MA. Raayatpanah

Constrained subgraph selection over coded packet networks

Mohammad Ali Raayatpanah

Department of Mathematics Sciences and Computer,
Kharazmi University,
Tehran, Iran.

August 19, 2015

Outline

- Introduction to subgraph selection
- Minimum-cost subgraph selection with a single multicast session
- Constrained subgraph selection with a single multicast session
- Constrained subgraph selection with multiple multicast sessions

Outline

- Introduction to subgraph selection
- Minimum-cost subgraph selection with a single multicast session
- Constrained subgraph selection with a single multicast session
- Constrained subgraph selection with multiple multicast sessions

Outline

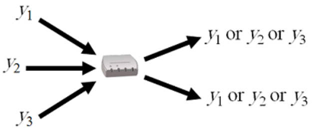
- Introduction to subgraph selection
- Minimum-cost subgraph selection with a single multicast session
- Constrained subgraph selection with a single multicast session
- Constrained subgraph selection with multiple multicast sessions

Outline

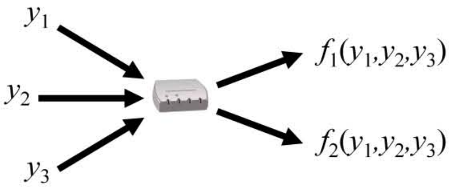
- Introduction to subgraph selection
- Minimum-cost subgraph selection with a single multicast session
- Constrained subgraph selection with a single multicast session
- Constrained subgraph selection with multiple multicast sessions

Routing vs Network Coding

- Traditional routing in a node



- Network coding in a Node



Some benefits of network coding over routing

- Higher throughput
- Higher reliability
- Higher security
- Cheaper routing costs networks
- Lower delays

Some benefits of network coding over routing

- Higher throughput
- Higher reliability
- Higher security
- Cheaper routing costs networks
- Lower delays

Some benefits of network coding over routing

- Higher throughput
- Higher reliability
- Higher security
- Cheaper routing costs networks
- Lower delays

Some benefits of network coding over routing

- Higher throughput
- Higher reliability
- Higher security
- Cheaper routing costs networks
- Lower delays

Some benefits of network coding over routing

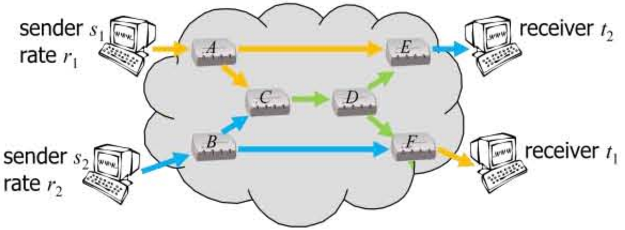
- Higher throughput
- Higher reliability
- Higher security
- Cheaper routing costs networks
- Lower delays

Some benefits of network coding over routing

- Higher throughput
- Higher reliability
- Higher security
- Cheaper routing costs networks
- Lower delays

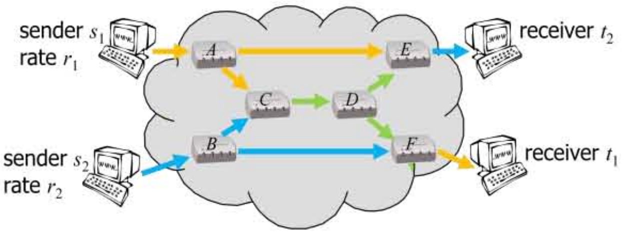
Network Model

- A network can be expressed as a directed graph $G = (N, A)$
  1. N denotes the set of nodes (routers or switches)
  2. A denotes the set of directed arcs
     Arcs represent the communication link between nodes



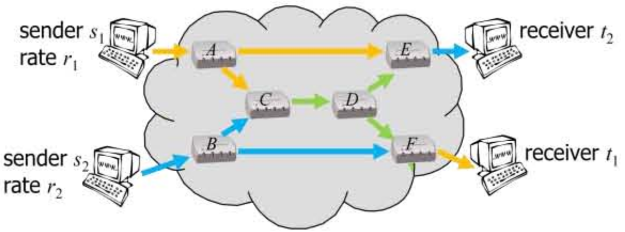
Network Model

- A network can be expressed as a directed graph $G = (N, A)$
  1. N denotes the set of nodes (routers or switches)
  2. A denotes the set of directed arcs.
     Arcs represent the communication link between nodes.



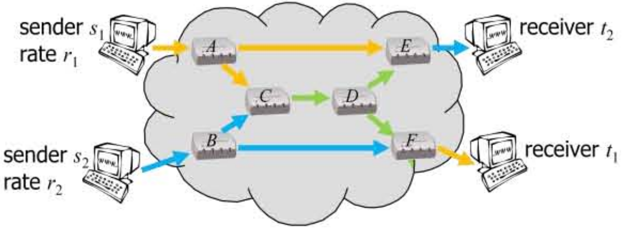
Network Model

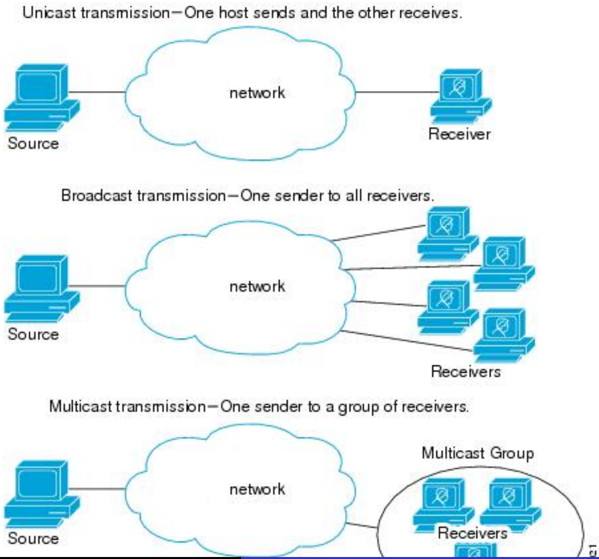
- A network can be expressed as a directed graph $G = (N, A)$
  1. N denotes the set of nodes (routers or switches)
  2. A denotes the set of directed arcs.
     Arcs represent the communication link between nodes.



Network Model

- A network can be expressed as a directed graph $G = (N, A)$
  1. N denotes the set of nodes (routers or switches)
  2. A denotes the set of directed arcs.
     Arcs represent the communication link between nodes.




Unicast transmission—One host sends and the other receives.
network
Source
Receiver
Broadcast transmission—One sender to all receivers.
network
Source
Receivers
Multicast transmission—One sender to a group of receivers.
Multicast Group
network
Source
Receivers

- Network coding can achieve the maximum multicast rate
  1. It is not achievable by routing alone.
- The problem of establishing multicast connection with network coding can be decomposed into two parts:
  1. Determining the subgraph to code over
  2. Determining the code to use over that subgraph

- Network coding can achieve the maximum multicast rate
  1. It is not achievable by routing alone.
- The problem of establishing multicast connection with network coding can be decomposed into two parts:
  1. Determining the subgraph to code over
  2. Determining the code to use over that subgraph.

- Network coding can achieve the maximum multicast rate
  1. It is not achievable by routing alone.
- The problem of establishing multicast connection with network coding can be decomposed into two parts:
  1. Determining the subgraph to code over
  2. Determining the code to use over that subgraph

- Network coding can achieve the maximum multicast rate
  1. It is not achievable by routing alone.
- The problem of establishing multicast connection with network coding can be decomposed into two parts:
  1. Determining the subgraph to code over
  2. Determining the code to use over that subgraph

- Network coding can achieve the maximum multicast rate
  1. It is not achievable by routing alone.
- The problem of establishing multicast connection with network coding can be decomposed into two parts:
  1. Determining the subgraph to code over
  2. Determining the code to use over that subgraph

difference between Subgraph selection and coding

- Subgraph selection and coding are very different problems!

