# Fair and Efficient Router Congestion Control

Xiaojie Gao, Kamal Jain, Leonard J. Schulman

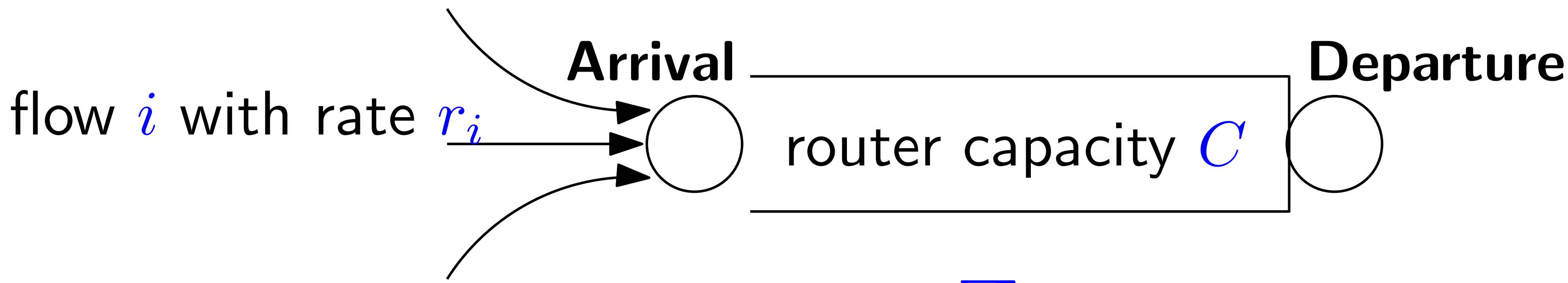
In Proceedings of SODA 2004, Pages 1050-1059.

Presented by ZHOU Zhen, cszz

for COMP 670O Spring 2006, HKUST

# Congestion Control

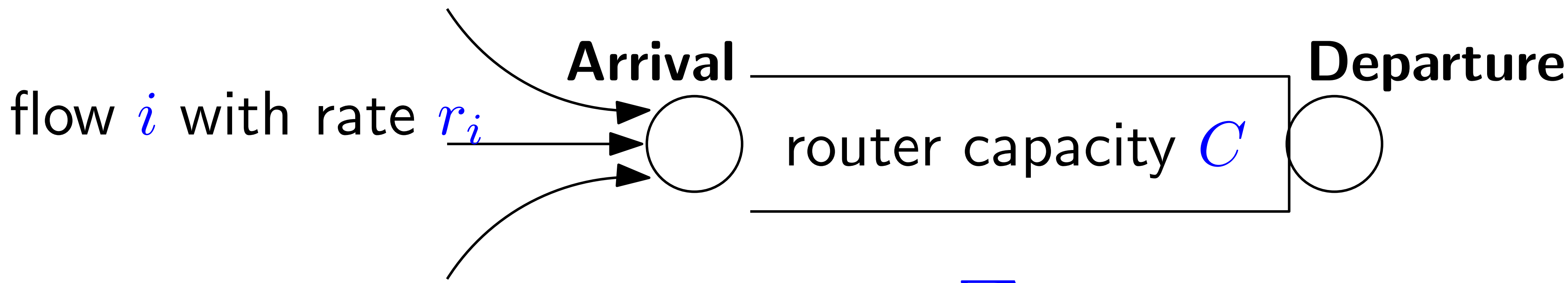
Congestion: queuing system is operated near capacity.




When $\sum_i r_i \geq C$, queue builds up.

# Congestion Control

Congestion: queuing system is operated near capacity.




When $\sum_i r_i \geq C$, queue builds up.

Congestion control at routers: dropping packets.

- Encourage senders to increase their transmission rates, worsening the congestion and destabilizing the system.
- Let a few insistent unresponsive senders take over most of the router capacity.

# Congestion Control: Allocation vs. Penalties

- Assume per flow control (as opposed to stateless control).
- Allocations: allocate to each source its share of the router capacity.

  e.g. Fair queuing: *create separate queues for the packets from each source, and generally forward packets from different sources in round-robin fashion.*

# Congestion Control: Allocation vs. Penalties

- Assume per flow control (as opposed to stateless control).
- Allocations: allocate to each source its share of the router capacity.

  e.g. Fair queuing: *create separate queues for the packets from each source, and generally forward packets from different sources in round-robin fashion.*
- Penalties: discourage aggressive behaviour of sources, by actively penalizing sources that transmit more than their share.

  e.g. CHOKe [Pan, Prabhakar and Psounis. 2000]: *when the queue is long, each incoming packet is compared against another randomly selected packet from the queue; if they are from the same source, both are dropped.*

