Capacity Regions for Multiple Unicasts Flows using Inter-session Network Coding

Abdallah Khreishah, Chih-Chun Wang
Center for Wireless Systems and Applications
School of ECE
Purdue University

Ness B. Shroff
Department of Electrical & Computer Engineering
Ohio State University
Single Session

The Routing Solution

\[ s \xrightarrow{X_1} t_2 \xrightarrow{X_2} t_1 \]

\[ X_1, X_2 \]

\[ X_1, X_2, X_2 \]
Single Session

The Routing Solution

Rate: 3/2

$X_1, X_2, X_3 \quad X_2, X_1, X_3$
Single Session

The Routing Solution

The Network Coding Solution

Rate: 3/2
Single Session

The Routing Solution

The Network Coding Solution

Rate: 3/2

Rate: 2
**Theorem 1**  [Ahlswede et al. 00] For a single multicast session, rate $R$ is achievable if for all dest. $t_i$, the min-cut/max-flow $\rho_G(s, t_i)$ between $s$ and $t_i$ satisfies

$$R \leq \rho_G(s, t_i), \ \forall i.$$
Multicast

**Theorem 1** [Ahlswede et al. 00] For a single multicast session, rate $R$ is achievable if for all dest. $t_i$, the min-cut/max-flow $\rho_G(s, t_i)$ between $s$ and $t_i$ satisfies

$$R \leq \rho_G(s, t_i), \forall i.$$ 

**Intra-session Multicast** [Chen et al. 07]

$$\max_{R_i} \sum_i U_i(R_i)$$

subject to \(\sum_i f_{i,e} \leq c_e, \forall e \in E\)

$$\forall i, \{f_{i,e}\}_{e \in E} \text{ and } R_i \text{ satisfy the min-cut max-flow conditions.}$$
Multiple Sessions unicast

- Each source $s_i$ wants to send messages to destination $t_i$ at rate $R_i$. 
Multiple Sessions unicast

- Each source $s_i$ wants to send messages to destination $t_i$ at rate $R_i$.

- Routing solution $\iff$ Each session $i$ takes an exclusive share of the network.
Each source $s_i$ wants to send messages to destination $t_i$ at rate $R_i$.

Routing solution $\iff$ Each session $i$ takes an exclusive share of the network.

One possible formulation

$$\sum_{n \in \Gamma_O(g)} x_n(i) - \sum_{n \in \Gamma_I(g)} x_n(i) = \begin{cases} R_i & g = s_i \\ -R_i & g = t_i \\ 0 & \text{else} \end{cases}$$ (1)

$$\sum_{i=1}^{I} x_n(i) \leq C_n \quad \forall n \in E$$ (2)
Two simple unicasts

The Butterfly

\[ X_1 \rightarrow X_1 + X_2 \rightarrow X_2 \]
The \textit{TRLKM} region

- By [Traskov \textit{et al.} 06]

- Resolves butterfly bottlenecks in the network by introducing virtual flows $p$, $q$, and $r$. 

The TRLKM region

- By [Traskov et al. 06]

- Resolves butterfly bottlenecks in the network by introducing virtual flows $p$, $q$, and $r$. 
The TRLKM region

- Resolves butterfly bottlenecks in the following manner:
The *RSC* Algorithm

- By [Eryilmaz et al. 07]
  Similar to [Ho et al. 06].
- At each link \((n, k)\) compute two weights by queue lengths exchange:
  - \(\rho^*_{(n,k)}[t]\) corresponds to routing.

---

Fig. 2 of [Eryilmaz et al. 07]

\(\text{Remedy Session } f: \quad b_1 \rightarrow c_1\)

\(\text{Remedy Session } g: \quad b_2 \rightarrow c_2\)

\(\text{Multicast Session } (f, g): \quad n \rightarrow (c_1, c_2)\)
The \textit{RSC} Algorithm

- By [Eryilmaz et al. 07]
  Similar to [Ho et al. 06].

- At each link \((n, k)\) compute two weights by queue lengths exchange:
  - \(\rho^*_{(n,k)}[t]\) corresponds to routing.
  - \(\sigma^*_{(n,k)}[t]\) corresponds to inter-session coding.

\textit{Fig. 2 of [Eryilmaz et al. 07]}
The \textit{RSC} Algorithm cont.