  1. Coding generally uses techniques from information theory and coding theory
  2. Subgraph selection is essentially a problem of network resource allocation and generally uses techniques from networking theory

- **In this talk**, we focus to find an efficient subgraph that allows the given multicast connection to be established over coded packet networks
- *The analogous problem for routed network is the Steiner tree problem, which is NP complete.*

Difference between Subgraph selection and coding

- Subgraph selection and coding are very different problems!
  1. Coding generally uses techniques from information theory and coding theory
  2. Subgraph selection is essentially a problem of network resource allocation and generally uses techniques from networking theory.
- **In this talk**, we focus to find an efficient subgraph that allows the given multicast connection to be established over coded packet networks
- *The analogous problem for routed network is the Steiner tree problem, which is NP complete.*

- Subgraph selection and coding are very different problems!
  1. Coding generally uses techniques from information theory and coding theory
  2. Subgraph selection is essentially a problem of network resource allocation and generally uses techniques from networking theory.
- **In this talk**, we focus to find an efficient subgraph that allows the given multicast connection to be established over coded packet networks
- *The analogous problem for routed network is the Steiner tree problem, which is NP complete.*

- Subgraph selection and coding are very different problems!
  1. Coding generally uses techniques from information theory and coding theory
  2. Subgraph selection is essentially a problem of network resource allocation and generally uses techniques from networking theory.
- **In this talk**, we focus to find an efficient subgraph that allows the given multicast connection to be established over coded packet networks
- *The analogous problem for routed network is the Steiner tree problem, which is NP complete.*

difference between Subgraph selection and coding

- Subgraph selection and coding are very different problems!
  1. Coding generally uses techniques from information theory and coding theory
  2. Subgraph selection is essentially a problem of network resource allocation and generally uses techniques from networking theory.
- **In this talk**, we focus to find an efficient subgraph that allows the given multicast connection to be established over coded packet networks
- *The analogous problem for routed network is the Steiner tree problem, which is NP complete.*

Difference between Subgraph selection and coding

- Subgraph selection and coding are very different problems!
  1. Coding generally uses techniques from information theory and coding theory
  2. Subgraph selection is essentially a problem of network resource allocation and generally uses techniques from networking theory.
- **In this talk**, we focus to find an efficient subgraph that allows the given multicast connection to be established over coded packet networks
- *The analogous problem for routed network is the Steiner tree problem, which is NP complete.*

Min-Cost Subgraph Selection

- We specify a multicast connection with a triplet (s, T, R),
  1. s is the source of the connection
  2. T is the set of receivers
  3. R is the multicast Rate

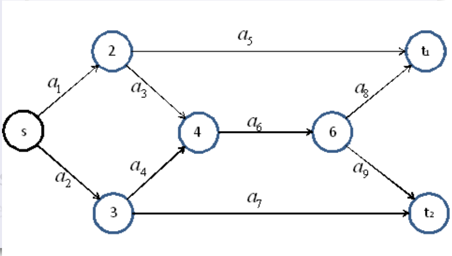


Figure: Butterfly network with multicast from s to t_1 and t_2.

Min-Cost Subgraph Selection

- We specify a multicast connection with a triplet (s, T, R),
  1. s is the source of the connection
  2. T is the set of receivers
  3. R is the multicast Rate

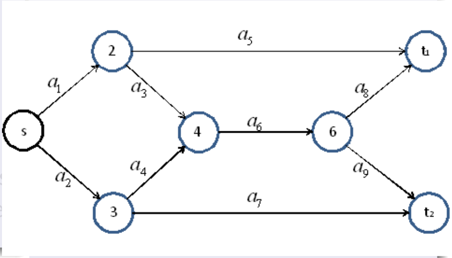


Figure: Butterfly network with multicast from s to t_1 and t_2.

Min-Cost Subgraph Selection

- We specify a multicast connection with a triplet (s, T, R),
  1. s is the source of the connection
  2. T is the set of receivers
  3. R is the multicast Rate

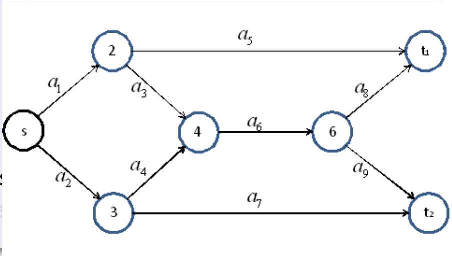


Figure: Butterfly network with multicast from s to t_1 and t_2.

Min-Cost Subgraph Selection

- We specify a multicast connection with a triplet (s, T, R),
  1. s is the source of the connection
  2. T is the set of receivers
  3. R is the multicast Rate

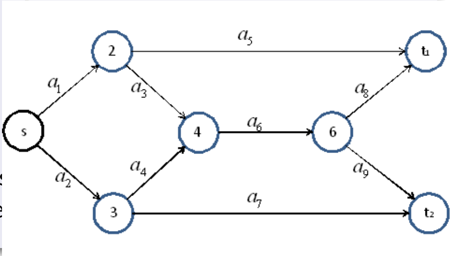
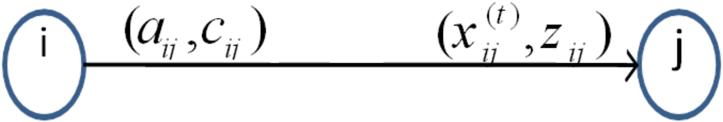


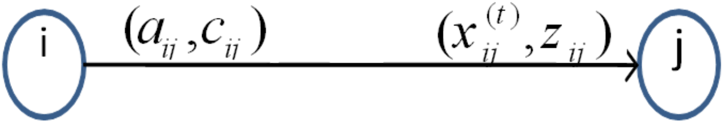
Figure: Butterfly network with multicast from s to t_1 and t_2.

VARIABLE AND PARAMETER NOTATIONS

- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i, j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i, j)
- Parameter notations
  - Cost per unit rate, a_{ij}
  - Capacity, c_{ij}

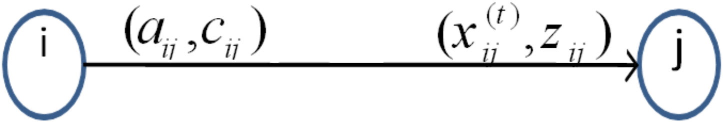


- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i,j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i,j)
- Parameter notations
  1. Cost per unit rate a_{ij}
  2. Capacity c_{ij}



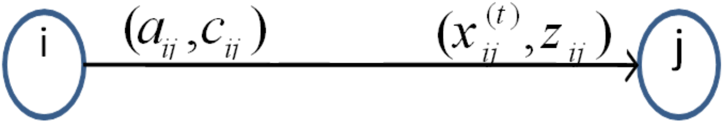
VARIABLE AND PARAMETER NOTATIONS

- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i, j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i, j)
- Parameter notations
  1. Cost per unit rate, a_{ij}
  2. Capacity, c_{ij}

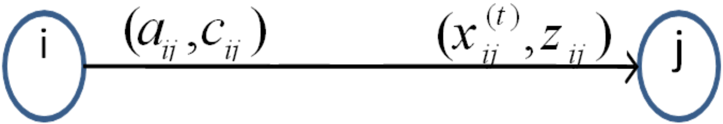


VARIABLE AND PARAMETER NOTATIONS

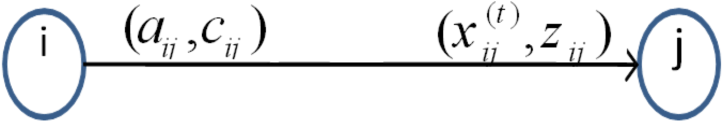
- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i, j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i, j)
- Parameter notations
  1. Cost per unit rate, a_{ij}
  2. Capacity, c_{ij}



- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i, j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i, j)
- Parameter notations
  1. Cost per unit rate, a_{ij}
  2. Capacity, c_{ij}

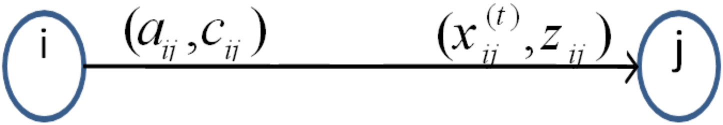


- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i,j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i,j)
- Parameter notations
  1. Cost per unit rate, a_{ij}
  2. Capacity, c_{ij}



VARIABLE AND PARAMETER NOTATIONS

- Variable notations
  1. $x_{ij}^{(t)}$ denotes the flow rate toward receiver t on link (i, j)
  2. z_{ij} denote the rate at which coded packets are injected onto link (i, j)
- Parameter notations
  1. Cost per unit rate, a_{ij}
  2. Capacity, c_{ij}



RELATIONSHIP BETWEEN x AND z

MA. Raayatpanah

- $$z_{ij} = \max_{t \in T}(x_{ij}^{(t)}).$$
- **Subgraph definition:**
  1. The rate vector z, consisting of z_{ij}, $(i,j) \in A$, is called a subgraph.

RELATIONSHIP BETWEEN x AND z

- $$z_{ij} = \max_{t \in T}(x_{ij}^{(t)}).$$
- **Subgraph definition:**
  1. The rate vector z, consisting of z_{ij}, $(i, j) \in A$, is called a subgraph.

