Delay-Constrained Unicast: Improved upper bounds

Sudeep Kamath

PRINCETON UNIVERSITY

Joint work with

Chandra Chekuri

Sreeram Kannan

Pramod Viswanath

DIMACS workshop on Network Coding, 17 December 2015

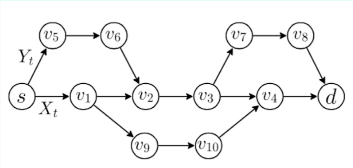
Delay-constrained unicast [Wang-Chen '14]

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D

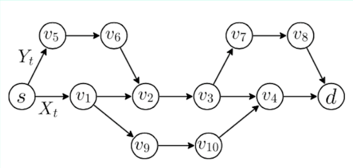
Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D

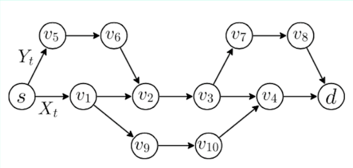


For this network with D=6,

$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



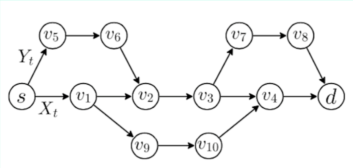
For this network with D=6,

$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

- Practical constraint - eg. video streaming, financial data

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



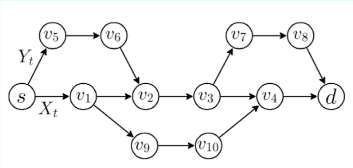
For this network with D=6,

$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

- Practical constraint - eg. video streaming, financial data
- Intra-flow coding has fewer security and privacy concerns

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



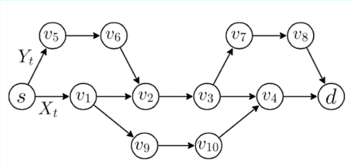
For this network with D=6,

$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

- Practical constraint - eg. video streaming, financial data
- Intra-flow coding has fewer security and privacy concerns
- Implementation aligned with self-interest

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D

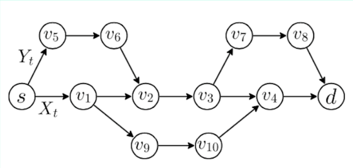


For this network with D=6,

$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



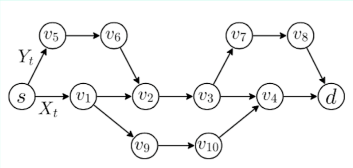
For this network with D=6,

$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

How large can this ratio be?

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



For this network with D=6,

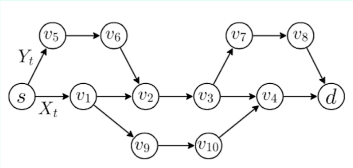
$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

How large can this ratio be?

$$2 \leq \sup_{\text{graphs}} \frac{\text{Capacity}}{\text{Flow}} \leq D+1$$

Delay-constrained unicast [Wang-Chen '14]

