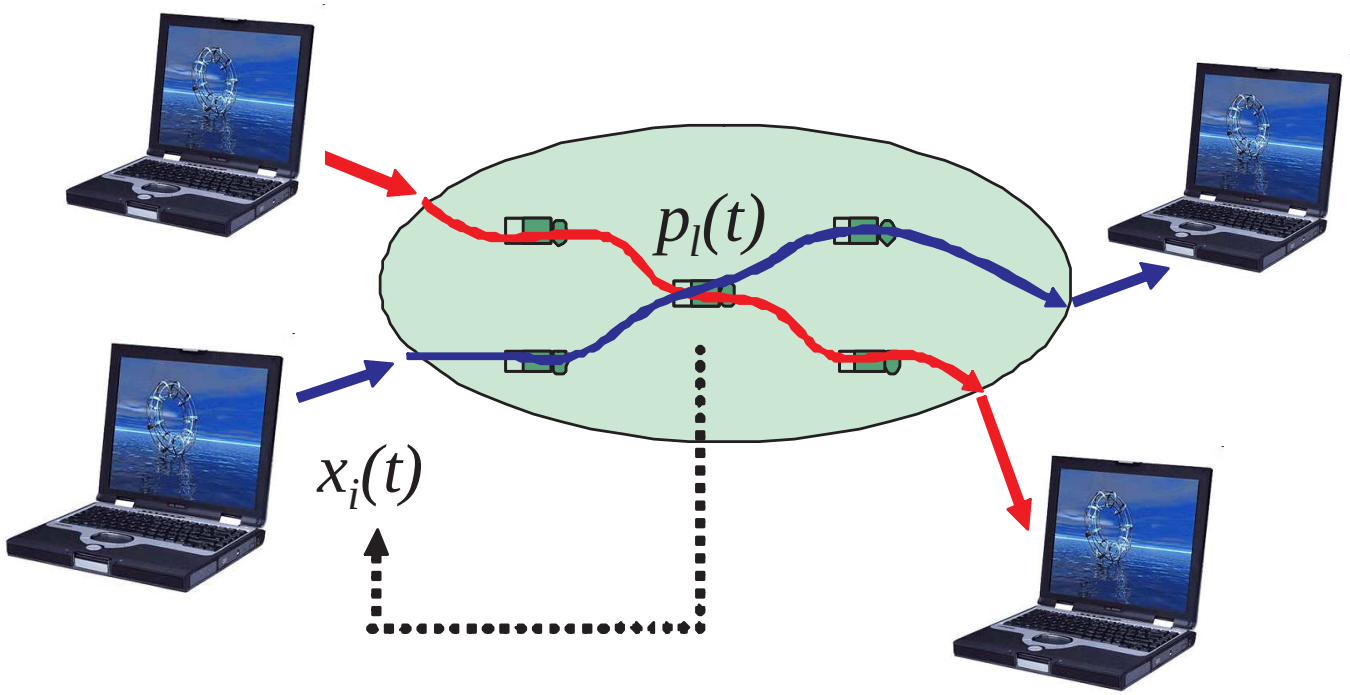
Internet Flow Control: From Optimization to Game Theory

A. Kevin Tang

School of ECE, Cornell University
Joint work with Lachlan Andrew at Caltech

CISS, Princeton, March 19 2008

Introduction: Congestion Control



- Sources: control data rates according to congestion signals
- Links: adapt congestion signals based on bandwidth utilization
- A purely distributed system!

A Global View of Congestion Control

A distributed feedback control system

- Resource Allocation: Equilibrium (Optimization Theory)
- Interconnected Dynamical System: Dynamics (Control Theory)

A Global View of Congestion Control

A distributed feedback control system

- Resource Allocation: Equilibrium (Optimization Theory)
- Interconnected Dynamical System: Dynamics (Control Theory)

Help answer questions like

- How to understand the whole system for arbitrary (possibly very complex) topology?
- Can all local behaviors of sources and links achieve a globally coordinated goal?
- Are current congestion control schemes scalable? Better ones?

[Kelly-Maulloo-Tan 98], [Low-Lapsley 99], [Mo-Walrand 00]...