- if $\rho_{(n,k)}^*[t] > \sigma_{(n,k)}^*[t]$ perform routing, otherwise do intersession coding.
The \textit{RSC} Algorithm cont.

- if $\rho_{(n,k)}^*[t] > \sigma_{(n,k)}^*[t]$ perform routing, otherwise do intersession coding.

- The backlog algorithm can \textbf{distributively stabilize} any rates in the $\mathcal{TRLKM}$ region.
The $RSC$ Algorithm cont.

- if $\rho_{(n,k)}^*[t] > \sigma_{(n,k)}^*[t]$ perform routing, otherwise do intersession coding.

- The backlog algorithm can distributively stabilize any rates in the $TRLKM$ region.

- Drawbacks:
The **RSC** Algorithm cont.

- if $\rho_{(n,k)[t]}^* > \sigma_{(n,k)[t]}^*$ perform routing, otherwise do intersession coding.

- The **backlog algorithm** can **distributively stabilize** any rates in the $\mathcal{TRLKM}$ region.

- **Drawbacks:**
  - High Complexity policy.
The $RSC$ Algorithm cont.

- if $\rho^*_{(n,k)[t]} > \sigma^*_{(n,k)[t]}$ perform routing, otherwise do intersession coding.

- The backlog algorithm can **distributively stabilize** any rates in the $\mathcal{T\mathcal{R}\mathcal{L}\mathcal{K}\mathcal{M}}$ region.

- Drawbacks:
  - High Complexity policy.
  - Coding is dependent on queuing
The \textit{RSC} Algorithm cont.

- if $\rho_{(n,k)[t]}^* > \sigma_{(n,k)[t]}^*$ perform routing, otherwise do intersession coding.

- The \textbf{backlog algorithm} can \textit{distributively stabilize} any rates in the \textit{TRLKM} region.

- **Drawbacks:**
  - High Complexity policy.
  - Coding is dependent on queuing
  - No rate control mechanism
The \textit{RSC} Algorithm cont.

- if $\rho_{(n,k)}^*[t] > \sigma_{(n,k)}^*[t]$ perform routing, otherwise do intersession coding.

- The \textbf{backlog algorithm} can \textit{distributively stabilize} any rates in the $\mathcal{TRLKM}$ region.

- **Drawbacks:**
  - High Complexity policy.
  - Coding is dependent on queuing
  - No rate control mechanism
  - Considers only butterfly coding opportunities.
Two simple unicasts

The Grail

\[ X_1 + X_2 \]

\[ X_1 \]

\[ X_1 \]

\[ X_2 \]

\[ t_2 \]

\[ t_1 \]
The \( I \) – \( TRLKM \) region

- Resolves butterfly and grail bottlenecks in the network by introducing virtual flows \( p, q, r, \) and \( l \).
The $\mathcal{I} - TRLKM$ region

- Resolves butterfly and grail bottlenecks in the network by introducing virtual flows $p$, $q$, $r$, and $l$. 

```
\begin{align*}
q(1 \rightarrow 2) &= 1 \\
l(1 \rightarrow 2) &= 1 \\
p(1 \rightarrow 2) &= 1 \\
l(2 \rightarrow 1) &= 1 \\
p(2 \rightarrow 1) &= 1 \\
l(1 \rightarrow 2) &= 1 \\
p(2 \rightarrow 1) &= 1 \\
l(1 \rightarrow 2) &= 1 \\
r(1 \rightarrow 2) &= 1 \\
r(2 \rightarrow 1) &= 1 \\
r(2 \rightarrow 1) &= 1
\end{align*}
```
The \( I - TRLKM \) region

- Resolves grail bottlenecks in the following manner:
The $\mathcal{I} - \mathcal{RSC}$ Algorithm

- Compute three weights based on queue length exchange:
  - $\rho^*_{(n,k)}$ corresponds to routing.
  - $\sigma^*_{(n,k)}[t]$ corresponds to butterfly inter-session coding.
  - $\sigma^*_{1(n,k)}[t]$ corresponds to grail inter-session coding.

- If $\rho^*_{(n,k)}[t] = \max\{\rho^*_{(n,k)}[t], \sigma^*_{(n,k)}[t], \sigma^*_{1(n,k)}[t]\}$ perform routing.

- If $\sigma^*_{(n,k)} = \max\{\rho^*_{(n,k)}, \sigma^*_{(n,k)}[t], \sigma^*_{1(n,k)}[t]\}$ perform butterfly net coding.