RELATIONSHIP BETWEEN x AND z

- $$z_{ij} = \max_{t \in T}(x_{ij}^{(t)}).$$
- **Subgraph definition:**
  1. The rate vector z, consisting of z_{ij}, $(i,j) \in A$, is called a subgraph,

RELATIONSHIP BETWEEN x AND z

- $$z_{ij} = \max_{t \in T}(x_{ij}^{(t)}).$$
- **Subgraph definition:**
  1. The rate vector z, consisting of z_{ij}, $(i, j) \in A$, is called a subgraph,

$$\min \sum_{(i,j)\in A} a_{ij} z_{ij}$$

$$s.t. z_{ij} = \max_{t\in T}(x_{ij}^{(t)}),$$

$$\sum_{\{j|\,(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|\,(j,i)\in A\}} x_{ji}^{(t)} = \begin{cases} R, & \text{i=}s; \\ -R, & \text{i=}t; \\ 0, & \text{otherwi} \end{cases}$$

$$z_{ij} \leq c_{ij},$$

- Total cost
- Coded packet rate
- Conservation constraint
- Capacity constraint

$$\min \sum_{(i,j)\in A} a_{ij} z_{ij}$$

$$s.t. z_{ij} = \max_{t\in T}(x_{ij}^{(t)}),$$

$$\sum_{\{j|\,(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|\,(j,i)\in A\}} x_{ji}^{(t)} = \begin{cases} R, & \text{i=}s; \\ -R, & \text{i=}t; \\ 0, & \text{otherwi} \end{cases}$$

$$z_{ij} \le c_{ij},$$

- Total cost
- Coded packet rate
- Conservation constraint
- Capacity constraint

Minimum-Cost Multicast Over Coded Packet Networks

$$\min \sum_{(i,j)\in A} a_{ij} z_{ij}$$

$$s.t. z_{ij} = \max_{t\in T}(x_{ij}^{(t)}),$$

$$\sum_{\{j|\,(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|\,(j,i)\in A\}} x_{ji}^{(t)} = \begin{cases} R, & \text{i=}s; \\ -R, & \text{i=}t; \\ 0, & \text{otherwi} \end{cases}$$

$$z_{ij} \leq c_{ij},$$

- Total cost
- Coded packet rate
- Conservation constraint
- Capacity constraint

$$\min \sum_{(i,j)\in A} a_{ij} z_{ij}$$

$$s.t. z_{ij} = \max_{t\in T}(x_{ij}^{(t)}),$$

$$\sum_{\{j|\,(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|\,(j,i)\in A\}} x_{ji}^{(t)} = \begin{cases} R, & \text{i=}s; \\ -R, & \text{i=}t; \\ 0, & \text{otherwi} \end{cases}$$

$$z_{ij} \leq c_{ij},$$

- Total cost
- Coded packet rate
- Conservation constraint
- Capacity constraint

$$\min \sum_{(i,j)\in A} a_{ij} z_{ij}$$

$$s.t. z_{ij} = \max_{t\in T}(x_{ij}^{(t)}),$$

$$\sum_{\{j|\,(i,j)\in A\}} x_{ij}^{(t)} - \sum_{\{j|\,(j,i)\in A\}} x_{ji}^{(t)} = \begin{cases} R, & \text{i=}s; \\ -R, & \text{i=}t; \\ 0, & \text{otherwi} \end{cases}$$

$$z_{ij} \leq c_{ij},$$

- Total cost
- Coded packet rate
- Conservation constraint
- Capacity constraint

- **Theorem:**
  There exists a network code flow arbitrarily close to z_{ij} on each link (i, j) for supporting a multicast connection of rate R from source s to T if and only if the min-cut from s to any $t \in T$ is greater than or equal to R, (Proof follows from min-cut max-flow).
- This model can be solved in a
  1. Distributed way (using Lagrangian relaxation)
  2. Polynomial-time

- **Theorem:**
  There exists a network code flow arbitrarily close to z_{ij} on each link (i, j) for supporting a multicast connection of rate R from source s to T if and only if the min-cut from s to any $t \in T$ is greater than or equal to R, (Proof follows from min-cut max-flow).
- This model can be solved in a
  1. Distributed way (using Lagrangian relaxation)
  2. Polynomial time

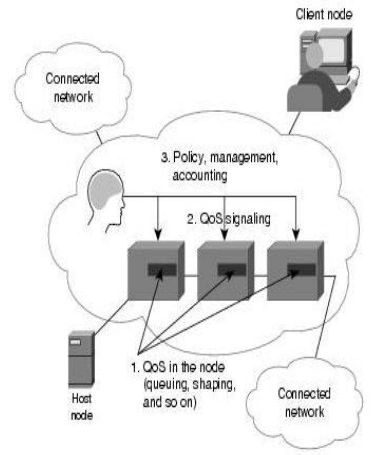
- **Theorem:**
  There exists a network code flow arbitrarily close to z_{ij} on each link (i,j) for supporting a multicast connection of rate R from source s to T if and only if the min-cut from s to any $t \in T$ is greater than or equal to R, (Proof follows from min-cut max-flow).
- This model can be solved in a
  1. Distributed way (using Lagrangian relaxation)
  2. Polynomial-time

- **Theorem:**
  There exists a network code flow arbitrarily close to z_{ij} on each link (i, j) for supporting a multicast connection of rate R from source s to T if and only if the min-cut from s to any $t \in T$ is greater than or equal to R, (Proof follows from min-cut max-flow).
- This model can be solved in a
  1. Distributed way (using Lagrangian relaxation)
  2. Polynomial-time

- **Theorem:**
  There exists a network code flow arbitrarily close to z_{ij} on each link (i, j) for supporting a multicast connection of rate R from source s to T if and only if the min-cut from s to any $t \in T$ is greater than or equal to R, (Proof follows from min-cut max-flow).
- This model can be solved in a
  1. Distributed way (using Lagrangian relaxation)
  2. Polynomial-time

What is QoS?

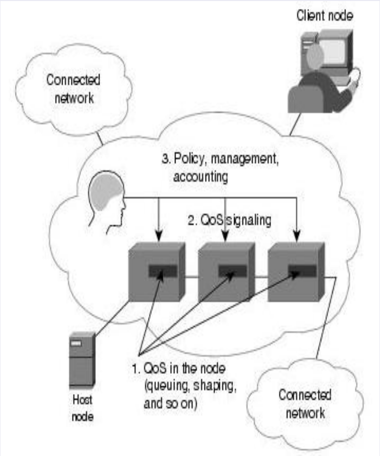
- **Quality of Service** (QoS) is the capability of a network to provide better service
- Without QoS, when you send some packet on the network, the packet can arrive in any order or take an undefined time to arrive



Introduction to QoS

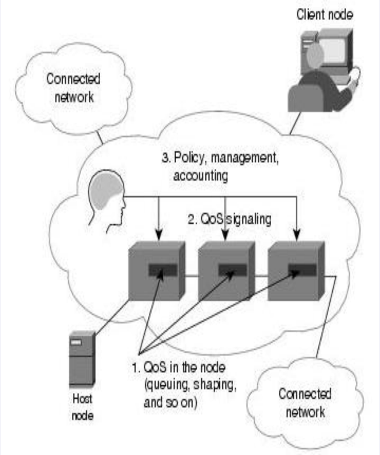
What is QoS?

- **Quality of Service** (QoS) is the capability of a network to provide better service
- Without QoS, when you send some packet on the network, the packet can arrive in any order or take an undefined time to arrive



What is QoS?

- **Quality of Service** (QoS) is the capability of a network to provide better service
- Without QoS, when you send some packet on the network, the packet can arrive in any order or take an undefined time to arrive



Various formal metrics to measure QoS

- **Delay**
  1. The time taken by a packet to travel through the network from one end to another.
- **Delay Jitter**
  1. The variation in the delay encountered by similar packets following the same route through the network.
- **Throughput**
  1. The rate at which packets go through the network.
- **Packet loss rate**
  1. The rate at which packets are dropped, get lost or become corrupted (some bits are changed in the packet) while going through the network.

Various formal metrics to measure QoS

- **Delay**
  1. The time taken by a packet to travel through the network from one end to another.
- **Delay Jitter**
  1. The variation in the delay encountered by similar packets following the same route through the network.
- **Throughput**
  1. The rate at which packets go through the network.
- **Packet loss rate**
  1. The rate at which packets are dropped, get lost or become corrupted (some bits are changed in the packet) while going through the network.

- **Delay**
  1. The time taken by a packet to travel through the network from one end to another.
- **Delay Jitter**
  1. The variation in the delay encountered by similar packets following the same route through the network.
- **Throughput**
  1. The rate at which packets go through the network.
- **Packet loss rate**
  1. The rate at which packets are dropped, get lost or become corrupted (some bits are changed in the packet) while going through the network.

Various formal metrics to measure QoS

- **Delay**
  1. The time taken by a packet to travel through the network from one end to another.
- **Delay Jitter**
  1. The variation in the delay encountered by similar packets following the same route through the network.
- **Throughput**
  1. The rate at which packets go through the network.
- **Packet loss rate**
  1. The rate at which packets are dropped, get lost or become corrupted (some bits are changed in the packet) while going through the network.