Model and Notations

- L links, indexed by $l = 1, \ldots, L$. Link l has a finite capacity c_l and a congestion signal $p_l(t)$.
- N flows, indexed by $i = 1, \ldots, N$. Flow i maintains a rate $x_i(t)$.
- f_i abstracts the TCP algorithm of flow i.
- g_l describes the AQM (Active Queue Management) algorithm at link l.

$$\begin{aligned} \dot{x}_i &= f_i\left(x_i(t), \sum_{l \in L(i)} p_l(t)\right) \\ \dot{p}_l &= g_l\left(\sum_{i:l \in L(i)} x_i(t), p_l(t)\right) \end{aligned}$$

An Example of TCP: TCP Reno

TCP Reno Updating rule (Additive Increase Multiplicative Decrease):

$$\dot{x}_i = f_i\left(x_i(t), \sum_{l\in L(i)} p_l(t)\right) = \frac{1-q_i(t)}{\tau_i(t)^2} - \frac{1}{2}q_i(t)x_i^2(t)$$

$q_i(t) = \sum_{l\in L(i)} p_l(t)$: packet loss rate;
$\tau_i(t)$: round trip time (RTT).

An Example of TCP: TCP Reno

TCP Reno Updating rule (Additive Increase Multiplicative Decrease):

$$\dot{x}_i = f_i\left(x_i(t), \sum_{l\in L(i)} p_l(t)\right) = \frac{1-q_i(t)}{\tau_i(t)^2} - \frac{1}{2}q_i(t)x_i^2(t)$$

$q_i(t) = \sum_{l\in L(i)} p_l(t)$: packet loss rate;
$\tau_i(t)$: round trip time (RTT).
At equilibrium:

$$q_i = \frac{2}{\tau_i^2 x_i^2}$$

An Example of TCP: TCP Reno

TCP Reno Updating rule (Additive Increase Multiplicative Decrease):

$$\dot{x}_i = f_i\left(x_i(t), \sum_{l\in L(i)} p_l(t)\right) = \frac{1-q_i(t)}{\tau_i(t)^2} - \frac{1}{2}q_i(t)x_i^2(t)$$

$q_i(t) = \sum_{l\in L(i)} p_l(t)$: packet loss rate;
$\tau_i(t)$: round trip time (RTT).
At equilibrium:

$$q_i = \frac{2}{\tau_i^2 x_i^2}$$

Utility function:

$$\max_{x_i\geq 0} \quad U_i(x_i) - x_i q_i$$

$$U_i(x_i) = -\frac{2}{x_i\tau_i^2}$$

Current Theory and Its Impact

Main Theorem: The equilibrium rate vector solves

$$\max_{x \geq 0} \sum_i U_i(x_i) \quad \text{subject to} \quad \sum_{i:l \in L(i)} x_i \leq c_l$$

The source and link protocols serve as a primal-dual algorithm.

Current Theory and Its Impact

Main Theorem: The equilibrium rate vector solves

$$\max_{x \geq 0} \sum_i U_i(x_i) \quad \text{subject to} \quad \sum_{i:l \in L(i)} x_i \leq c_l$$

The source and link protocols serve as a primal-dual algorithm.

Proof: the physical congestion signal $\Leftrightarrow$ the mathematical lagrange multipliers.

Current Theory and Its Impact

Main Theorem: The equilibrium rate vector solves

$$\max_{x \geq 0} \sum_i U_i(x_i) \quad \text{subject to} \quad \sum_{i:l \in L(i)} x_i \leq c_l$$

The source and link protocols serve as a primal-dual algorithm.

Proof: the physical congestion signal $\Leftrightarrow$ the mathematical lagrange multipliers.

Implication, Application and Impact:

Basic properties: existence, uniqueness, optimality

Engineering practice: new protocol design

Wireless networks

Other directions: random arrivals and departures of flows ...

Outline

- Introduction to congestion control and its current theory
- Equilibrium of heterogeneous congestion control

  Heterogeneity of congestion signals

  The same congestion signal but different algorithms: Homogeneous
- A Game theoretical framework
- Conclusion

Heterogeneous Congestion Control

- TCP Reno does not work well for networks with large bandwidth-delay products.

  Requires extremely small equilibrium loss rate.

  Becomes harder to maintain flow level stability.

Heterogeneous Congestion Control

- TCP Reno does not work well for networks with large bandwidth-delay products.

  Requires extremely small equilibrium loss rate.

  Becomes harder to maintain flow level stability.

- One solution: use other congestion signals

- Heterogeneous Protocols: Protocols that use different congestion signals

Why Study the Heterogeneous Case?