Single flow with delay constraint D



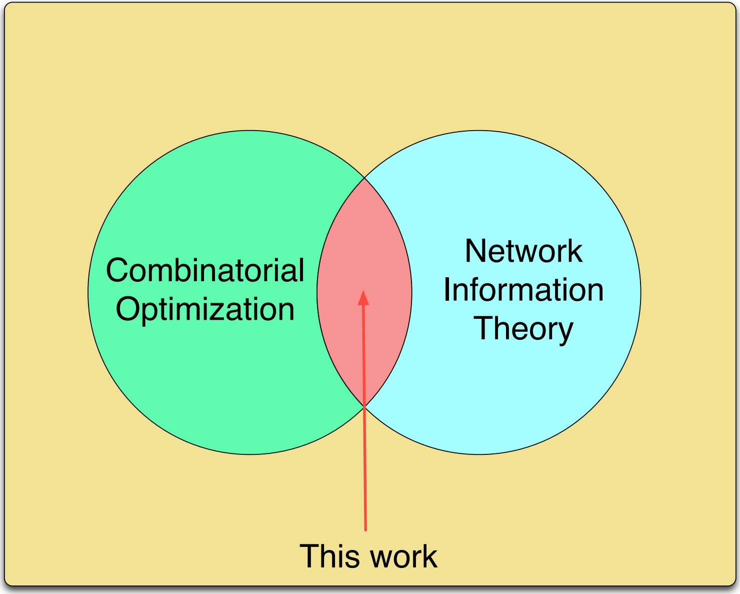
For this network with D=6,

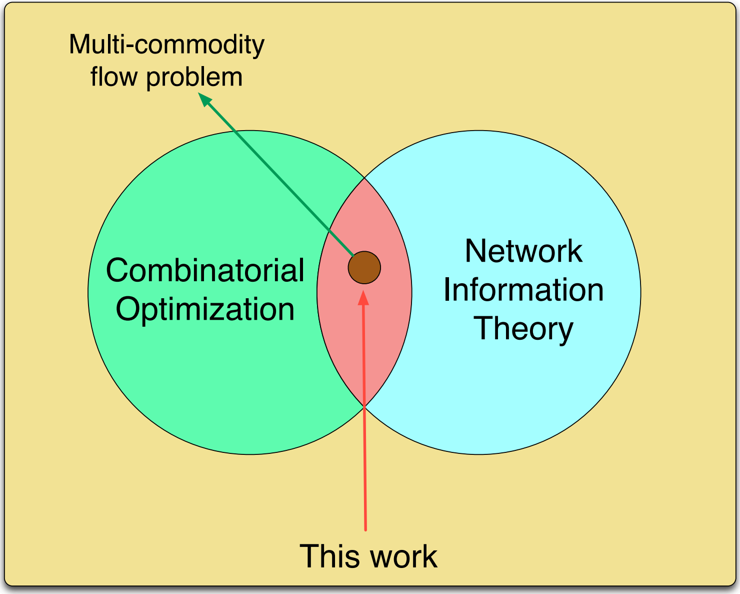
$$\frac{\text{Capacity}}{\text{Flow}} = \frac{4}{3}$$

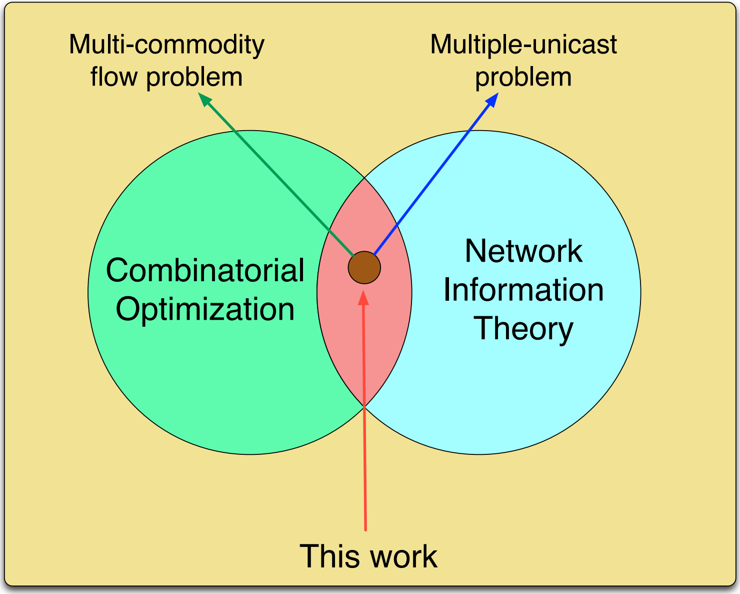
How large can this ratio be?

$$2 \leq \sup_{\text{graphs}} \frac{\text{Capacity}}{\text{Flow}} \leq D+1$$

We improve over this


Combinatorial Optimization
Network Information Theory
This work


Multi-commodity
flow problem
Combinatorial
Optimization
Network
Information
Theory
This work

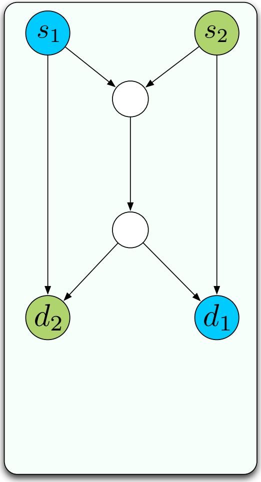

Multi-commodity
flow problem
Multiple-unicast
problem
Combinatorial
Optimization
Network
Information
Theory
This work

Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Multi-commodity flow / Multiple-unicast

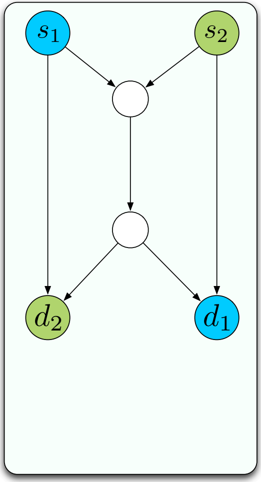
Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$



Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

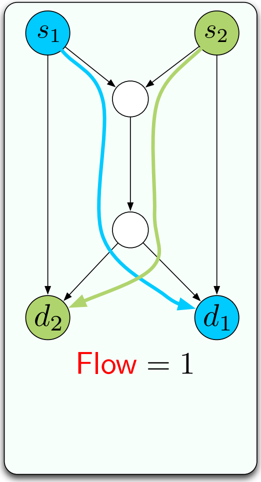
Flow: Maximum total commodity flow



Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

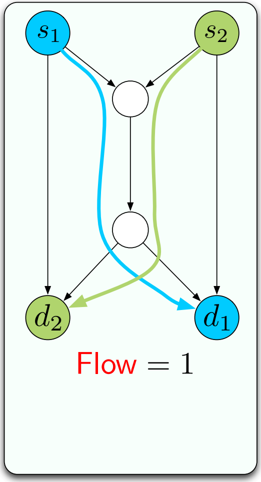


Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

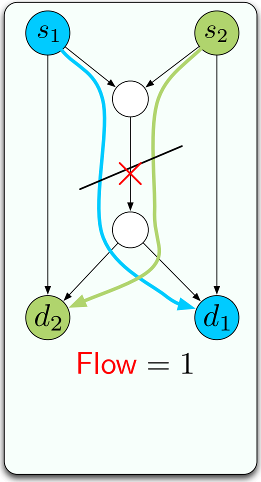


Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

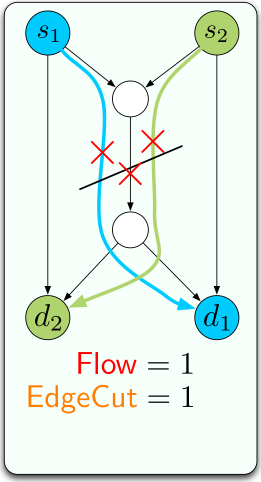


Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$



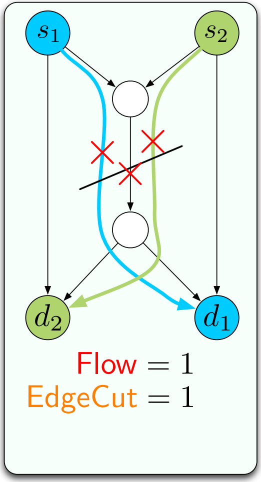
Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

Capacity: Maximum information flow



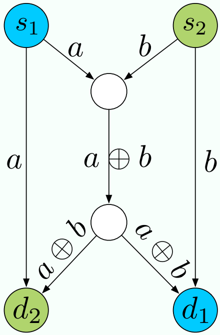
Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

Capacity: Maximum information flow



Flow $= 1$
EdgeCut $= 1$

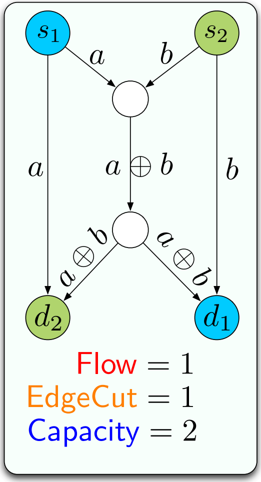
Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

Capacity: Maximum information flow



Multi-commodity flow / Multiple-unicast

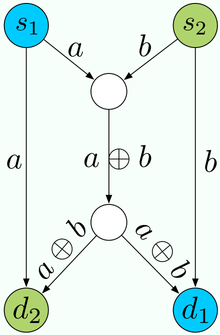
Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

Capacity: Maximum information flow

EdgeCut $\neq$ Cutset bound



Flow $= 1$
EdgeCut $= 1$
Capacity $= 2$

Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

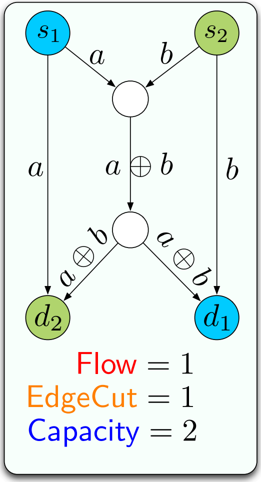
Flow: Maximum total commodity flow

EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

Capacity: Maximum information flow

EdgeCut $\neq$ Cutset bound

Capacity $\leq$ Cutset bound



Multi-commodity flow / Multiple-unicast

Given a directed graph and k source-destination pairs $\{(s_i, d_i)\}$

Flow: Maximum total <u>commodity</u> flow

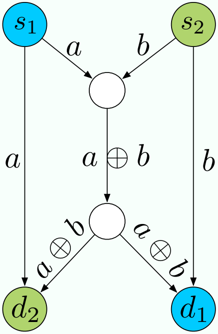
EdgeCut: Fewest edges whose removal disconnects all paths from s_i to d_i $\forall i$

Capacity: Maximum <u>information</u> flow

EdgeCut $\neq$ Cutset bound

Capacity $\leq$ Cutset bound

However, we may have EdgeCut $<$ Capacity



Flow $= 1$
EdgeCut $= 1$
Capacity $= 2$

For k = 1: $\text{Flow} = \text{EdgeCut} = \text{Capacity}$

(Max-Flow Min-Cut Theorem)