- If $\sigma^*_{1(n,k)} = \max\{\rho^*_{(n,k)}, \sigma^*_{(n,k)}[t], \sigma^*_{1(n,k)}[t]\}$ perform grail net coding.
So Far

- Structure based capacity regions.
So Far

- Structure based capacity regions.
  - Butterfly.
So Far

- Structure based capacity regions.
  - Butterfly.
  - Grail
So Far

- Structure based capacity regions.
  - Butterfly.
  - Grail
- The routing capacity region is path-based
So Far

- Structure based capacity regions.
  - Butterfly.
  - Grail

- The routing capacity region is path-based

- **Complexity** issue for centralized and Backlog algorithms for structure based capacity regions.
So Far

- Structure based capacity regions.
  - Butterfly.
  - Grail

- The routing capacity region is path-based

- **Complexity** issue for centralized and Backlog algorithms for structure based capacity regions.

- No rate control or utility maximization in the backlog algorithms.
So Far

- Structure based capacity regions.
  - Butterfly.
  - Grail
- The routing capacity region is path-based
- **Complexity** issue for centralized and Backlog algorithms for structure based capacity regions.
- No rate control or utility maximization in the backlog algorithms.
- Coding is dependent on Queueing.
So Far

- Structure based capacity regions.
  - Butterfly.
  - Grail

- The routing capacity region is path-based

- **Complexity** issue for centralized and Backlog algorithms for structure based capacity regions.

- No rate control or utility maximization in the backlog algorithms.

- **Coding** is dependent on Queueing.

- Try path-based regions using inter-session network coding.
Setting: General finite directed acyclic graphs, unit edge capacity, \((s_1, t_1) \& (s_2, t_2)\), two integer symbols \(X_1\) and \(X_2\).

Number of Coinciding Paths of edge \(e\): \(\mathcal{P} = \{P_1, \ldots, P_k\}\), and \(\text{ncp}_\mathcal{P}(e) = |\{P \in \mathcal{P} : e \in P\}|\).
Preliminaries — 2 Unicasts

- Setting: General finite directed acyclic graphs, unit edge capacity, \((s_1, t_1) \& (s_2, t_2)\), two integer symbols \(X_1\) and \(X_2\).

- Number of Coinciding Paths of edge \(e\): \(\mathcal{P} = \{P_1, \cdots, P_k\}\), and \(\text{ncp}_\mathcal{P}(e) = |\{P \in \mathcal{P} : e \in P\}|\).

**Theorem 2** Network coding \(\iff\) one of the following two holds.

1. \(\exists \mathcal{P} = \{P_{s_1,t_1}, P_{s_2,t_2}\}, \text{ such that } \max_{e \in E} \text{ncp}_\mathcal{P}(e) \leq 1\).

2. \(\exists \mathcal{P} = \{P_{s_1,t_1}, P_{s_2,t_2}, P_{s_2,t_1}\} \text{ and } \mathcal{Q} = \{Q_{s_1,t_1}, Q_{s_2,t_2}, Q_{s_1,t_2}\} \text{ s.t. } \max_{e \in E} \text{ncp}_\mathcal{P}(e) \leq 2 \text{ and } \max_{e \in E} \text{ncp}_\mathcal{Q}(e) \leq 2\).
The \( \mathcal{WS} \) Region

Represent the network \( G \) as a superposition of one \( G_r \) and finitely many \( G_p \) such that:

-
The $\mathcal{WS}$ Region

Represent the network $G$ as a superposition of one $G_r$ and finitely many $G_p$ such that:

- Routing is supported at $G_r$. 

The \( \mathcal{WS} \) Region

Represent the network \( G \) as a superposition of one \( G_r \) and finitely many \( G_p \) such that:

- Routing is supported at \( G_r \).
- Pairwise network coding is supported between two sessions on every \( G_p \).
Formulation

$I$: the no. coexisting unicast sessions \((s_i, t_i)\)

\(\mathcal{P}(i)\): the set of all \((s_i, t_i)\) paths

\(\mathbb{P}(i, j)\): the set of all \((P_{s_i, t_i}, P_{s_j, t_i}, P_{s_j, t_j})\) tuples

\(E_{e,i}^k\): = 1, if link \(e\) uses the \(k\)-th path in \(\mathcal{P}(i)\)

= 0, otherwise

\(H_{e,ij}^l\): = 2, if for the \(l\)-th tuple in \(\mathbb{P}(i, j)\), \(\text{ncp}(e) = 3\)

= 1, if for the \(l\)-th tuple in \(\mathbb{P}(i, j)\), \(\text{ncp}(e) = 1, 2\)

= 0, if for the \(l\)-th tuple in \(\mathbb{P}(i, j)\), \(\text{ncp}(e) = 0\)

\(x_i^k\): the routing rate through the \(k\)-th path of session \(i\).