# Abstract

- Routers set up a game between the senders, who are competing for link capacity.
- Motivation:
  *How to set the rules (i.e. packet-dropping protocol) of a queuing system so that all the users would have a self-interest in controlling congestion when it happens.*
- Mechanism design through auction theory:
  Drop packets of the highest-rate sender, in case of congestion.
  1. Game equilibrium is desirable: high total rate is achieved and is shared widely among all senders.
  2. Equilibrium should be re-established quickly in response to changes in transmission rates

# Outline

- Background and Introduction
- The Protocol
- Computational Performance
- Game-Theoretic Performance
- Network Equilibrium

# Protocol Overview

Sets up a single-item auction for the players (flows), the *highest* bid (flow rate) “wins” the penalty.
So all the senders compete to not be the highest rate flow; this eliminates the congestion.

# Protocol Overview

Sets up a single-item auction for the players (flows), the *highest* bid (flow rate) "wins" the penalty.
So all the senders compete to not be the highest rate flow; this eliminates the congestion.

Protocol parameters:

- $Q$, the total number of packets in the queue.
- $m_i$, the total number packets in the queue from source $i$.
- A hash table for each flow $i$ presently in the queue.
- $MAX$, a pointer to the leading source in the hash table.
- Buffer indicators: $F$ for full, $H$ for high, $L$ for low.

# Protocol Recipe (1)

**On packet arrivals**

1. The packet source $i$ is identified.
2. (a) If $Q > H$, stamp the packet $DROP$;

   (b) otherwise if $H \geq Q > L$ and $i = MAX$, stamp the packet $DROP$;

   (c) otherwise, stamp the packet $SEND$.
3. The packet is appended to the tail of the queue, together with the stamp. (If the stamp is $DROP$, the packet data can be deleted, but the header is retained so long as the packet is on the queue.)
4. $Q$ and $m_i$ are incremented by $1$. (If $m_i$ was $0$, a new record is created in the hash table.)
5. If $m_i > m_{MAX}$, then the $MAX$ pointer is reassigned the value $i$.

# Protocol Recipe (2)

**On packet departures**

1. The packet source $i$ is identified.
2. $Q$ and $m_i$ are decremented by $1$. (If $m_i$ was $0$, a record is eliminated in the hash table.)
3. If the packet is stamped $SEND$, it is routed to its destination; otherwise, it is dropped.
4. If $i = MAX$ but $m_i$ is no longer maximal, $MAX$ is reassigned to the maximal sender.

# Protocol Variant

**On packet arrivals**

1. The packet source $i$ is identified.
2. (a) If $Q > H$, stamp the packet $DROP$;
   (b) * If $H \geq Q > L$ and $m_i \geq \frac{H-Q}{H-L} m_{MAX}$, stamp the packet $DROP$.
   (c) otherwise, stamp the packet $SEND$.

...

The variant has no advantage over the original protocol with regard to Nash equilibria. However, its sliding scale for drops has a substantial advantage over both CHOKe and the original protocol in the effectiveness with which multiple unresponsive flows are handled.

# Outline

- Background and Introduction
- The Protocol
- Computational Performance

  *How much computational resources are required for executing the protocol at each router?*
- Game-Theoretic Performance
- Network Equilibrium

# Computational Performance

- Time complexity: $O(1)$ per packet
  On each arrival/departure, we need to update $m_i$. They are grouped and arranged in an accessory doubly-linked list on sorted $m_i$ values for indexing source $i$.

- Space complexity: for storing the hash table
  Linear space & constant access time (FKS hash table [Fredman, Komlós and Szemeréi. 1984]).
  Linear space & logarithmic access time (balanced binary search tree on source labels).

- Distributed protocol.

# Outline

- Background and Introduction
- The Protocol
- Computational Performance
- Game-Theoretic Performance

  *How the game theoretical requirements (fairness) are met by the proposed protocol? How the equilibrium is reached?*
- Network Equilibrium

# More Notations

- $C$, the capacity of the router.
- $r_i$, the desired transmission rate of source $i$.
- $R_i$, the max-min fairness rate of the source $i$.
- $s_i$, be the actual Poisson rate chosen by source $i$.
- $a_i$, the throughput of source $i$.
- $B$ Poisson sources with rates $s_1 \geq s_2 \geq \cdots \geq s_B$.
- $\tilde{B} = \min_j \left\{ \sum_{i=1}^{j} s_i \geq \frac{1}{2} \sum_{i=1}^{B} s_i \right\}$, an undercount of the number of sources. It counts only the sources contributing a substantial fraction of the total traffic. (arg min?)

# Fairness Objective

Objectives:
The achieved rates of the various sources should be fair: high-volume sources should not be able to crowd out low-volume sources.