For k = 1: Flow $=$ EdgeCut $=$ Capacity

(Max-Flow Min-Cut Theorem)

For k ≥ 2:

For k = 1: $\text{Flow} = \text{EdgeCut} = \text{Capacity}$

(Max-Flow Min-Cut Theorem)

For k ≥ 2:

$\text{Flow} \leq \text{EdgeCut}$

$\text{Flow} \leq \text{Capacity}$

$\text{EdgeCut} \lesseqgtr \text{Capacity}$

For k = 1: Flow $=$ EdgeCut $=$ Capacity

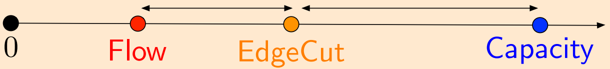
(Max-Flow Min-Cut Theorem)

For k ≥ 2:

Flow $\leq$ EdgeCut

Flow $\leq$ Capacity

EdgeCut $\lessgtr$ Capacity



For k = 1: Flow $=$ EdgeCut $=$ Capacity

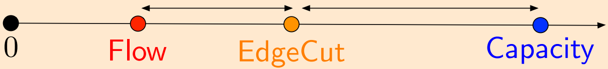
(Max-Flow Min-Cut Theorem)

For k ≥ 2:

Flow $\leq$ EdgeCut

Flow $\leq$ Capacity

EdgeCut $\lessgtr$ Capacity



[Saks-Samorodnitsky-Zosin '04], [Harvey-Kleinberg-Lehman '06], [Ambühl-Mastrolilli-Svensson '07], [Chuzhoy-Khanna '07], [Dougherty-Freiling-Zeger '05], [Chan-Grant '08]

For k = 1: Flow $=$ EdgeCut $=$ Capacity

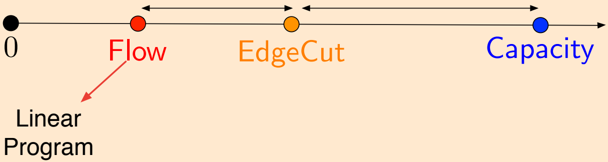
(Max-Flow Min-Cut Theorem)

For k ≥ 2:

Flow $\leq$ EdgeCut

Flow $\leq$ Capacity

EdgeCut $\lesssim$ Capacity



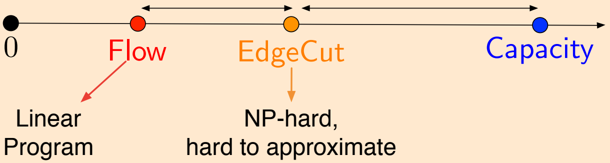
[Saks-Samorodnitsky-Zosin '04], [Harvey-Kleinberg-Lehman '06], [Ambühl-Mastrolilli-Svensson '07], [Chuzhoy-Khanna '07], [Dougherty-Freiling-Zeger '05], [Chan-Grant '08]

For k = 1: Flow $=$ EdgeCut $=$ Capacity

(Max-Flow Min-Cut Theorem)

For k ≥ 2:

Flow $\leq$ EdgeCut | Flow $\leq$ Capacity | EdgeCut $\lessgtr$ Capacity



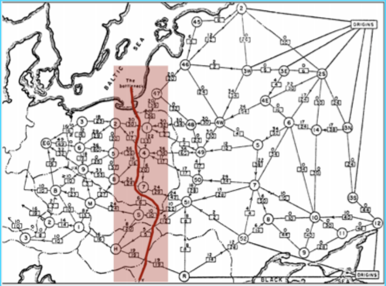
[Saks-Samorodnitsky-Zosin '04], [Harvey-Kleinberg-Lehman '06],
[Ambühl-Mastrolilli-Svensson '07], [Chuzhoy-Khanna '07],
[Dougherty-Freiling-Zeger '05], [Chan-Grant '08]

For k = 1: Flow = EdgeCut = Capacity

Why bother with EdgeCut if Flow is a linear program?

For k = 1: Flow = EdgeCut = Capacity

Why bother with EdgeCut if Flow is a linear program?