NC and QoS

- Nowadays, NC can be able to support multimedia applications like:
  1. Video conferencing,
  2. Audio conferencing,
  3. FTP, HTTP service
- These real-time transactions are sensitive to network characteristics, such as delay, delay variation, bandwidth, and cost,

- Nowadays, NC can be able to support multimedia applications like:
  1. Video conferencing,
  2. Audio conferencing,
  3. FTP, HTTP service
- These real-time transactions are sensitive to network characteristics, such as delay, delay variation, bandwidth, and cost,

NC and QoS

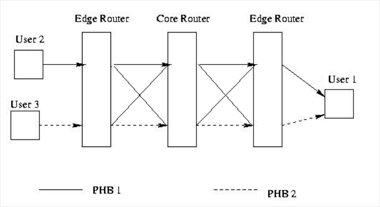
- Nowadays, NC can be able to support multimedia applications like:
  1. Video conferencing,
  2. Audio conferencing,
  3. FTP, HTTP service
- These real-time transactions are sensitive to network characteristics, such as delay, delay variation, bandwidth, and cost,

NC and QoS

- Nowadays, NC can be able to support multimedia applications like:
  1. Video conferencing,
  2. Audio conferencing,
  3. FTP, HTTP service
- These real-time transactions are sensitive to network characteristics, such as delay, delay variation, bandwidth, and cost,

NC and QoS?

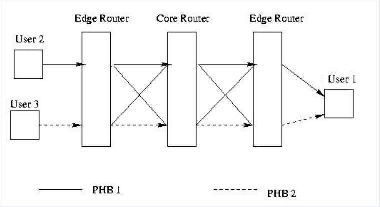
- To avoid breaks in continuity of audio and video playback, it is necessary to
  1. Guarantee end-to-end QoS parameters
  2. keep the overall cost of the solution low.



WHY DO WE NEED QoS

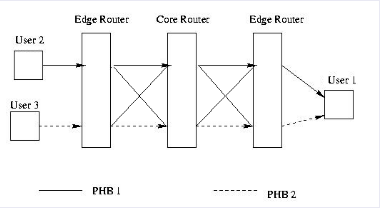
NC and QoS?

- To avoid breaks in continuity of audio and video playback, it is necessary to
  1. Guarantee end-to-end QoS parameters
  2. keep the overall cost of the solution low.



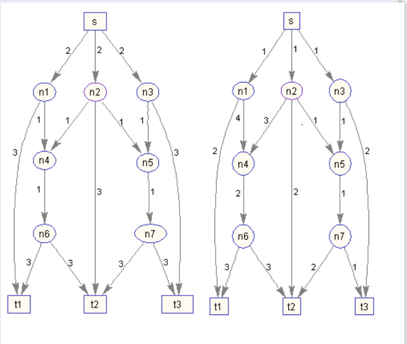
NC and QoS?

- To avoid breaks in continuity of audio and video playback, it is necessary to
    1. Guarantee end-to-end QoS parameters
    2. keep the overall cost of the solution low.

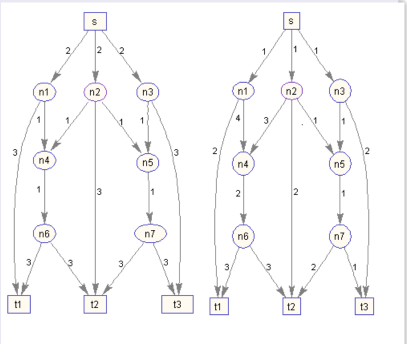


Constrained multicast sub-graph over coded packet networks

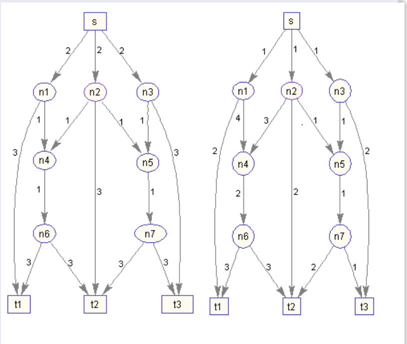
- Consider a single session multicast in a network.
- Each link is marked with its cost per unit rate and weight
- The weight could include delay, jitter, bandwidth, packet delivery ratio, and packet loss ratio.



- Consider a single session multicast in a network.
- Each link is marked with its cost per unit rate and weight
- The weight could include delay, jitter, bandwidth, packet delivery ratio, and packet loss ratio.



- Consider a single session multicast in a network.
- Each link is marked with its cost per unit rate and weight
- The weight could include delay, jitter, bandwidth, packet delivery ratio, and packet loss ratio.



- The problem is to find a subgraph over coded packet networks with
  1. Minimum cost
  2. Satisfying bandwidth constraints.
  3. Longest end-to-end weight from the source to each destination does not exceed an upper bound.

- The problem is to find a subgraph over coded packet networks with
  1. Minimum cost
  2. Satisfying bandwidth constraints.
  3. Longest end-to-end weight from the source to each destination does not exceed an upper bound.

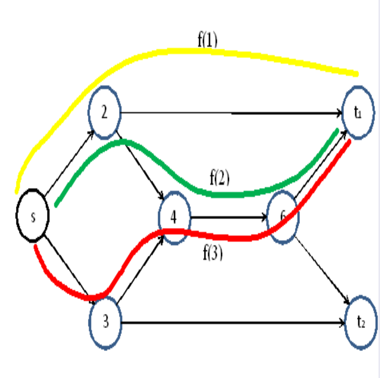
CONSTRAINED SUBGRAPH PROBLEM

- The problem is to find a subgraph over coded packet networks with
  1. Minimum cost
  2. Satisfying bandwidth constraints.
  3. Longest end-to-end weight from the source to each destination does not exceed an upper bound.

- The problem is to find a subgraph over coded packet networks with
  1. Minimum cost
  2. Satisfying bandwidth constraints.
  3. Longest end-to-end weight from the source to each destination does not exceed an upper bound.

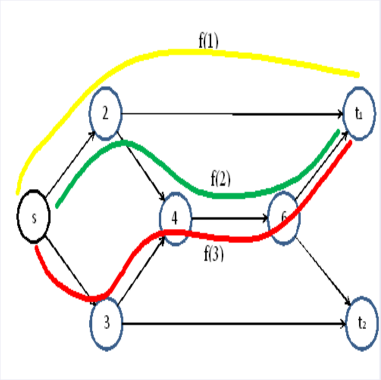
Path-based formulation

- Let $P^{(k)}$ denote the collection of all directed paths from source node s to destination node k in the underlying network G.
- For example, we have three paths from s to t_1: P_1 (Yellow one), P_2 (Green one), P_3 (Red one),
- Define variable $f(p)$ as the flow on path $p \in P^{(k)}$.



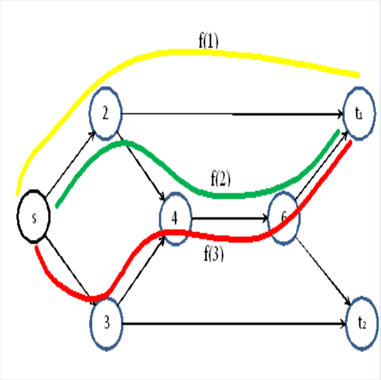
Path-based formulation

- Let $P^{(k)}$ denote the collection of all directed paths from source node s to destination node k in the underlying network G.
- For example, we have three paths from s to t_1: P_1 (Yellow one), P_2 (Green one), P_3 (Red one),
- Define variable $f(p)$ as the flow on path $p \in P^{(k)}$.



Path-based formulation

- Let $P^{(k)}$ denote the collection of all directed paths from source node s to destination node k in the underlying network G.
- For example, we have three paths from s to t_1: P_1 (Yellow one), P_2 (Green one), P_3 (Red one),
- Define variable $f(p)$ as the flow on path $p \in P^{(k)}$.



MA. Raayatpanah

- The weight of path $p \in P^{(k)}$ is defined as follows:

$$W^{(k)}(p) = \sum_{e \in p} w_e. \tag{1}$$

- The following constraint is considered to guarantee the longest end-to-end violation.
-

$$\max_{p \in P^{(k)}} (W^{(k)}(p)) \leq U^{(k)}, \qquad \forall k \in K. \tag{2}$$

- $U^{(k)}$ is an upper bound on the longest end-to-end weight from source node s to destination node k

- The weight of path $p \in P^{(k)}$ is defined as follows:

$$W^{(k)}(p) = \sum_{e \in p} w_e. \tag{1}$$

- The following constraint is considered to guarantee the longest end-to-end violation.