\(g_{ij}^{lm}\): joint coding rate between session \(i\) and \(j\).
Formulation cont.

\[
\max_{\vec{x}, \vec{g}} \sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mathcal{P}(i)} x_i^k + \sum_{i \neq j} \sum_{l=1}^{\mathcal{P}(j)} \sum_{m=1}^{\mathcal{P}(i)} g_{ij}^{lm} \right)
\]

s.t.
\[
\sum_{i=1}^{I} \mathcal{P}(i) |E_{e,i}^k x_i^k| + \sum_{i=1}^{I} \sum_{i<j} \sum_{l=1}^{\mathcal{P}(i)} \sum_{m=1}^{\mathcal{P}(j)} \max(H_{e,ij}^l, H_{e,ji}^m) g_{ij}^{lm} \leq C_e, \forall e
\]

\[
x_i^k \geq 0, \quad g_{ij}^{lm} = g_{ji}^{ml} \geq 0, \quad \forall i \neq j, l, m
\]
Incorporating the Proximal Meth.

\[ \sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mathcal{P}(i)} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} g_{ij}^{lm} \right) \] may not be strictly concave.
Incorporating the Proximal Meth.

- $\sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mid \mathcal{P}(i) \mid} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\mid \mathcal{P}(i,j) \mid} \sum_{m=1}^{\mid \mathcal{P}(j,i) \mid} g_{ij}^{lm} \right)$ may not be strictly concave.

- The proximal method with auxiliary var. $\overrightarrow{y}, \overrightarrow{h}$:

$$\max_{\{\overrightarrow{x}, \overrightarrow{g}\}} \sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mid \mathcal{P}(i) \mid} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\mid \mathcal{P}(i,j) \mid} \sum_{m=1}^{\mid \mathcal{P}(j,i) \mid} g_{ij}^{lm} \right)$$

$$- \sum_{i=1}^{I} \sum_{k} \frac{C_i}{2} (x_i^k - y_i^k)^2 - \sum_{i \neq j} \sum_{l=1}^{\mid \mathcal{P}(i,j) \mid} \sum_{m=1}^{\mid \mathcal{P}(j,i) \mid} \frac{d_i}{2} (g_{ij}^{lm} - h_{ij}^{lm})^2$$
Incorporating the Proximal Meth.

- \[ \sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mathcal{P}(i)} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} g_{ij}^{lm} \right) \] may not be strictly concave.

- The proximal method with auxiliary var. \( \vec{y}, \vec{h} \):

\[
\max_{\{\vec{x}, \vec{g}\}} \sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mathcal{P}(i)} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} g_{ij}^{lm} \right) - \sum_{i=1}^{I} \sum_{k} \frac{c_i}{2} (x_i^k - y_i^k)^2 - \sum_{i \neq j} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} \frac{d_i}{2} (g_{ij}^{lm} - h_{ij}^{lm})^2
\]

- The Slater condition holds.

- Solve the dual of the intermediate problem.
The Proximal Method (Cont’d)

- The Lagrangian $L_{\vec{y}}(\vec{x}', \vec{g}', \vec{\lambda}', \vec{\mu}')$ is

$$
\sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mid P(i) \mid} x_i^k + \sum_{j \neq i}^{\mid P(i,j) \mid} \sum_{l=1}^{\mid P(j,i) \mid} \sum_{m=1}^{\mid P(i,j) \mid} g_{ij}^{lm} \right)

- \sum_{i=1}^{I} \sum_{k} \frac{c_i}{2} (x_i^k - y_i^k)^2

- \sum_{i=1}^{I} \sum_{j \neq i}^{\mid P(i,j) \mid} \sum_{l=1}^{\mid P(j,i) \mid} \sum_{m=1}^{\mid P(i,j) \mid} d_{ij} (g_{ij}^{lm} - h_{ij}^{lm})^2