- Various proposals that use different congestion signals

  queueing delay (CARD, DUAL, Vegas, FAST)

  packet loss (Reno and its variants)

  both loss and delay (Westwood, Compound TCP)

  one bit ECN (IETF RFC 2481)

Why Study the Heterogeneous Case?

- Various proposals that use different congestion signals

  queueing delay (CARD, DUAL, Vegas, FAST)

  packet loss (Reno and its variants)

  both loss and delay (Westwood, Compound TCP)

  one bit ECN (IETF RFC 2481)

- The Linux operating system already allows users to choose from a variety of congestion control algorithms since kernel version 2.6.13

Why Study the Heterogeneous Case?

- Various proposals that use different congestion signals

  queueing delay (CARD, DUAL, Vegas, FAST)

  packet loss (Reno and its variants)

  both loss and delay (Westwood, Compound TCP)

  one bit ECN (IETF RFC 2481)

- The Linux operating system already allows users to choose from a variety of congestion control algorithms since kernel version 2.6.13

- Compound TCP which uses multiple congestion signals is part of MS Windows Vista and Windows 2008 TCP stack.

Why Study the Heterogeneous Case?

- Various proposals that use different congestion signals

  queueing delay (CARD, DUAL, Vegas, FAST)

  packet loss (Reno and its variants)

  both loss and delay (Westwood, Compound TCP)

  one bit ECN (IETF RFC 2481)

- The Linux operating system already allows users to choose from a variety of congestion control algorithms since kernel version 2.6.13

- Compound TCP which uses multiple congestion signals is part of MS Windows Vista and Windows 2008 TCP stack.

- Network will become more heterogeneous.

Why Study the Heterogeneous Case?

- Various proposals that use different congestion signals

  queueing delay (CARD, DUAL, Vegas, FAST)

  packet loss (Reno and its variants)

  both loss and delay (Westwood, Compound TCP)

  one bit ECN (IETF RFC 2481)

- The Linux operating system already allows users to choose from a variety of congestion control algorithms since kernel version 2.6.13

- Compound TCP which uses multiple congestion signals is part of MS Windows Vista and Windows 2008 TCP stack.

- Network will become more heterogeneous.

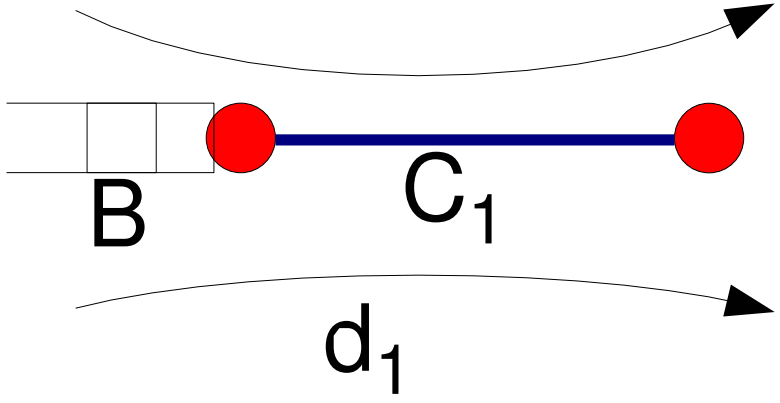
- How do we understand it? How can we manage it?