For k = 1: Flow = EdgeCut = Capacity

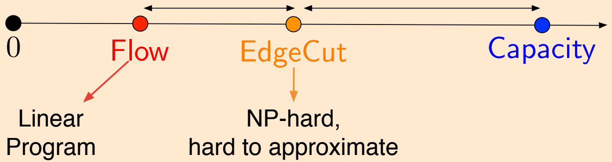
(Max-Flow Min-Cut Theorem)

For k ≥ 2:

Flow $\leq$ EdgeCut

Flow $\leq$ Capacity

EdgeCut $\lessgtr$ Capacity



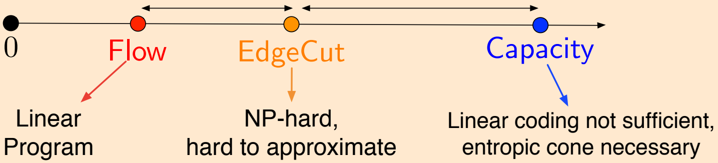
[Saks-Samorodnitsky-Zosin '04], [Harvey-Kleinberg-Lehman '06], [Ambühl-Mastrolilli-Svensson '07], [Chuzhoy-Khanna '07], [Dougherty-Freiling-Zeger '05], [Chan-Grant '08]

For k = 1: Flow $=$ EdgeCut $=$ Capacity

(Max-Flow Min-Cut Theorem)

For k ≥ 2:

Flow $\leq$ EdgeCut | Flow $\leq$ Capacity | EdgeCut $\lesssim$ Capacity



[Saks-Samorodnitsky-Zosin '04], [Harvey-Kleinberg-Lehman '06],
[Ambühl-Mastrolilli-Svensson '07], [Chuzhoy-Khanna '07],
[Dougherty-Freiling-Zeger '05], [Chan-Grant '08]

For k = 1: Flow $=$ EdgeCut $=$ Capacity

(Max-Flow Min-Cut Theorem)

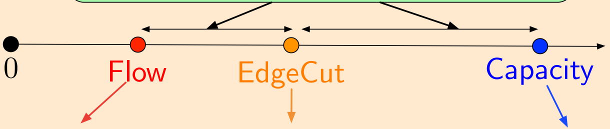
For k ≥ 2:

Flow $\leq$ EdgeCut

Flow $\leq$ Capacity

EdgeCut $\lessgtr$ Capacity

Can be multiplicative factor k apart



Linear Program

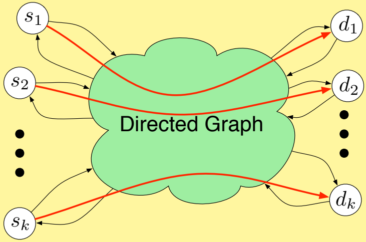
NP-hard, hard to approximate

Linear coding not sufficient, entropic cone necessary

[Saks-Samorodnitsky-Zosin '04], [Harvey-Kleinberg-Lehman '06], [Ambühl-Mastrolilli-Svensson '07], [Chuzhoy-Khanna '07], [Dougherty-Freiling-Zeger '05], [Chan-Grant '08]

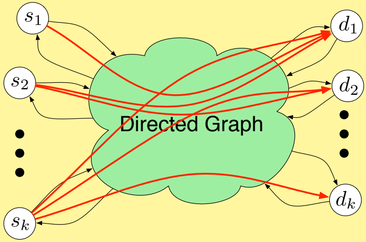
Triangle-cast

Triangle-cast



Multiple-unicast: flow from s_i to d_i for all i

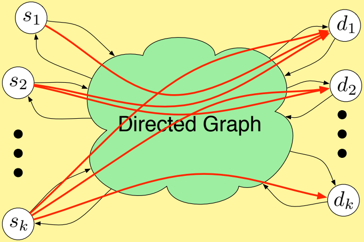
Triangle-cast



Multiple-unicast: flow from s_i to d_i for all i

Triangle-cast: flow from s_i to d_j for all $i \geq j$

Triangle-cast

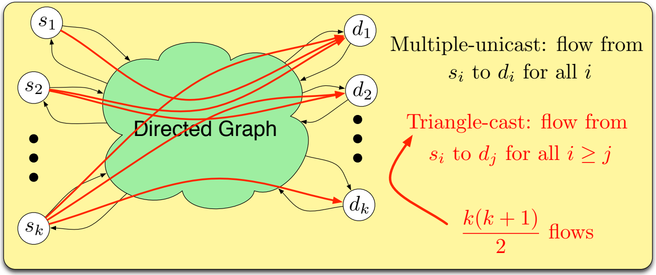


Multiple-unicast: flow from s_i to d_i for all i

Triangle-cast: flow from s_i to d_j for all $i \geq j$

$\frac{k(k+1)}{2}$ flows

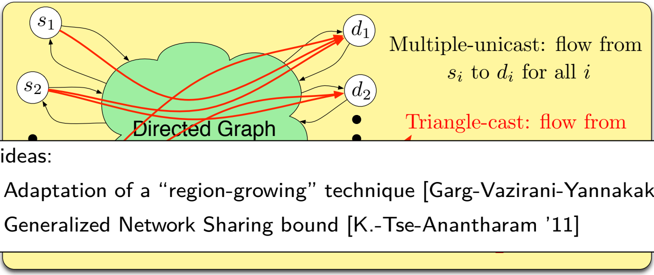
Triangle-cast



Main Result 1: For triangle-cast as above,

$$\frac{\text{EdgeCut}}{4\log_e(k+1)} \leq \text{Flow} \leq \text{Capacity} \leq \text{EdgeCut}$$