- \sum_{e} \lambda_e \left( \sum_{i=1}^{I} \sum_{k=1}^{\mid P(i) \mid} E_{e,i}^k x_i^k + \sum_{i=1}^{I} \sum_{i<j}^{\mid P(i,j) \mid} \sum_{l=1}^{\mid P(j,i) \mid} \sum_{m=1}^{\mid P(i,j) \mid} \max(H_{e,ij}^l, H_{e,ji}^m) g_{ij}^{lm} - C_e \right)

- \sum_{i=1}^{I} \sum_{i<j} \sum_{l} \sum_{m} \mu_{ij}^{lm} (g_{ij}^{lm} - g_{ji}^{ml})
The Proximal Method (Cont’d)

- The Lagrangian $L_{\overrightarrow{y}, \overrightarrow{h}}(\overrightarrow{x}', \overrightarrow{g}', \overrightarrow{\lambda}', \overrightarrow{\mu}')$ is

$$\sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\mathcal{P}(i)} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} g_{ij}^{lm} \right) - \sum_{i=1}^{I} \sum_{k} \frac{c_i}{2} (x_i^k - y_i^k)^2 - \sum_{i=1}^{I} \sum_{j \neq i} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} \frac{d_i}{2} (g_{ij}^{lm} - h_{ij}^{lm})^2$$

$$- \sum_{e} \lambda_e \left( \sum_{i=1}^{I} \sum_{k=1}^{\mathcal{P}(i)} E_{e,i}^k x_i^k + \sum_{i=1}^{I} \sum_{i<j} \sum_{l=1}^{\mathcal{P}(i,j)} \sum_{m=1}^{\mathcal{P}(j,i)} \max(H_{e,ij}^l, H_{e,ji}^m) g_{ij}^{lm} - C_e \right)$$

$$- \sum_{i=1}^{I} \sum_{i<j} \sum_{l} \sum_{m} \mu_{ij}^{lm} (g_{ij}^{lm} - g_{ji}^{ml})$$

- Separable!
The Distributed Solver

- Repeat the following $K$ times:
  - Solve $D_{\overrightarrow{y}, \overrightarrow{h}}(\overrightarrow{\lambda}, \overrightarrow{\mu}) = \max_{\overrightarrow{x}, \overrightarrow{g}} L_{\overrightarrow{y}, \overrightarrow{h}}(\overrightarrow{x}, \overrightarrow{g}, \overrightarrow{\lambda}, \overrightarrow{\mu})$ via separability.
  - Solve the dual problem $\min D_{\overrightarrow{y}, \overrightarrow{h}}(\overrightarrow{\lambda}, \overrightarrow{\mu})$ by the gradient method with step size $\alpha$.
  - Update $\overrightarrow{y} \leftarrow \overrightarrow{x}^\star$, $\overrightarrow{h} \leftarrow \overrightarrow{g}^\star$, and go back to the beginning.
Algo A Summary

- Source Algorithm:

\[
\{ \overrightarrow{x}(t, r), \overrightarrow{g}(t, r) \} = \arg \max_{\{\overrightarrow{x}, \overrightarrow{g}\} \geq 0} \]

\[
U_i(\sum_{k=1}^{\vert \mathcal{P}(i) \vert} x_i^k + \sum_{i \neq j} \sum_{l=1}^{\vert \mathcal{P}(i,j) \vert} \sum_{m=1}^{\vert \mathcal{P}(j,i) \vert} g_{ij}^{lm}) - \sum_{k} \frac{c_i}{2} (x_i^k - y_i^k)^2
\]

\[
- \sum_{i \neq j} \sum_{l=1}^{\vert \mathcal{P}(i,j) \vert} \sum_{m=1}^{\vert \mathcal{P}(j,j) \vert} \frac{d_i}{2} (g_{ij}^{lm} - h_{ij}^{lm})^2 - \sum_{k} (\sum_{e} E_{e,i}^k \lambda_e) x_i^k
\]

\[
- \sum_{i \neq j} \sum_{l} \sum_{m} (\sum_{e} \max(H_e^{i,j}, H_e^{j,i}) \lambda_e) g_{ij}^{lm} - \sum_{i < j} \sum_{l} \sum_{m} \mu_{ij}^{lm} g_{ij}^{lm}
\]