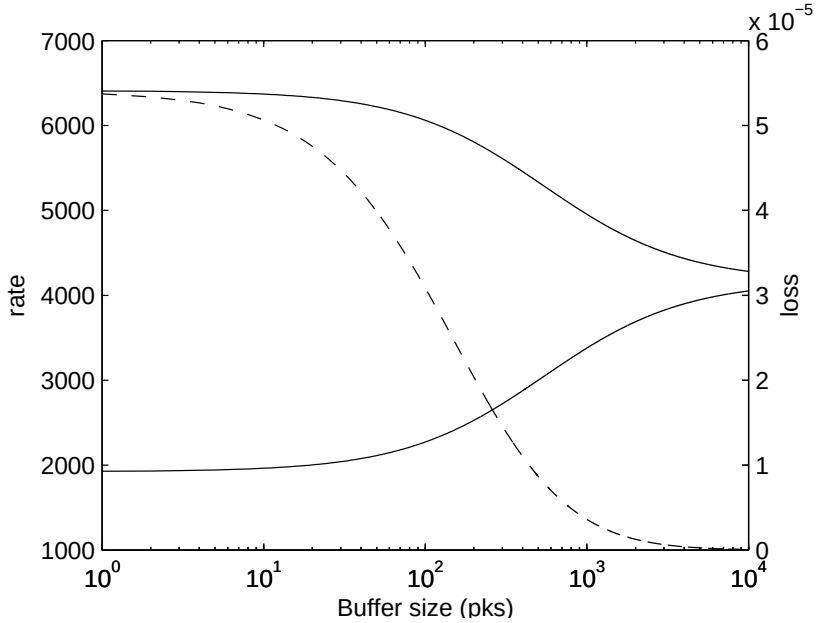
Predictions of the Current Theory

- The equilibrium congestion signals (dual variables) and therefore the equilibrium rates (primal variables) are independent of the link settings.

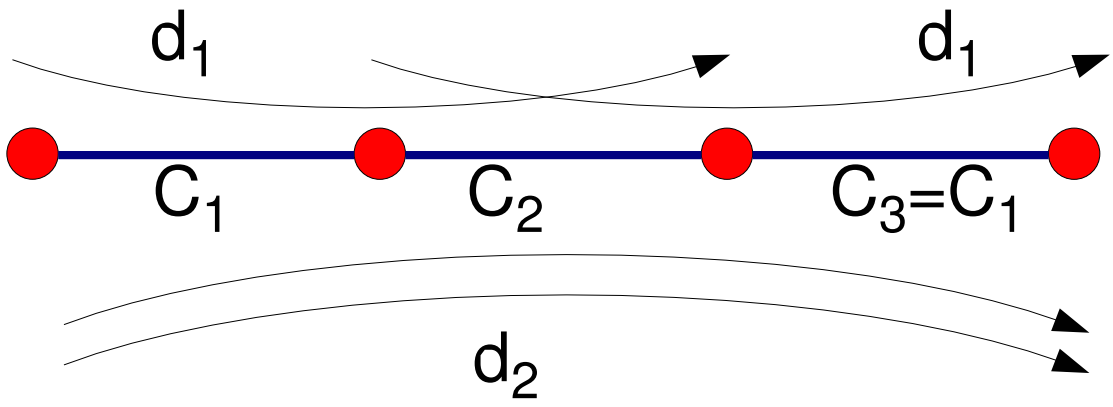
- The system always admits a unique equilibrium.

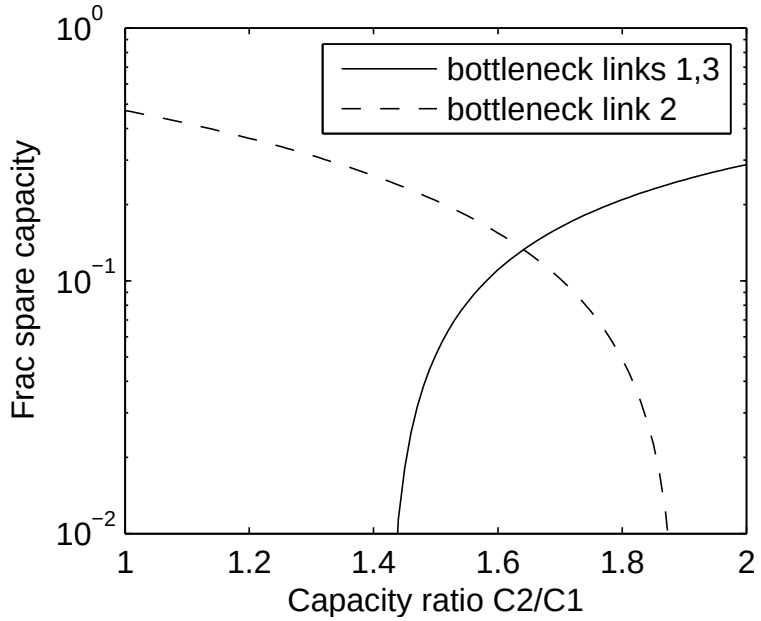
Motivating Example 1: Dependence on Link Parameters





Motivating Example 2: Multiple Equilibrium





Model

Link l: an intrinsic price p_l, other "effective prices" $m_l^j(p_l)$.