Triangle-cast



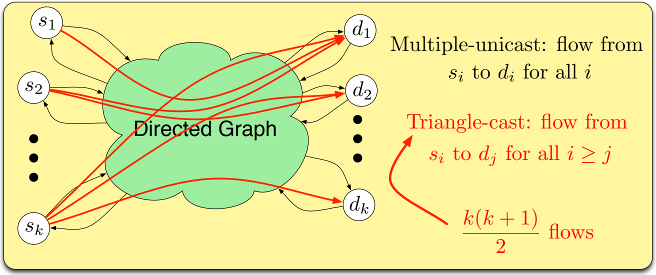
Main ideas:

- Adaptation of a "region-growing" technique [Garg-Vazirani-Yannakakis '96]
- Generalized Network Sharing bound [K.-Tse-Anantharam '11]

Main Result 1: For triangle-cast as above,

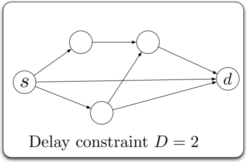
$$\frac{\text{EdgeCut}}{4\log_e(k+1)} \leq \text{Flow} \leq \text{Capacity} \leq \text{EdgeCut}$$

Triangle-cast

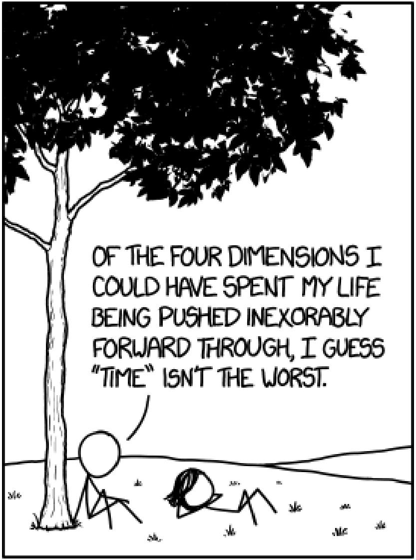


Main Result 1: For triangle-cast as above,

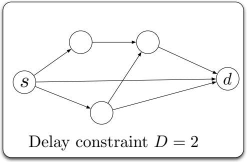
$$\frac{\text{EdgeCut}}{4\log_e(k+1)} \leq \text{Flow} \leq \text{Capacity} \leq \text{EdgeCut}$$