- $$\max_{p \in P^{(k)}} \left(W^{(k)}(p)\right) \le U^{(k)}, \qquad \forall k \in K. \tag{2}$$

- $U^{(k)}$ is an upper bound on the longest end-to-end weight from source node s to destination node k

PATH WEIGHT

- The weight of path $p \in P^{(k)}$ is defined as follows:

$$W^{(k)}(p) = \sum_{e \in p} w_e. \tag{1}$$

- The following constraint is considered to guarantee the longest end-to-end violation.
-

$$\max_{p \in P^{(k)}} (W^{(k)}(p)) \leq U^{(k)}, \qquad \forall k \in K. \tag{2}$$

- $U^{(k)}$ is an upper bound on the longest end-to-end weight from source node s to destination node k

PATH WEIGHT

- The weight of path $p \in P^{(k)}$ is defined as follows:

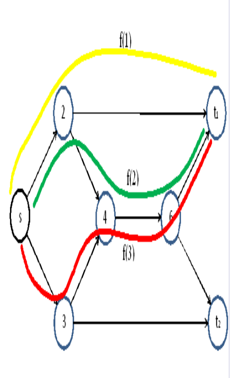
$$W^{(k)}(p) = \sum_{e \in p} w_e. \tag{1}$$

- The following constraint is considered to guarantee the longest end-to-end violation.
- $$\max_{p \in P^{(k)}} (W^{(k)}(p)) \leq U^{(k)}, \qquad \forall k \in K. \tag{2}$$
- $U^{(k)}$ is an upper bound on the longest end-to-end weight from source node s to destination node k

Flow and code rate

- The amount of a link flow, $x_e^{(k)}$, is computed from path flows by the following relation.
- $$x_e^{(k)} = \sum_{p \in P^{(k)}} \delta_e(p) f(p)$$
- For example $x_{6t_1}^{(1)}$ is equal to $f(2) + f(3)$.
- The rate at which coded packets are injected onto link e.

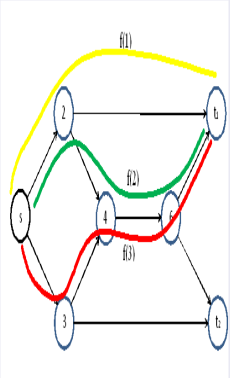
$$z_e = \max_{k \in K}\left(\sum_{p \in P^{(k)}} \delta_e(p) f(p)\right).$$



Flow and code rate

- The amount of a link flow, $x_e^{(k)}$, is computed from path flows by the following relation.
- $$x_e^{(k)} = \sum_{p \in P^{(k)}} \delta_e(p) f(p)$$
- For example $x_{6t_1}^{(1)}$ is equal to $f(2) + f(3)$.
- The rate at which coded packets are injected onto link e.

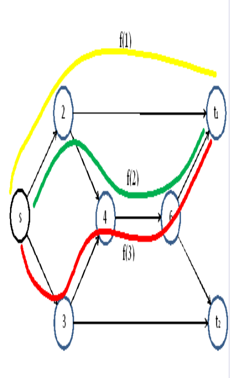
$$z_e = \max_{k \in K}\Big(\sum_{p \in P^{(k)}} \delta_e(p) f(p) \Big).$$



Flow and code rate

- The amount of a link flow, $x_e^{(k)}$, is computed from path flows by the following relation.
-
$$x_e^{(k)} = \sum_{p \in P^{(k)}} \delta_e(p) f(p)$$
- For example $x_{6t_1}^{(1)}$ is equal to $f(2) + f(3)$.
- The rate at which coded packets are injected onto link e.

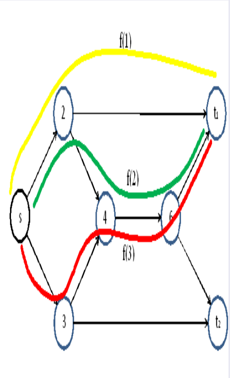
$$z_e = \max_{k \in K} (\sum_{p \in P^(k)} \delta_e(p) f(p)).$$



Flow and code rate

- The amount of a link flow, $x_e^{(k)}$, is computed from path flows by the following relation.
- $$x_e^{(k)} = \sum_{p \in P^{(k)}} \delta_e(p) f(p)$$
- For example $x_{6t_1}^{(1)}$ is equal to $f(2) + f(3)$.
- The rate at which coded packets are injected onto link e.

$$z_e = \max_{k \in K}\Big(\sum_{p \in P^(k)} \delta_e(p) f(p) \Big).$$



$$
\begin{aligned}
&\min \sum_{e\in E} c_e z_e \\
&s.t. \sum_{p\in P^{(k)}} f(p) = R, && \forall k \in K,\\
&z_e = \max_{k\in K}\Big(\sum_{p\in P^{(k)}} \delta_e(p) f(p)\Big), && \forall e \in E,\\
&0 \le z_e \le u_e, && \forall e \in E,\\
&\max_{p\in P^{(k)}} \big(W^{(k)}(p)\big) \le U^{(k)}, && \forall k \in K.\\
&W^{(k)}(p) = \sum_{e\in p} w_e, && \forall p \in P^{(k)},\\
&0 \le f(p), && \forall p \in P^{(k)}.
\end{aligned}
$$

- Minimizes total cost
- Flow conservation
- Coded packet Rate
- Capacity constraint
- End-to-end weight

$$
\begin{aligned}
& \min \sum_{e \in E} c_e z_e \\
& s.t. \sum_{p \in P^{(k)}} f(p) = R, && \forall k \in K, \\
& z_e = \max_{k \in K}\Big(\sum_{p \in P^{(k)}} \delta_e(p) f(p)\Big), && \forall e \in E, \\
& 0 \le z_e \le u_e, && \forall e \in E, \\
& \max_{p \in P^{(k)}} \big(W^{(k)}(p)\big) \le U^{(k)}, && \forall k \in K. \\
& W^{(k)}(p) = \sum_{e \in p} w_e, && \forall p \in P^{(k)}, \\
& 0 \le f(p), && \forall p \in P^{(k)}.
\end{aligned}
$$

- Minimizes total cost
- Flow conservation
- Coded packet Rate
- Capacity constraint
- $\Big\}$ End-to-end weight

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e z_e \\
s.t. \quad & \sum_{p \in P^{(k)}} f(p) = R, && \forall k \in K, \\
& z_e = \max_{k \in K}\Big(\sum_{p \in P^{(k)}} \delta_e(p) f(p)\Big), && \forall e \in E, \\
& 0 \le z_e \le u_e, && \forall e \in E, \\
& \max_{p \in P^{(k)}} \left(W^{(k)}(p)\right) \le U^{(k)}, && \forall k \in K. \\
& W^{(k)}(p) = \sum_{e \in p} w_e, && \forall p \in P^{(k)}, \\
& 0 \le f(p), && \forall p \in P^{(k)}.
\end{aligned}
$$

- Minimizes total cost
- Flow conservation
- Coded packet Rate
- Capacity constraint
- $\Big\}$ End-to-end weight

Path-based formulation for the problem of finding an constrained multicast sub-graph

$$
\begin{aligned}
&\min \sum_{e\in E} c_e z_e \\
&s.t. \sum_{p\in P^{(k)}} f(p) = R, && \forall k \in K, \\
&z_e = \max_{k\in K}\Big(\sum_{p\in P^{(k)}} \delta_e(p) f(p)\Big), && \forall e \in E, \\
&0 \le z_e \le u_e, && \forall e \in E, \\
&\max_{p\in P^{(k)}} (W^{(k)}(p)) \le U^{(k)}, && \forall k \in K. \\
&W^{(k)}(p) = \sum_{e\in p} w_e, && \forall p \in P^{(k)}, \\
&0 \le f(p), && \forall p \in P^{(k)}.
\end{aligned}
$$

- Minimizes total cost
- Flow conservation
- Coded packet Rate
- Capacity constraint
- End-to-end weight

Path-based formulation for the problem of finding an constrained multicast sub-graph

$$
\begin{aligned}
\min\ & \sum_{e\in E} c_e z_e \\
s.t.\ & \sum_{p\in P^{(k)}} f(p) = R, && \forall k\in K,\\
& z_e = \max_{k\in K}\Big(\sum_{p\in P^{(k)}} \delta_e(p) f(p)\Big), && \forall e\in E,\\
& 0 \le z_e \le u_e, && \forall e\in E,\\
& \max_{p\in P^{(k)}} (W^{(k)}(p)) \le U^{(k)}, && \forall k\in K.\\
& W^{(k)}(p) = \sum_{e\in p} w_e, && \forall p\in P^{(k)},\\
& 0 \le f(p), && \forall p\in P^{(k)}.
\end{aligned}
$$