e.g., p_l: queue length, p_l^1: loss probability, p_l^2: queueing delay.
With RED algorithm:

$$m_l^1(p_l) = \max(1, kp_l) \qquad m_l^2(p_l) = \frac{p_l}{c_l}$$

Model

Link l: an intrinsic price p_l, other "effective prices" $m_l^j(p_l)$.

e.g., p_l: queue length, p_l^1: loss probability, p_l^2: queueing delay.
With RED algorithm:

$$m_l^1(p_l) = \max(1, kp_l) \qquad m_l^2(p_l) = \frac{p_l}{c_l}$$

Homogeneous case:

$$\dot{x}_i = f_i\left(x_i(t), \sum_{l \in L(i)} p_l(t)\right)$$

Heterogeneous case:

$$\dot{x}_i = f_i\left(x_i(t), \sum_{l \in L(i)} m_l^1(p_l(t)), \ldots, \sum_{l \in L(i)} m_l^J(p_l(t))\right)$$

Partial Utility

$$q_i^j = \sum_{l \in L(i)} m_l^j(p_l)$$

At equilibrium, we have

$$x_i = f_i^0(q_i^1, q_i^2, \ldots, q_i^J)$$

Here, f_i^0 only depends on the TCP algorithm of flow i.

Partial Utility

$$q_i^j = \sum_{l \in L(i)} m_l^j(p_l)$$

At equilibrium, we have

$$x_i = f_i^0(q_i^1, q_i^2, \ldots, q_i^J)$$

Here, f_i^0 only depends on the TCP algorithm of flow i.
It can be arranged as

$$q_i^j = f_i^{-j}(x_i, q_i^{-j}),$$

Define partial utility functions

$$U_i^j(x_i, q_i^{-j}) = \int f_i^{-j}(x_i, q_i^{-j})\, dx.$$

Note that U_j^i is strictly concave increasing, because f_i^{-j} is decreasing.

TCP Reno as an example

Take the standard TCP Reno as an example.

$$x_i = f_i^0(p_i, \tau_i) = \frac{1}{\tau_i + d_i}\sqrt{\frac{2}{p_i}}$$

d_i: the fixed "propagation" delay;
τ_i: the total queueing delay experienced;
p_i: the packet loss probability.

TCP Reno as an example

Take the standard TCP Reno as an example.

$$x_i = f_i^0(p_i, \tau_i) = \frac{1}{\tau_i + d_i}\sqrt{\frac{2}{p_i}}$$

d_i: the fixed "propagation" delay;
τ_i: the total queueing delay experienced;
p_i: the packet loss probability.

$$\begin{aligned} q_i^1 = p_i = f_i^{-1}(x_i, \tau_i) &= \frac{2}{x_i^2(\tau_i + d_i)^2} \\ U_i^1(x_i, \tau_i) &= -\frac{2}{x_i(\tau_i + d_i)^2} \\ q_i^2 = \tau_i = f_i^{-2}(x_i, p_i) &= \frac{1}{x_i}\sqrt{\frac{2}{p_i}} - d_i \\ U_i^2(x_i, p_i) &= \sqrt{\frac{2}{p_i}}\log(x_i) - x_i d_i. \end{aligned}$$

Define a Game: the primal version

J players, with the jth player able to choose p_l^j for each link l, subject to the feasibility constraint

$$Rf_i^0(q_i) \leq c.$$

The payoff for the jth player:

$$\sum_i U_i^j(f_i^0(q_i), q_i^{-j})$$

Define a Game: the primal version

J players, with the jth player able to choose p_l^j for each link l, subject to the feasibility constraint

$$Rf_i^0(q_i) \leq c.$$

The payoff for the jth player:

$$\sum_i U_i^j(f_i^0(q_i), q_i^{-j})$$

In other words, for each $j \in \{1, 2, \ldots, J\}$, the jth player tries to solve

$$\max_{x \geq 0} \sum_i U_i^j(x_i; q_i^{-j}) \text{ subject to } Rx \leq c$$

Define a Game: the dual version

Given that this is a convex optimization with Slater's condition satisfied, strong duality holds and we can also equivalently look at its dual:

$$\min_{p^j \geq 0} \quad \sum_i \max_{x_i \geq 0} \left(U_i^j(x_i; q_i^{-j}) - x_i \sum_l R_{li} p_l^j \right) + \sum_l c_l p_l^j .$$

Define a Game: the dual version

Given that this is a convex optimization with Slater's condition satisfied, strong duality holds and we can also equivalently look at its dual:

$$\min_{p^j \geq 0} \quad \sum_i \max_{x_i \geq 0} \left(U_i^j(x_i; q_i^{-j}) - x_i \sum_l R_{li} p_l^j \right) + \sum_l c_l p_l^j .$$

Theorem 1 *The game always admits at least one pure Nash equilibrium.*

Proof sketch: The joint strategy set is nonempty, convex and compact subset of a Euclidian space. Apply the theorem of Rosen.