\[
+ \sum_{i > j} \sum_{l} \sum_{m} \mu_{ji}^{lm} g_{ij}^{lm}
\]
Algo A Summary

- **Link Algorithm:**

\[
\lambda_e(t, r + 1) = [\lambda_e(t, r) + \alpha_e \left( \sum_{i=1}^{I} \sum_{k=1}^{|P(i)|} E_{e,i}^k x_i^k(t, r) + \right) + \sum_{i=1}^{I} \sum_{l=1}^{\lfloor P(i,j) \rfloor \lfloor P(j,i) \rfloor} \sum_{l=1}^{\lfloor P(i,j) \rfloor \lfloor P(j,i) \rfloor} \sum_{m=1}^{\lfloor P(i,j) \rfloor \lfloor P(j,i) \rfloor} \max(H_{e,ij}^l, H_{e,ji}^m) g_{ij}^l(t, r) - C_e ]^+.
\]

- **Sink Algorithm:**

\[
\mu_{ij}^{lm}(t, r+1) = \mu_{ij}^{lm}(t, r) + \beta_{ij}^{lm} (g_{ij}^{lm}(t, r) - g_{ji}^{ml}(t, r)) \quad \forall i < j.
\]
The Convergence Result

**Theorem 3**  If the step size $\alpha$ of the gradient method (for the dual) and the proximal method coefficients $c_i$ and $d_i$ satisfy the following:

\[
\alpha \left( 2 + \sum_{i=1}^{I} \sum_{k=1}^{P(i)} E_{e,i}^k + \frac{1}{4} \sum_{i=1}^{I} \sum_{i\neq j}^{I} \sum_{l=1}^{P(i,j)} \sum_{m=1}^{P(j,i)} (\max(H_{e,ij}^l, H_{e,ji}^m))^2 \right) < 2 \min_{i} \min(c_i, d_i),
\]

then as $K \to \infty$, the proximal method converges to the optimal $\vec{x}_{\text{opt}}$ and $\vec{g}_{\text{opt}}$ for the original problem.

- For bounded $K$, the convergence is verified by simulations.
- It can also be proved similar to that in [Lin and Shroff 06].
The Coding Scheme

- Rate control is achieved via distributed algorithms.
- Coding scheme?
The Coding Scheme

- Rate control is achieved via distributed algorithms.
- Coding scheme?

\[ \begin{align*}
  s_1 & \xrightarrow{X_1} t_2 \\
  s_2 & \xrightarrow{X_2} t_1 \\
  X_1 + X_2 & \xrightarrow{X_1 + X_2} \\
  X_1 & \xrightarrow{X_1} \\
  X_2 & \xrightarrow{X_2} \\
  X_1 + X_2 & \xrightarrow{X_1 + X_2} \\
  t_1 & \xrightarrow{X_1} \\
  t_2 & \xrightarrow{X_2} 
\end{align*} \]
The Coding Scheme

- Rate control is achieved via distributed algorithms.
- Coding scheme? Modified random linear coding.
The Coding Scheme

- Rate control is achieved via distributed algorithms.
- Coding scheme? **Modified random linear coding.**

![Diagram of the coding scheme]

**Theorem 4** With modified random linear coding over $\text{GF}(q)$, the success probability is

$$\text{Prob}(\text{success}) \geq \left(1 - \frac{4}{q}\right)^{6|E|}.$$
The Coding Scheme

- Rate control is achieved via distributed algorithms.
- Coding scheme? Modified random linear coding.

**Theorem 4** With modified random linear coding over $\text{GF}(q)$, the success probability is

$$\text{Prob}(\text{success}) \geq \left(1 - \frac{4}{q}\right)^{6|E|}.$$ 

Coding is independent of Queuing & rate allocation!
The Implementation Issues

The control messages to collect the info. nec. for maximizing the Lagrangian.