Max-min fairness:
$R_i = \min\{r_i, \alpha\}$ where $\alpha = \alpha(C, \{r_i\})$ is the supremum of the values for which $\sum_i R_i < C$. (This includes the possibility $\alpha = \infty$ if $\sum_i r_i < C$.)

# Fairness Objective

Objectives:
The achieved rates of the various sources should be fair: high-volume sources should not be able to crowd out low-volume sources.

Max-min fairness:
$R_i = \min\{r_i, \alpha\}$ where $\alpha = \alpha(C, \{r_i\})$ is the supremum of the values for which $\sum_i R_i < C$. (This includes the possibility $\alpha = \infty$ if $\sum_i r_i < C$.)

A set of rates $\{R_i\}$ is said to be *max-min fair* if it is feasible and for each $i \in B$, $R_i$ cannot be increased while maintaining feasibility without decreasing $R_{i'}$ for some flow $i'$ for which $R_{i'} \leq R_i$. [Bertsekas and Gallager. *Data Networks*].

E.g. Given $C = 4$, $r_1 = 1$, $r_2 = 2$ and $r_3 = 3$,
Results in $R_1 = 1$, $R_2 = 1.5$ and $R_3 = 1.5$. ($\alpha = 1.5$.)

# Equilibrium Guarantees (B1)

B1. Assume an idealized situation in which the router, and all the sources, know the source rates $\{s_i\}$; and in which the queue buffer is unbounded. This idealized game should have a unique Nash equilibrium which is the max-min-fairness rates $R_i$ as determined by the inputs $C$ and $\{r_i\}$.

# Equilibrium Guarantees (B1)

B1. Assume an idealized situation in which the router, and all the sources, know the source rates $\{s_i\}$; and in which the queue buffer is unbounded. This idealized game should have a unique Nash equilibrium which is the max-min-fairness rates $R_i$ as determined by the inputs $C$ and $\{r_i\}$.

If $\sum s_i > C$, the router simply drops all the packets of the highest-rate source; if several tie for the highest rate, it rotates randomly between them.

Theorem. If the sources have desired rates $\{r_i\}$ for which $r_i > C$, and can transmit at any Poisson rates $0 \leq s_i \leq r_i$, then their only Nash equilibrium is to transmit at rates $s_i = R_i$.

Because no source could do better by transmitting at a rate $s_i \geq \alpha$.

# Equilibrium Guarantees (B2)

Assume the buffer is large enough so that $L \in \Omega(\tilde{B}\frac{1}{\epsilon^2}\ln\frac{1}{\epsilon})$.

Theorem. If the protocol is administering traffic from sources (by a router that can only use its history to govern its actions) with desired rates $\{r_i\}$ for which $\sum r_i > C$, and which can transmit at any Poisson rates $0 \le s_i \le r_i$, there is a small $\epsilon > 0$ such that any source sending at rate $s_i \le (1-\epsilon)\alpha$, will achieve throughput $a_i \ge s_i(1-\epsilon)$.

The hypotheses guarantee a ratio of at most $1-\epsilon$ between $s_i$ and $\alpha$. Packets from source $i$ will be dropped only when the queue grows to size $L$, and $m_i$ is higher than that of all other sources.

# Variable-Rate Transmission (B3)

Theorem. For a source which is allowed to send packets at arbitrary times (while all other sources are restricted to being Poisson), the long-term throughput is no more than $1 + \epsilon$ times that of the best Poisson strategy ($s_1$).

# Variable-Rate Transmission (B3)

Theorem. For a source which is allowed to send packets at arbitrary times (while all other sources are restricted to being Poisson), the long-term throughput is no more than $1 + \epsilon$ times that of the best Poisson strategy ($s_1$).

Proof: Assume packets from $S$ were dropped befor but start to be sent after $T$.