- Minimizes total cost
- Flow conservation
- Coded packet Rate
- Capacity constraint
- End-to-end weight

$$\min \sum_{e \in E} c_e z_e$$

$$s.t. \sum_{p \in P^{(k)}} f(p) = R, \qquad \forall k \in K,$$

$$z_e = \max_{k \in K}\Big(\sum_{p \in P^{(k)}} \delta_e(p) f(p)\Big), \qquad \forall e \in E,$$

$$0 \leq z_e \leq u_e, \qquad \forall e \in E,$$

$$\max_{p \in P^{(k)}} (W^{(k)}(p)) \leq U^{(k)}, \qquad \forall k \in K.$$

$$W^{(k)}(p) = \sum_{e \in p} w_e, \qquad \forall p \in P^{(k)},$$

$$0 \leq f(p), \qquad \forall p \in P^{(k)}.$$

- Minimizes total cost
- Flow conservation
- Coded packet Rate
- Capacity constraint
- $\Big\}$ End-to-end weight

- This model can be converted into a mixed-integer linear programming problem.
- The problem is also NP-hard. Because a constrained shortest path problem can be reduced to it.
- The problem can be solved in a distributed method
  1. The proposed algorithm include
  2. Column generation method to find upper bounds on the optimum objective value
  3. Relaxation method to find lower bounds on the optimum objective value

- This model can be converted into a mixed-integer linear programming problem.
- The problem is also NP-hard. Because a constrained shortest path problem can be reduced to it.
- The problem can be solved in a distributed method
  1. The proposed algorithm include:
  2. Column generation method to find upper bounds on the optimum objective value
  3. Relaxation method to find lower bounds on the optimal objective value

COMPLEXITY ANALYSIS AND THE PROPOSED METHOD

- This model can be converted into a mixed-integer linear programming problem.
- The problem is also NP-hard. Because a constrained shortest path problem can be reduced to it.
- The problem can be solved in a distributed method
  1. The proposed algorithm include:
  2. Column generation method to find upper bounds on the optimum objective value
  3. Relaxation method to find lower bounds on the optimum objective value

COMPLEXITY ANALYSIS AND THE PROPOSED METHOD

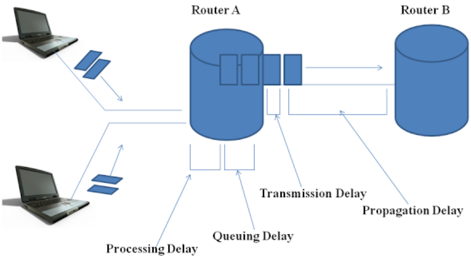
- This model can be converted into a mixed-integer linear programming problem.
- The problem is also NP-hard. Because a constrained shortest path problem can be reduced to it.
- The problem can be solved in a distributed method
  1. The proposed algorithm include:
  2. Column generation method to find upper bounds on the optimum objective value
  3. Relaxation method to find lower bounds on the optimum objective value

- This model can be converted into a mixed-integer linear programming problem.
- The problem is also NP-hard. Because a constrained shortest path problem can be reduced to it.
- The problem can be solved in a distributed method
  1. The proposed algorithm include:
  2. Column generation method to find upper bounds on the optimum objective value
  3. Relaxation method to find lower bounds on the optimum objective value

- This model can be converted into a mixed-integer linear programming problem.
- The problem is also NP-hard. Because a constrained shortest path problem can be reduced to it.
- The problem can be solved in a distributed method
  1. The proposed algorithm include:
  2. Column generation method to find upper bounds on the optimum objective value
  3. Relaxation method to find lower bounds on the optimum objective value

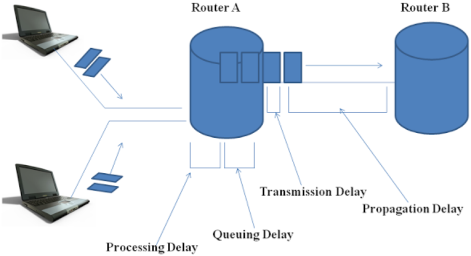
Stochastic Delay

- Delay is one of the most important QoS parameters for real time services,
- In a single multicast session, the delay usually assume a fixed deterministic value.
- In multiple multicast sessions,the delay usually assumed to be **stochastic**,



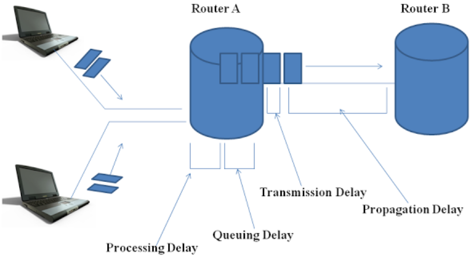
Stochastic Delay

- Delay is one of the most important QoS parameters for real time services,
- In a single multicast session, the delay usually assume a fixed deterministic value.
- In multiple multicast sessions,the delay usually assumed to be **stochastic**,



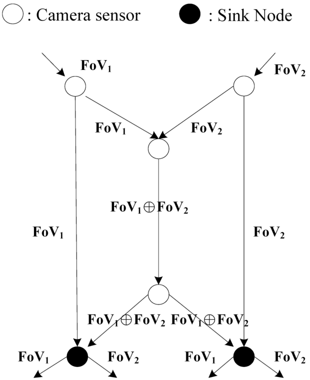
Stochastic Delay

- Delay is one of the most important QoS parameters for real time services,
- In a single multicast session, the delay usually assume a fixed deterministic value.
- In multiple multicast sessions,the delay usually assumed to be **stochastic**,

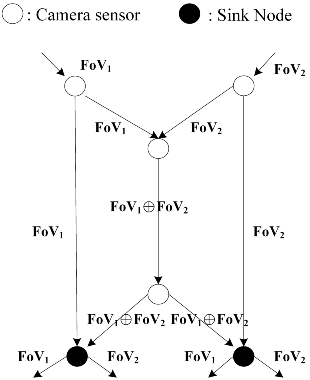


MULTIPLE MULTICAST SESSIONS

- Each session $m \in M$ is identified by the source-destination pair (s_m, T_m, R_m),
  1. s_m is the source node
  2. T_m is the set of receivers of session m.
  3. R_m is multicast rate



- Each session $m \in M$ is identified by the source-destination pair (s_m, T_m, R_m),
  1. s_m is the source node
  2. T_m is the set of receivers of session m.
  3. R_m is multicast rate



Bounds on end-to-end statistical delay constraints

- Assume that the random variable d_e is characterized by
  1. Mean, $\bar{d}_e$,
  2. Variance, σ_e^2
- Let $P^{m,k}$ denote the collection of all directed paths from source node, s^m, to destination node, k, in session m.
- The end-to-end statistical delay of path $p \in P^{m,k}$ is defined as follows:

$$D^{m,k}(p) = \sum_{e \in p} d_e. \qquad (3)$$

Bounds on end-to-end statistical delay constraints

- Assume that the random variable d_e is characterized by
  1. Mean, $\bar{d}_e$,
  2. Variance, σ_e^2
- Let $P^{m,k}$ denote the collection of all directed paths from source node, s^m, to destination node, k, in session m.
- The end-to-end statistical delay of path $p \in P^{m,k}$ is defined as follows:

$$D^{m,k}(p) = \sum_{e \in p} d_e. \tag{3}$$

- Assume that the random variable d_e is characterized by
  1. Mean, $\bar{d}_e$,
  2. Variance, σ_e^2
- Let $P^{m,k}$ denote the collection of all directed paths from source node, s^m, to destination node, k, in session m.
- The end-to-end statistical delay of path $p \in P^{m,k}$ is defined as follows:

$$D^{m,k}(p) = \sum_{e \in p} d_e. \qquad (3)$$

- Assume that the random variable d_e is characterized by
  1. Mean, $\bar{d}_e$,
  2. Variance, σ_e^2
- Let $P^{m,k}$ denote the collection of all directed paths from source node, s^m, to destination node, k, in session m.
- The end-to-end statistical delay of path $p \in P^{m,k}$ is defined as follows:

$$D^{m,k}(p) = \sum_{e \in p} d_e. \tag{3}$$

Bounds on end-to-end statistical delay constraints

- Assume that the random variable d_e is characterized by
  1. Mean, $\bar{d}_e$,
  2. Variance, σ_e^2
- Let $P^{m,k}$ denote the collection of all directed paths from source node, s^m, to destination node, k, in session m.
- The end-to-end statistical delay of path $p \in P^{m,k}$ is defined as follows:

$$D^{m,k}(p) = \sum_{e \in p} d_e. \tag{3}$$

- $D^{m,k}_{max}$ denotes the maximum tolerable delay,
- $\beta^{m,k}$ denotes the violation probability of the delay constraint from source node, s^m, to destination node, k, in session m,
- $$Pr(D^{m,k}(p) \leq D^{m,k}_{max}) = 1 - \beta^{m,k}. \qquad (4)$$

- $D_{max}^{m,k}$ denotes the maximum tolerable delay,
- $\beta^{m,k}$ denotes the violation probability of the delay constraint from source node, s^m, to destination node, k, in session m,
- $$Pr(D^{m,k}(p) \leq D_{max}^{m,k}) = 1 - \beta^{m,k}. \qquad (4)$$