\[
\begin{align*}
\sum_{i=1}^{I} U_i & \left( \sum_{k=1}^{\|P(i)\|} x_{i}^{k} + \sum_{j \neq i} \sum_{l=1}^{\|P(i,j)\|} \sum_{m=1}^{\|P(j,i)\|} g_{ij}^{lm} \right) \\
- \sum_{i=1}^{I} \frac{c_i}{2} \left( x_{i}^{k} - y_{i}^{k} \right)^2 & - \sum_{i=1}^{I} \sum_{j \neq i} \sum_{l=1}^{\|P(i,j)\|} \sum_{m=1}^{\|P(j,i)\|} d_{i}^{lm} \left( g_{ij}^{lm} - h_{ij}^{lm} \right)^2 \\
- \sum_{e} \lambda_{e} & \left( \sum_{i=1}^{I} \sum_{k=1}^{\|P(i)\|} E_{e,i}^{k} x_{i}^{k} + \sum_{i=1}^{I} \sum_{i<j} \sum_{l=1}^{\|P(i,j)\|} \sum_{m=1}^{\|P(j,i)\|} \max(H_{e,ij}^{l}, H_{e,ji}^{m}) g_{ij}^{lm} - C_{e} \right) \\
- \sum_{i=1}^{I} \sum_{i<j} \sum_{l} \sum_{m} \mu_{ij}^{lm} \left( g_{ij}^{lm} - g_{ji}^{ml} \right)
\end{align*}
\]
The Implementation Issues

- The control messages to collect the info. nec. for maximizing the Lagrangian.

\[ \sum_{i=1}^{I} U_i \left( \sum_{k=1}^{\text{|P}(i)|} x_i^k + \sum_{j \neq i} \sum_{l=1}^{\text{|P}(i,j)|} \sum_{m=1}^{\text{|P}(j,i)|} g_{ij}^{lm} \right) \]

\[- \sum_{i=1}^{I} \sum_{k} \frac{c_i}{2} (x_i^k - y_i^k)^2 - \sum_{i=1}^{I} \sum_{j \neq i} \sum_{l=1}^{\text{|P}(i,j)|} \sum_{m=1}^{\text{|P}(j,i)|} \frac{d_i}{2} (g_{ij}^{lm} - h_{ij}^{lm})^2 \]

\[- \sum_{e} \lambda_e \left( \sum_{i=1}^{I} \sum_{k=1}^{\text{|P}(i)|} E_{e,i}^k x_i^k + \sum_{i=1}^{I} \sum_{i<j} \sum_{l=1}^{\text{|P}(i,j)|} \sum_{m=1}^{\text{|P}(j,i)|} \max(H_{e,ij}^l, H_{e,ji}^m) g_{ij}^{lm} - C_e \right) \]

\[- \sum_{i=1}^{I} \sum_{i<j} \sum_{l} \sum_{m} \mu_{ij}^{lm} (g_{ij}^{lm} - g_{ji}^{ml}) \]

- Adaptively select \( P(i) \) and \( P(i,j) \).
Numerical Experiments

\[ X_1 + X_2, \quad \lambda, \mu = 0.0005 \]

\[ \lambda, \mu = 0.0005 \]

- \( K = 5 \)
- \( K = 10 \)
- \( K = 15 \)
- \( K = 20 \)
Capacity

**Theorem 5**  *For any network with $I$ unicast sessions, any rate vector $(R_1, \ldots, R_I)$ that is achievable with the $\mathcal{TRLKM}$ or the $\mathcal{WS}$ region is also achievable with the $\mathcal{I} - \mathcal{TRLKM}$ region.*
Capacity & Fairness

- **Utility gain** $\mathcal{U}_i$: \[ \frac{\sum U_i(\text{intersession NC.}) - \sum U_i(\text{routing})}{\sum U_i(\text{routing})} \]

- **Throughput gain** $\mathcal{T}_i$: \[ \frac{\sum r_i(\text{intersession NC.}) - \sum r_i(\text{routing})}{\sum r_i(\text{routing})} \]
Capacity & Fairness

- Utility gain $\mathcal{U}G$: 
  \[
  \frac{\sum U_i(\text{intersession NC.}) - \sum U_i(\text{routing})}{\sum U_i(\text{routing})}.
  \]

- Throughput gain $\mathcal{T}G$: 
  \[
  \frac{\sum r_i(\text{intersession NC.}) - \sum r_i(\text{routing})}{\sum r_i(\text{routing})}.
  \]
Capacity & Fairness

- **Utility gain \( \mathcal{UG} \):**
  \[
  \frac{\sum U_i(\text{intersession NC.}) - \sum U_i(\text{routing})}{\sum U_i(\text{routing})}
  \]

- **Throughput gain \( \mathcal{TG} \):**
  \[
  \frac{\sum r_i(\text{intersession NC.}) - \sum r_i(\text{routing})}{\sum r_i(\text{routing})}
  \]