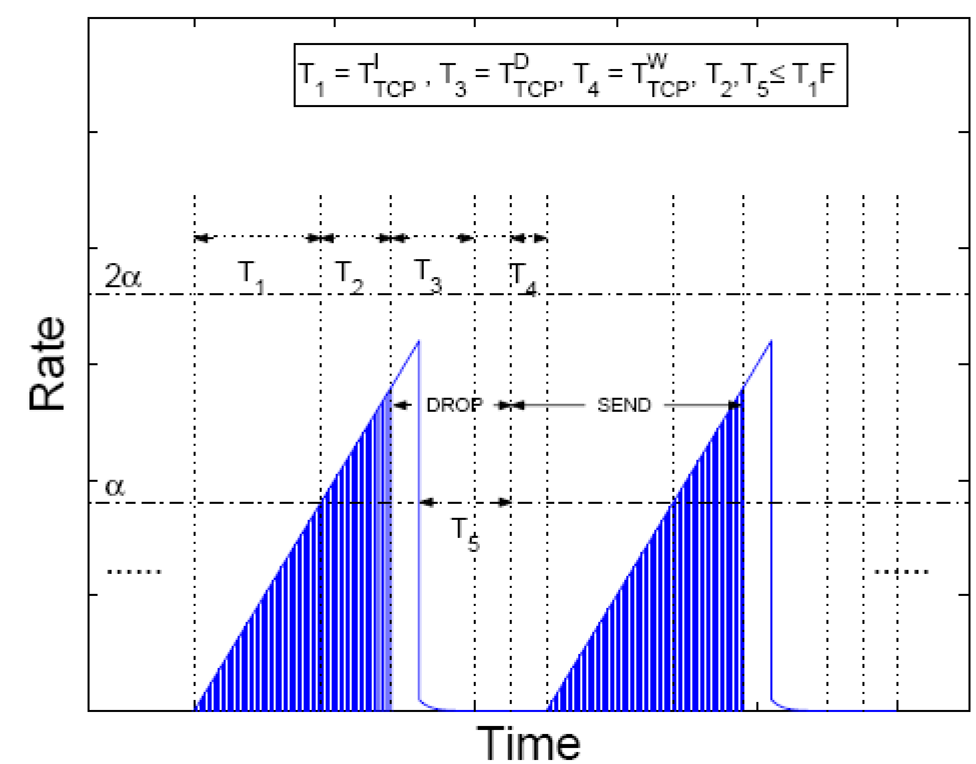
Consider any time interval $[t_i, t'_i] \subset [T, T']$ when packet $i$ from $S$ is in the queue. Since that packet is marked $SEND$, then $m_S < MAX_{t_i}$. Split $[T, T']$ into $n + 1$ sub-intervals. Therefore (let $t'_0 = T'$), the number of sent packets is no more than
$MAX_{t_0}(t'_0 - t'_1) + MAX_{t_1}(t'_1 - t'_2) + \cdots + MAX_{t_{n-1}}(t'_{n-1} - t'_n)$.

Since in any $[t'_{i+1}, t'_i]$, $E[MAX_{t_i}] \leq s_1(t'_i - t'_{i+1})(1 + \epsilon)$,
then $a_S \leq (1 + \epsilon)s_1$.

# Performance of TCP (B4)

B4. Under certain assumptions on the timing of acknowledgments, the throughput of a TCP source with unbounded desired rate, playing against Poisson sources with desired rates $r_2, r_3, \ldots$, is within a constant factor of the max-min-fairness value $\alpha(C, \{\infty, r2, r3, \ldots\})$.

The reason that TCP interacts so well with the protocol is that it backs off very quickly from congestion, and therefore, will quickly stop being "punished" by the protocol; and that it subsequently "creeps" up toward the max-min-fairness threshold $\alpha$ (before again having to back off).




Not typical multiplicative-decrease TCP.

# Outline

- Background and Introduction
- The Protocol
- Computational Performance
- Game-Theoretic Performance
- Network Equilibrium

  *What the Nash equilibria are in the case of a general network, with several flows traveling across specified routes, and with the protocol implemented at each router of the network.*

# Network Equilibrium

In any equilibrium, it is advantageous for every source to be transmitting at no more than its throughput.

Lemma. If each flow is sending at its throughput, then there is a unique Nash equilibrium, the “max-min-fairness” allocation.

Proof: Max-min allocation is unique for finite number of flows and routers.
Assume $\vec{x}$ is the N.E. allocation and $\vec{y}$ is any other allocation.
If $\exists s$ s.t. $y_s > x_s$, then $x_s < r_s$. Then by N.E., $x_s$ is the highest rate at some router. So $\exists t \neq s$, s.t. $y_t < x_t \leq x_s$, (as the total capacity is constant).

Therefore $\vec{x}$ is the max-min allocation (maximizing the minimum).

# Network Equilibrium

In any equilibrium, it is advantageous for every source to be transmitting at no more than its throughput.

Lemma. If each flow is sending at its throughput, then there is a unique Nash equilibrium, the "max-min-fairness" allocation.

Proof: Max-min allocation is unique for finite number of flows and routers.
Assume $\vec{x}$ is the N.E. allocation and $\vec{y}$ is any other allocation.
If $\exists s$ s.t. $y_s > x_s$, then $x_s < r_s$. Then by N.E., $x_s$ is the highest rate at some router. So $\exists t \neq s$, s.t. $y_t < x_t \leq x_s$, (as the total capacity is constant).

Therefore $\vec{x}$ is the max-min allocation (maximizing the minimum).

Theorem. For any collection of network flows there is a unique Nash equilibrium, equal to the "max-min-fairness" allocation; in this equilibrium there are no packet drops.

# The End

Thank you.