STATISTICAL DELAY CONSTRAINTS

- $D_{max}^{m,k}$ denotes the maximum tolerable delay,
- $\beta^{m,k}$ denotes the violation probability of the delay constraint from source node, s^m, to destination node, k, in session m,
- $$Pr(D^{m,k}(p) \leq D_{max}^{m,k}) = 1 - \beta^{m,k}. \quad (4)$$

- Using Markov's inequality, we have:
-

$$Pr(D^{m,k}(p) \geq D_{max}^{m,k}) \leq \frac{E(D^{m,k}(p))}{D_{max}^{m,k}}, \qquad (5)$$

- $E(D^{m,k}(p)) = \sum_{e \in p} \bar{d}_e$.
- Hence, $Delay(p)$ for path $p \in P^{m,k}$ is defined as follows:

$$\text{Delay}(p) = \begin{cases} \frac{\sum_{e \in p} \bar{d}_e}{D_{max}^{m,k}}, & \text{if } f(p) > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

STATISTICAL DELAY CONSTRAINTS

- Using Markov's inequality, we have:
-
$$Pr(D^{m,k}(p) \geq D^{m,k}_{max}) \leq \frac{E(D^{m,k}(p))}{D^{m,k}_{max}}, \qquad (5)$$
- $E(D^{m,k}(p)) = \sum_{e \in p} \bar{d}_e$.
- Hence, $Delay(p)$ for path $p \in P^{m,k}$ is defined as follows:

$$\text{Delay}(p) = \begin{cases} \frac{\sum_{e \in p} \bar{d}_e}{D^{m,k}_{max}}, & \text{if } f(p) > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

- Using Markov's inequality, we have:
-

$$Pr(D^{m,k}(p) \geq D^{m,k}_{max}) \leq \frac{E(D^{m,k}(p))}{D^{m,k}_{max}}, \qquad (5)$$

- $E(D^{m,k}(p)) = \sum_{e \in p} \bar{d}_e$.
- Hence, $Delay(p)$ for path $p \in P^{m,k}$ is defined as follows:

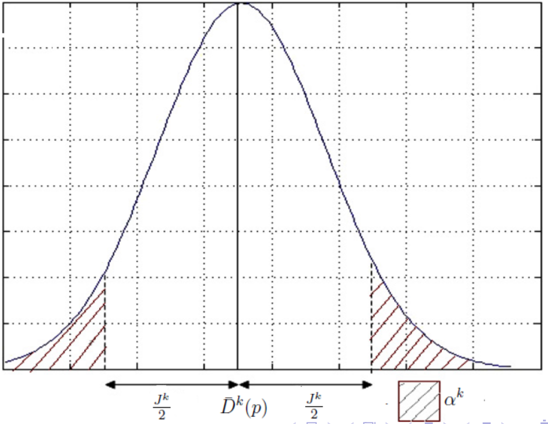
$$\text{Delay}(p) = \begin{cases} \frac{\sum_{e \in p} \bar{d}_e}{D^{m,k}_{max}}, & \text{if } f(p) > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

- Using Markov's inequality, we have:
- $$Pr(D^{m,k}(p) \geq D^{m,k}_{max}) \leq \frac{E(D^{m,k}(p))}{D^{m,k}_{max}}, \qquad (5)$$
- $E(D^{m,k}(p)) = \sum_{e \in p} \bar{d}_e$.
- Hence, $Delay(p)$ for path $p \in P^{m,k}$ is defined as follows:

$$\text{Delay}(p) = \begin{cases} \frac{\sum_{e \in p} \bar{d}_e}{D^{m,k}_{max}}, & \text{if } \text{f(p)} > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

Bounds on end-to-end jitter constraints

- **Jitter** can be defined as the maximum difference between the real-time packet delay and mean delay computed empirically.



Bounds on end-to-end jitter constraints

- The probability that the path, $p \in P^{m,k}$, satisfies the jitter constraint is
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \leq J^{m,k}) = 1 - \alpha^{m,k}$$
- Using Tchebitchev's inequality, we have
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \geq J^{m,k}) \leq \frac{V(D^{m,k}(p))}{(J^{m,k})^2}$$
- where $V(D^{m,k}(p))$ is the end-to-end delay's variance.

- The probability that the path, $p \in P^{m,k}$, satisfies the jitter constraint is
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \leq J^{m,k}) = 1 - \alpha^{m,k}$$
- Using Tchebitchev's inequality, we have
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \geq J^{m,k}) \leq \frac{V(D^{m,k}(p))}{(J^{m,k})^2}$$
- where $V(D^{m,k}(p))$ is the end-to-end delay's variance.

Bounds on end-to-end jitter constraints

- The probability that the path, $p \in P^{m,k}$, satisfies the jitter constraint is
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \leq J^{m,k}) = 1 - \alpha^{m,k}$$
- Using Tchebitchev's inequality, we have
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \geq J^{m,k}) \leq \frac{V(D^{m,k}(p))}{(J^{m,k})^2}$$
- where $V(D^{m,k}(p))$ is the end-to-end delay's variance.

Bounds on end-to-end jitter constraints

- The probability that the path, $p \in P^{m,k}$, satisfies the jitter constraint is
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \leq J^{m,k}) = 1 - \alpha^{m,k}$$
- Using Tchebitchev's inequality, we have
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \geq J^{m,k}) \leq \frac{V(D^{m,k}(p))}{(J^{m,k})^2}$$
- where $V(D^{m,k}(p))$ is the end-to-end delay's variance.

Bounds on end-to-end jitter constraints

- The probability that the path, $p \in P^{m,k}$, satisfies the jitter constraint is
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \leq J^{m,k}) = 1 - \alpha^{m,k}$$
- Using Tchebitchev's inequality, we have
- $$Pr(|D^{m,k}(p) - E(D^{m,k}(p))| \geq J^{m,k}) \leq \frac{V(D^{m,k}(p))}{(J^{m,k})^2}$$
- where $V(D^{m,k}(p))$ is the end-to-end delay's variance.

- With assuming independent delays for each link, we have
- $$V(D^{m,k}(p)) = \sum_{e \in p} \sigma_e^2$$
  .
- $Jitter(p)$ for path $p \in P^{m,k}$ is defined as follows:

$$\text{Jitter}(p) = \begin{cases} \frac{\sum_{e \in p} \sigma_e^2}{(J^{m,k})^2}, & \text{if f(p)} > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

- With assuming independent delays for each link, we have
- $$V(D^{m,k}(p)) = \sum_{e \in p} \sigma_e^2$$
  .
- $Jitter(p)$ for path $p \in P^{m,k}$ is defined as follows:

$$\text{Jitter}(p) = \begin{cases} \frac{\sum_{e \in p} \sigma_e^2}{(J^{m,k})^2}, & \text{if } f(p) > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

JITTER CONSTRAINT

- With assuming independent delays for each link, we have
- $$V(D^{m,k}(p)) = \sum_{e \in p} \sigma_e^2$$
  .
- $Jitter(p)$ for path $p \in P^{m,k}$ is defined as follows:

$$\text{Jitter}(p) = \begin{cases} \frac{\sum_{e \in p} \sigma_e^2}{(J^{m,k})^2}, & \text{if f(p)} > 0, \\ 0, & \text{Otherwise.} \end{cases}$$

RELATIONSHIP BETWEEN x AND z

- Then, the link flow, $x_e^{m,k}$, can be written into the path flows as follows:

$$x_e^{m,k} = \sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p). \tag{6}$$

- Coded packet rate injected on link e for session m is as .

$$z_e^m = \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big),$$

RELATIONSHIP BETWEEN x AND z

- Then, the link flow, $x_e^{m,k}$, can be written into the path flows as follows:

$$x_e^{m,k} = \sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p). \tag{6}$$

- Coded packet rate injected on link e for session m is as .

$$z_e^m = \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big),$$

Path-based formulation

$$\min \sum_{e \in A} \sum_{m \in M} c_e \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big)$$

$$s.t. \quad \sum_{p \in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big),$$

$$0 \leq \sum_{m \in M} z_e^m \leq u_e,$$

$$\max_{p \in P^{m,k}} \{\mathrm{Delay}(p)\} \leq \beta^{m,k},$$

$$\max_{p \in P^{m,k}} \{\mathrm{Jitter}(p)\} \leq \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

Path-based formulation

$$\min \sum_{e \in A} \sum_{m \in M} c_e \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big)$$

$$s.t. \quad \sum_{p \in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big),$$

$$0 \leq \sum_{m \in M} z_e^m \leq u_e,$$

$$\max_{p \in P^{m,k}} \{\mathrm{Delay}(p)\} \leq \beta^{m,k},$$

$$\max_{p \in P^{m,k}} \{\mathrm{Jitter}(p)\} \leq \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