Key: Individual strategy set may depend on strategies of other players. Therefore the standard Nash existence theorem does not apply.

J. Rosen. Existence and Uniqueness of Equilibrium Points for Concave N-Person Games *Econometrica*, 33(3):520-534, 1965.

Relation with the network equilibrium

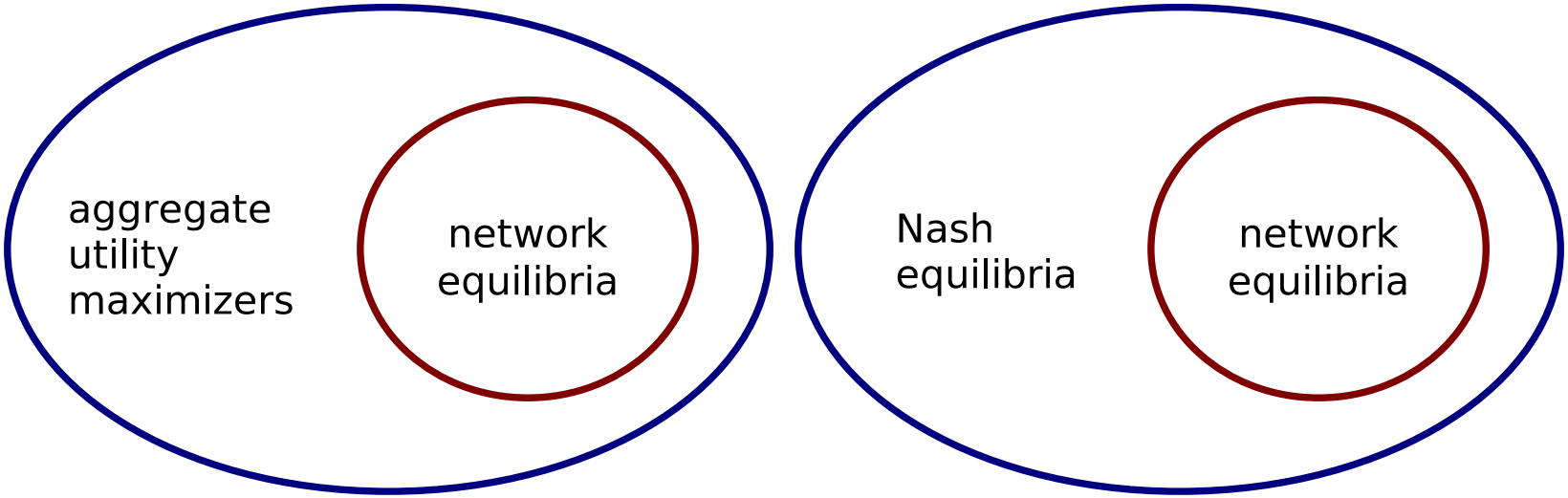
Theorem 2 *All network equilibria are Nash equilibria of the corresponding game.*

Proof is straightforward by the construction of utility functions.

Relation with the network equilibrium

Theorem 3 *All network equilibria are Nash equilibria of the corresponding game.*

Proof is straightforward by the construction of utility functions.



Summary

- Introduction to the internet congestion control and its current theory
- The current theory breaks down with protocols that use heterogeneous signals. We show interesting behaviors both theoretically and experimentally
- Introduce a new framework with game theory.

  All network equilibria is in the set of nash equilibrium of the game.

  Source algorithms (TCP) decide the game and the nash equilibria and link algorithms (AQM) decide which ones are actually network equilibria.

Future Directions

Many problems remain open. In particular,

- How to find nash equilibrium here?
- How to find network equilibria?
- This line of work may interest networking community as well as other communities such as dynamical systems, Operations research, and theoretical computer science.
- Existing closely related work such as monotone dynamical systems, S-modular games and general Nash equilibrium computation may be useful?
- Special structure here?