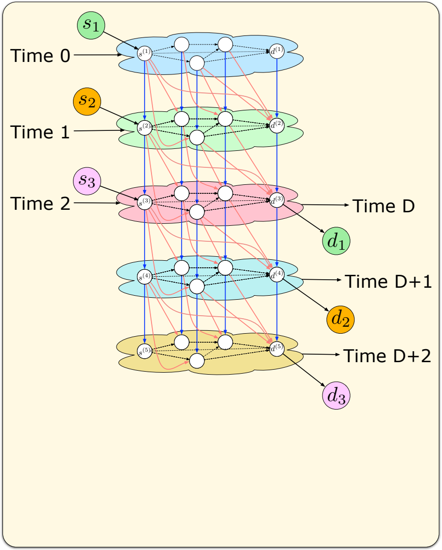
Delay constrained network

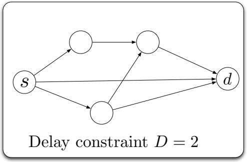


Reproduced from xkcd.com

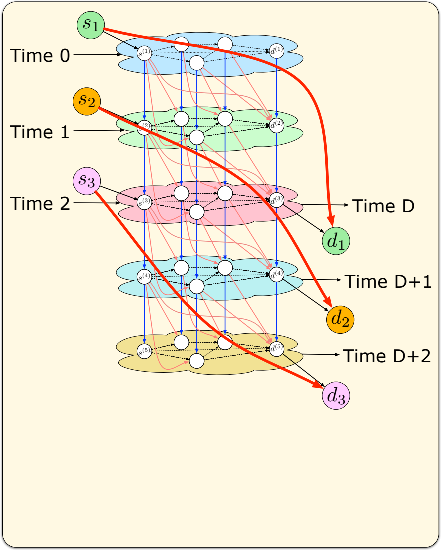


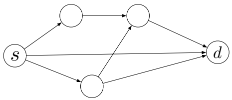
Delay constrained network





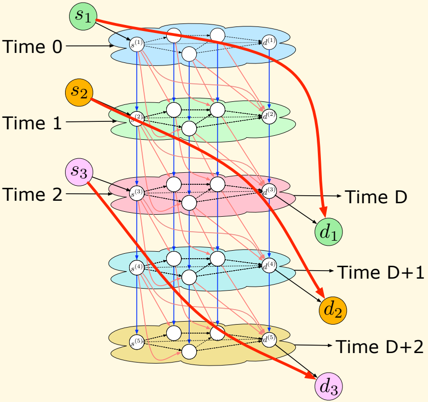
Delay constrained network



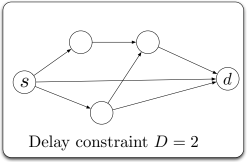


Delay constraint $D = 2$

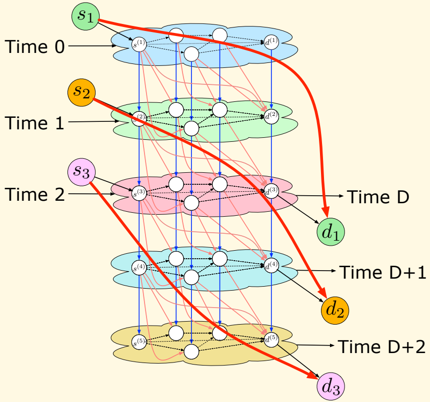
Delay constrained network



Multiple-unicast

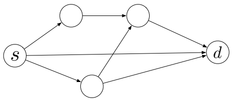


Delay constrained network



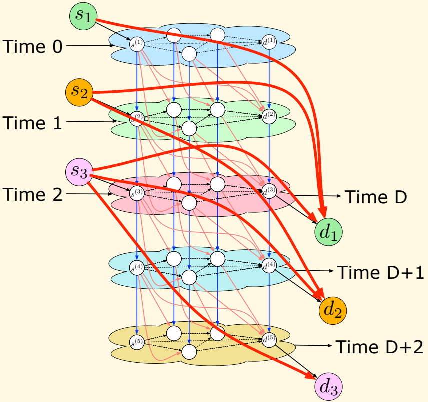
Multiple-unicast

Information delivered earlier is ok!



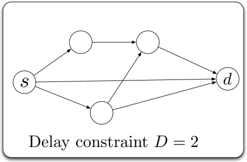
Delay constraint $D = 2$

Delay constrained network



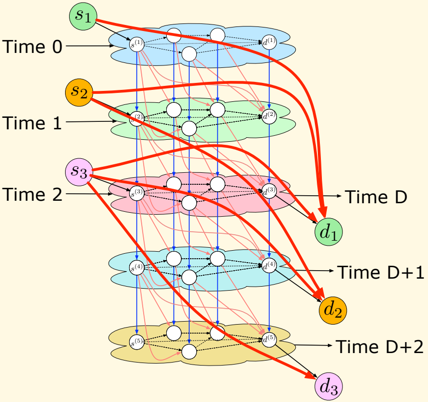
Multiple-unicast

Information delivered earlier is ok!



Delay constrained network

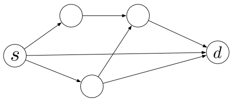
Delay constrained network



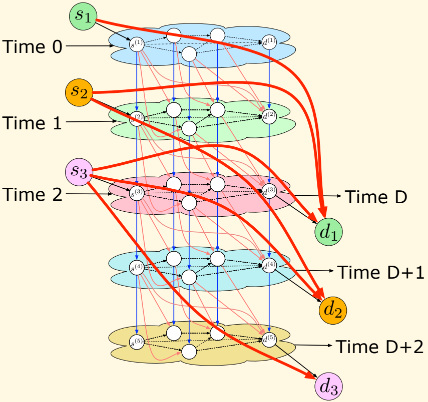
Multiple-unicast

Information delivered earlier is ok!

Hence, triangle-cast!



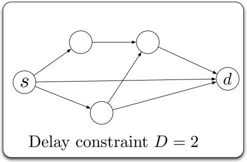
Delay constraint $D = 2$



Multiple-unicast

Information delivered earlier is ok!

Hence, triangle-cast!



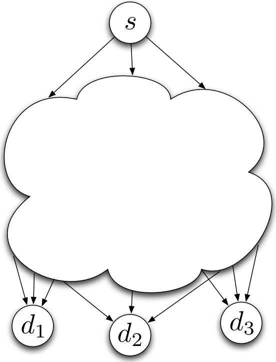
Delay constrained network

Main Result 2

$$\frac{\text{Capacity}}{\text{Flow}} \leq 8 \log_e(D+1)$$

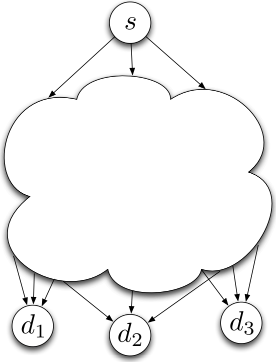
$D =$ Delay

Open 1: "Multicast"



Multicast: same information to all destinations

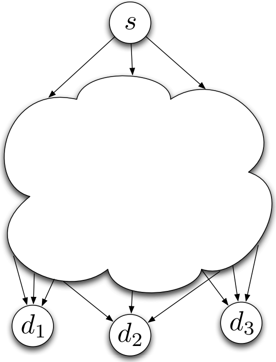
Open 1: "Multicast"



Multicast: same information to all destinations

What happens with delay constraint?