{\{I - TRKM, WS\} > \{TRLKM\}}
Sim. Results

\[ \log(\gamma + r_i) \]
Sim. Results

\[ \log(\gamma + r_i) \]

Percentage Gain

Utility gain

Throughput gain

30+%
Sim. Results

\[ r_i^{1-\alpha} \]
\[ \frac{1}{1-\alpha} \]

![Graph showing percentage gain with utility gain and throughput gain curves.](image-url)
Sim. Results

30+%
Capacity

\[ \{I - TRLKM, TRLKM\} > \{WS\} \]
Capacity

- $\{I - TRKM, TRLKM\} > \{WS\}$
Capacity

\[ \{I - TRKM\} > \{WS, TRLKM\} \]
Capacity

$\{I - TRKM\} > \{WS, TRLKM\}$
## Complexity & Dist. Implementation

<table>
<thead>
<tr>
<th></th>
<th>Constraints</th>
<th># Variables</th>
<th># Constraints</th>
<th>Coding scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRLKM</td>
<td>Linear</td>
<td>$&gt; 3</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>WS</td>
<td>Linear</td>
<td>$\sum_{i=1}^{I} \sum_{i&lt;j} J(i,j)J(j,i) + \sum_{i} J(i)$</td>
<td>$</td>
<td>E</td>
</tr>
<tr>
<td>$I - TRLKM$</td>
<td>Non Linear</td>
<td>$&gt; 3</td>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>
## Complexity & Dist. Implementation

<table>
<thead>
<tr>
<th>Constraints</th>
<th># Variables</th>
<th># Constraints</th>
<th>Coding scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRLKM</td>
<td>Linear</td>
<td>$&gt; 3</td>
<td>E</td>
</tr>
<tr>
<td>WS</td>
<td>Linear</td>
<td>$\sum_{i=1}^{I} \sum_{i&lt;j} J(i,j)J(j,i) + \sum_{i} J(i)$</td>
<td>$</td>
</tr>
<tr>
<td>$I - TRLKM$</td>
<td>Non Linear</td>
<td>$&gt; 3</td>
<td>E</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Approach</th>
<th>Rate control</th>
<th>Rate alloc and coding</th>
<th>Queues exchange</th>
<th>Adaptive complexity reduction</th>
<th>Coding scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSC</td>
<td>Structure based</td>
<td>NO</td>
<td>Dependent</td>
<td>YES</td>
<td>hard</td>
</tr>
<tr>
<td>AlgorithmA</td>
<td>Path based</td>
<td>YES</td>
<td>Independent</td>
<td>NO</td>
<td>easy</td>
</tr>
<tr>
<td>$I - RSC$</td>
<td>Structure based</td>
<td>NO</td>
<td>Dependent</td>
<td>YES</td>
<td>hard</td>
</tr>
</tbody>
</table>
Conclusions

- Introduced the $\mathcal{WS}$ and $\mathcal{I} - TRLKM$ capacity regions and compared them with the $TRLKM$ capacity region.
Conclusions

- Introduced the $\mathcal{WS}$ and $\mathcal{I} - \mathcal{TRLKM}$ capacity regions and compared them with the $\mathcal{TRLKM}$ capacity region.

- The distributed algorithm can be extended to include the wireless case.
Conclusions

- Introduced the \( WS \) and \( \mathcal{I} - \mathcal{TRLKM} \) capacity regions and compared them with the \( \mathcal{TRLKM} \) capacity region.

- The distributed algorithm can be extended to include the wireless case.

- The path-based construction admits new distributed rate control algorithms with lower complexity and distributed coding scheme.
Conclusions

- Introduced the $\mathcal{WS}$ and $\mathcal{I} - \mathcal{TRLKM}$ capacity regions and compared them with the $\mathcal{TRLKM}$ capacity region.
- The distributed algorithm can be extended to include the wireless case.
- The path-based construction admits new distributed rate control algorithms with lower complexity and distributed coding scheme.
- Intersession network coding promotes further fairness.
Conclusions

- Integration of the adaptive version of Algorithm A with the real networks as the internet.
Conclusions

- Integration of the adaptive version of Algorithm $A$ with the real networks as the internet.
- Need to consider coding between more than two sessions.
Conclusions

- Integration of the adaptive version of Algorithm A with the real networks as the internet.
- Need to consider coding between more than two sessions.
- Similar capacity regions can be used for multicast.