Path-based formulation

$$\min \sum_{e \in A} \sum_{m \in M} c_e \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big)$$

$$s.t. \quad \sum_{p \in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k \in T^m} \Big(\sum_{p \in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big),$$

$$0 \leq \sum_{m \in M} z_e^m \leq u_e,$$

$$\max_{p \in P^{m,k}} \{\text{Delay}(p)\} \leq \beta^{m,k},$$

$$\max_{p \in P^{m,k}} \{\text{Jitter}(p)\} \leq \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

Path-based formulation

$$\min \sum_{e\in A}\sum_{m\in M} c_e \max_{k\in T^m}\Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p)\Big)$$

$$s.t. \quad \sum_{p\in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k\in T^m}\Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p)\Big),$$

$$0 \le \sum_{m\in M} z_e^m \le u_e,$$

$$\max_{p\in P^{m,k}} \{\text{Delay}(p)\} \le \beta^{m,k},$$

$$\max_{p\in P^{m,k}} \{\text{Jitter}(p)\} \le \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

Path-based formulation

$$\min \sum_{e\in A}\sum_{m\in M} c_e \max_{k\in T^m}\Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p)\Big)$$

$$s.t. \quad \sum_{p\in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k\in T^m}\Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p)\Big),$$

$$0 \leq \sum_{m\in M} z_e^m \leq u_e,$$

$$\max_{p\in P^{m,k}} \{\text{Delay}(p)\} \leq \beta^{m,k},$$

$$\max_{p\in P^{m,k}} \{\text{Jitter}(p)\} \leq \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

Path-based formulation

$$\min \sum_{e\in A}\sum_{m\in M} c_e \max_{k\in T^m}\Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p)\Big)$$

$$s.t. \quad \sum_{p\in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k\in T^m}\Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p)\Big),$$

$$0 \le \sum_{m\in M} z_e^m \le u_e,$$

$$\max_{p\in P^{m,k}} \{\text{Delay}(p)\} \le \beta^{m,k},$$

$$\max_{p\in P^{m,k}} \{\text{Jitter}(p)\} \le \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

Path-based formulation

$$\min \sum_{e\in A} \sum_{m\in M} c_e \max_{k\in T^m} \Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big)$$

$$s.t. \quad \sum_{p\in P^{m,k}} f(p) = R^m,$$

$$z_e^m = \max_{k\in T^m} \Big(\sum_{p\in P^{m,k}} \delta_e^{m,k}(p) f(p) \Big),$$

$$0 \le \sum_{m\in M} z_e^m \le u_e,$$

$$\max_{p\in P^{m,k}} \{\text{Delay}(p)\} \le \beta^{m,k},$$

$$\max_{p\in P^{m,k}} \{\text{Jitter}(p)\} \le \alpha^{m,k},$$

- Minimizes the total cost
- Flow conservation constraint
- Coded packet rate
- Capacity constraint
- Delay constraints
- Jitter constraints

- The Model can be rewritten as a mixed-integer linear programming
- The problem is NP-hard. Because, a two-constraint knapsack problem can reduce to it .
- The proposed algorithm is based on a primal and dual decomposition methods.
  1. Primal decomposition method provides an upper bound of the objective value,
  2. Dual decomposition method provides a lower bound of the objective value

COMPLEXITY ANALYSIS AND THE PROPOSED METHOD

- The Model can be rewritten as a mixed-integer linear programming
- The problem is NP-hard. Because, a two-constraint knapsack problem can reduce to it .
- The proposed algorithm is based on a primal and dual decomposition methods.
    1. Primal decomposition method provides an upper bound of the objective value,
    2. Dual decomposition method provides a lower bound of the objective value

- The Model can be rewritten as a mixed-integer linear programming
- The problem is NP-hard. Because, a two-constraint knapsack problem can reduce to it .
- The proposed algorithm is based on a primal and dual decomposition methods.
  1. Primal decomposition method provides an upper bound of the objective value,
  2. Dual decomposition method provides a lower bound of the objective value

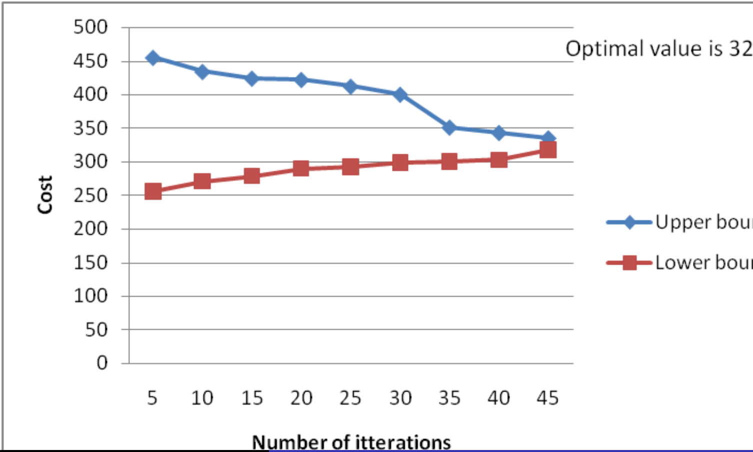
- The Model can be rewritten as a mixed-integer linear programming
- The problem is NP-hard. Because, a two-constraint knapsack problem can reduce to it .
- The proposed algorithm is based on a primal and dual decomposition methods.
  1. Primal decomposition method provides an upper bound of the objective value,
  2. Dual decomposition method provides a lower bound of the objective value

COMPLEXITY ANALYSIS AND THE PROPOSED METHOD

- The Model can be rewritten as a mixed-integer linear programming
- The problem is NP-hard. Because, a two-constraint knapsack problem can reduce to it .
- The proposed algorithm is based on a primal and dual decomposition methods.
    1. Primal decomposition method provides an upper bound of the objective value,
    2. Dual decomposition method provides a lower bound of the objective value

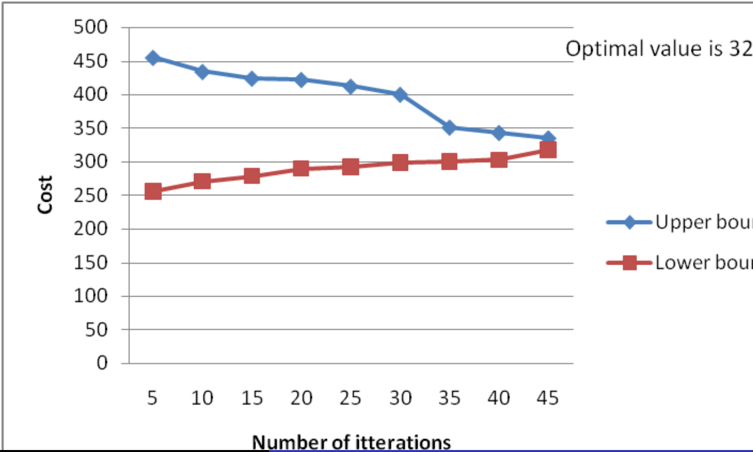
STOP CONDITION

- May stop the algorithm when the two bounds are sufficiently close to each other.



STOP CONDITION

- May stop the algorithm when the two bounds are sufficiently close to each other.



Summery

- Minimum-cost multicast over coded packet networks (Iun et al. 2006)
- Optimal-constrained multicast sub-graph over coded packet networks (Raayatpanah et al. 2013)
- Bounds on end-to-end statistical delay and jitter in multiple multicast coded packet networks (Raayatpanah et al. 2014)
- We can also consider the other real assumption to select subgraph.

Summery

- Minimum-cost multicast over coded packet networks (Lun et al. 2006)
- Optimal-constrained multicast sub-graph over coded packet networks (Raayatpanah et al. 2013)
- Bounds on end-to-end statistical delay and jitter in multiple multicast coded packet networks (Raayatpanah et al. 2014)
- We can also consider the other real assumption to select subgraph.

Summery

- Minimum-cost multicast over coded packet networks (Iun et al. 2006)
- Optimal-constrained multicast sub-graph over coded packet networks (Raayatpanah et al. 2013)
- Bounds on end-to-end statistical delay and jitter in multiple multicast coded packet networks (Raayatpanah et al. 2014)
- We can also consider the other real assumption to select subgraph.

Summery

- Minimum-cost multicast over coded packet networks (Iun et al. 2006)
- Optimal-constrained multicast sub-graph over coded packet networks (Raayatpanah et al. 2013)
- Bounds on end-to-end statistical delay and jitter in multiple multicast coded packet networks (Raayatpanah et al. 2014)
- We can also consider the other real assumption to select subgraph.

SUMMERY

- Minimum-cost multicast over coded packet networks (lun et al. 2006)
- Optimal-constrained multicast sub-graph over coded packet networks (Raayatpanah et al. 2013)
- Bounds on end-to-end statistical delay and jitter in multiple multicast coded packet networks (Raayatpanah et al. 2014)
- We can also consider the other real assumption to select subgraph.

THANK YOU FOR YOUR ATTENTION

ANY QUESTIONS