Open 1: "Multicast"



Multicast: same information to all destinations

What happens with delay constraint?

- Practical constraint
- Intra-flow coding
- Coding strategies? - Random coding does not work

Open 2: "Triangle-cast gap"

Theorem (this work)

For k-triangle-cast,

$$\frac{\text{EdgeCut}}{4\log_e(k+1)} \leq \text{Flow} \leq \text{Capacity} \leq \text{EdgeCut}$$

Open 2: "Triangle-cast gap"

Theorem (this work)

For k-triangle-cast,

$$\frac{\text{EdgeCut}}{4\log_e(k+1)} \leq \text{Flow} \leq \text{Capacity} \leq \text{EdgeCut}$$

Conjecture

For k-triangle-cast,

$$\frac{\text{EdgeCut}}{2} \leq \text{Flow} \leq \text{Capacity} \leq \text{EdgeCut}$$

Hence, for delay-constrained unicast,

$$\frac{\text{Capacity}}{\text{Flow}} \leq 2$$

Open 3: "Symmetry Principle"

Principle

Under suitable symmetry in traffic pattern, Flow, EdgeCut, Capacity are all not "too far" apart.

Open 3: "Symmetry Principle"

F : Flow
EC : EdgeCut
C : Capacity

Open 3: "Symmetry Principle"

F : Flow
EC : EdgeCut
C : Capacity

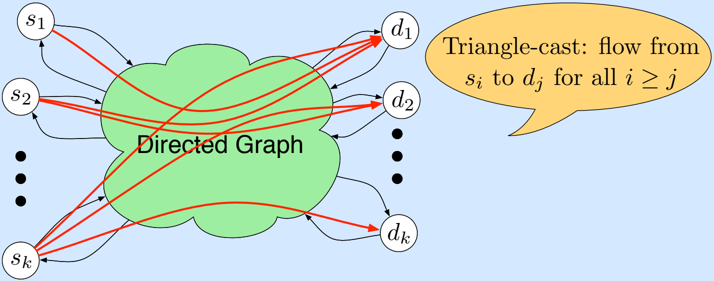
Bidirected Networks	$\frac{\text{EC}}{\Theta(\log k)} \leq \text{F} \leq \text{EC}$ [Leighton-Rao '88] [Linial-London-Rabinovich '94]	$\text{F} \leq \text{C} \leq \text{EC}$ [K.-Viswanath '12]
Symmetric Demands	$\frac{\text{EC}}{\Theta(\log^3 k)} \leq \text{F} \leq \text{EC}$ [Klein-Plotkin-Rao-Tardos '93]	$\text{F} \leq \text{C} \leq \text{EC}$ [K.-Viswanath '12]
Group-cast	$\frac{\text{EC}}{2} \leq \text{F} \leq \text{EC}$ [Naor-Zosin '01]	$\text{F} \leq \text{C} \leq 2 \times \text{EC}$ [K.-Viswanath '12]
Triangle-cast	$\frac{\text{EC}}{\Theta(\log k)} \leq \text{F} \leq \text{EC}$ [This work]	$\text{F} \leq \text{C} \leq \text{EC}$ [This work]

Open 3: "Symmetry Principle"

F : Flow
EC : EdgeCut
C : Capacity

Bidirected Networks	$\frac{EC}{\Theta(\log k)} \leq F \leq EC$ [Leighton-Rao '88] [Linial-London-Rabinovich '94]	$F \leq C \leq EC$ [K.-Viswanath '12]
Symmetric Demands	$\frac{EC}{\Theta(\log^3 k)} \leq F \leq EC$ [Klein-Plotkin-Rao-Tardos '93]	$F \leq C \leq EC$ [K.-Viswanath '12]
Group-cast	$\frac{EC}{2} \leq F \leq EC$ [Naor-Zosin '01]	$F \leq C \leq 2 \times EC$ [K.-Viswanath '12]
Triangle-cast	$\frac{EC}{\Theta(\log k)} \leq F \leq EC$ [This work]	$F \leq C \leq EC$ [This work]
???		

Conclusion



Showed a $4 \log_e(k+1)$ Flow – EdgeCut – Capacity approximation guarantee

$$\Downarrow$$

Delay-constrained unicast (delay D) has $\frac{\text{Capacity}}{\text{Flow}} \leq 8 \log_e(D+1)$