in E$ such that $u \in S \setminus X$ and $v \notin S \cup X$.

Let $E' = E' \cup \{(u, v)\}$.

If there exist $Y \in D$ such that $|\delta^+(Y) \setminus E'| < k - 1$, then $|\delta^+(Y) \setminus (E' \setminus \{(u, v)\})| = k - 1$ and hence $Y$ is a critical set and $(u, v) \in \delta^+(Y)$. Now, $X$ and $Y$ are critical and $Y \cup X \neq V$ since $v \notin Y \cup X$. Therefore, $Y \cup X$ is critical which contradicts the choice of $X$ as a maximal critical set. The theorem follows.
Let $G = (V, E)$ be an undirected graph, in which each edge has some weight, and let $T$ be a subset of vertices from $V$. A Steiner tree is a tree, with minimum sum for the weights of the edges, which contains all the vertices of $T$. The vertices in this tree which are not contained in $T$ are called Steiner vertices.
The Steiner tree problem in graphs is defined as a decision problem:

**Instance**
- an undirected graph $G = (V, E)$;
- a subset of the vertices $T \subseteq V$, called terminal nodes;
- a number $k \in \mathbb{N}$.

**Question**
Is there a subtree $G'$ of $G$ that includes all the vertices of $T$ (a spanning tree for $G'$) and contains at most $k$ edges?

**Theorem**
The Steiner tree problem is NP-complete.

Multicasting is NP-hard.
Steiner Trees

Rectilinear Steiner Trees

The vertices of the set $T$ are points in the plane and they should be connected only by horizontal and vertical lines, where the sum of the lengths of these lines is minimized.

The decision problem for the rectilinear Steiner tree problem is also NP-complete.
Introduction to Network Coding, Bounds and Constructions

A SHORT BREAK
A network $G = (V, E)$ is an acyclic directed graph. The network has $h$ messages that should be delivered from the sources (transmitters) to the receivers subject to the demands of the receivers.

The $h$ messages are represented by vectors of length $k$ over an alphabet $\Sigma$. Each edge carries a packet, which is a vector of length $n$ over $\Sigma$.

Each out-going edge $e$ of a vertex $v$ computes a function of the packets on the in-coming edges of $v$ and the messages of $v$. The result of this function is the packet on the edge $e$. 
The $h$ messages are represented by vectors of length $k$ over an alphabet $\Sigma$. Each edge carries a packet, which is a vector of length $n$ over $\Sigma$.

Each out-going edge $e$ of a vertex $v$ computes a function of the packets on the in-coming edges of $v$ and the messages of $v$. The result of this function is the packet on the edge $e$.

Can each receiver obtain its demands?

If yes, then the network is solvable.

If yes, what is the smallest alphabet size required?
Multicast Networks

**Multicast network**

A source $S$, $h$ messages, $N$ receivers.

A network with $h$ sources, one message per source, $N$ receivers.

From the network with $h$ sources we construct a new network $G = (V, E)$ by adding a new source $S$ and connecting it to the previous $h$ sources.

Alphabet size is not reduced

What about the other direction?

Equivalent?
Multicast Networks

A source $S$, $h$ messages, $N$ receivers.

A network with $h$ sources $S_1, \ldots, S_h$, one message per source, $N$ receivers.

From the network with one source and $h$ messages we construct a new network $G = (V, E)$ by adding $h$ new sources, each one will carry a different message, and connecting them to the previous source.

OR

Equivalent?

Alphabet size remains the same
The two messages received at node 4 are demanded by one of the receivers. The receiver node that demands the messages of node 4 cannot obtain these messages.

Each edge has capacity one.

No routing solution

Each message is sent to exactly one of the nodes (3 - 5).
Linear Network Coding

Each edge has capacity one.

Linear network coding

Assume edges (1, 3) and (1, 4) carry a linear combination of $A$ and $A'$. 

Node 5 cannot supply the rest of the information ($B$ or $B'$) to all the receivers.
Assume edges (1, 3) and (1, 4) carry a linear combination of $A$ and $A'$.

Node 5 cannot supply the rest of the information ($B$ or $B'$) to all the receivers.

To remove the combination of $B$ and $B'$ (sent by node 4) to obtain $A$ or $A'$ (in nodes 6 – 9), the edges (2, 4) and (2, 5) must carry the same information.

Either $B$ or $B'$ can be recovered, not both.
Assume edge \((1, 3)\) carries a linear combination of \(A\) and \(A'\) and \((1, 4)\) carries w.l.o.g. \(A\).

Node 5 cannot supply the rest of the information (\(B\) or \(B'\)) to all the receivers.

To remove the combination of \(B\) and \(B'\) (sent by node 4) to obtain \(A\) or \(A'\) (in nodes 6 – 9), the edges \((2, 4)\) and \((2, 5)\) must carry the same information.

Either \(B\) or \(B'\) can be recovered, not both.
W.l.o.g. edge (1, 3) carries $A$ and edge (2, 5) carries $B$.

Node 4 does not need to send any part of $A$ or $B$.

Hence, the edge (1, 4) carries $A'$ and the edge (2, 4) carries $B'$.

Similar arguments implies that edge (2, 5) does not carry a linear combination of $B$ and $B'$.

No solution.
Linear Network Coding

No linear coding solution.

The $M$-network

Koetter, Medard, Effros, Karger, Ho 2003
Scalar Routing vs. Vector Routing

Solved! with routing only!
Routing Capacity

Concepts for routing capacity

Cannons, Dougherty, Freiling, Zeger 2006

Messages - vectors of length $k$.
Information on edges - vectors of length $n$.

The routing capacity is the supremum of ratios of message dimension to edge capacity $\frac{k}{n}$ for which a routing solution exists.
Routing Capacity

Concepts for routing capacity

- **Messages** - vectors of length $k$.
- **Packets on edges** - vectors of length $n$.

The routing capacity is the supremum of ratios of message dimension to edge capacity $\frac{k}{n}$ for which a routing solution exists.

A $(k, n)$ fractional routing solution is a vector routing solution that uses messages with $k$ components and edges with capacity $n$.

Clearly, the routing capacity does not depend on the alphabet size!!!
Routing Capacity

Concepts for routing capacity

The routing capacity is the supremum of ratios of message dimension to edge capacity $\frac{k}{n}$ for which a routing solution exists.

Clearly, the routing capacity does not depend on the alphabet size!!!

Theorem

The routing capacity of every network is rational and achievable.

Cannons, Dougherty, Freiling, Zeger 2006
Routing Capacity

The butterfly network

The only path of $x$ from node 1 to node 6 is $(1, 3), (3, 4), (4, 6)$.

The only path of $y$ from node 2 to node 5 is $(2, 3), (3, 4), (4, 5)$.

Therefore, there must be enough capacity along edge $(3, 4)$ for both messages.

$$
\Rightarrow n \geq 2k \Rightarrow \frac{k}{n} \leq \frac{1}{2}
$$

$\epsilon = \frac{1}{2}$
Routing Capacity

If two of the three edges \((1, 2), (1, 3), (4, 5)\), will be removed, then there will be no path between the source and one of the receivers.

Therefore, each message symbol must appear in at least two of the three edges \((1, 2), (1, 3), (4, 5)\).

\[\Rightarrow 3n \geq 2(2k) \Rightarrow k/n \leq \frac{3}{4}\]

\[\epsilon = \frac{3}{4}\]
If each one of the three edges \((1, 2), (1, 3), (4, 5)\) carries a message of length \(n\), then one of receivers 6 and 7 obtain at most \(n + \frac{n}{2}\) symbols.

\[
\Rightarrow n + \frac{n}{2} \geq k \Rightarrow \frac{k}{n} \leq \frac{3}{2}
\]

\[\epsilon = \frac{3}{2}\]