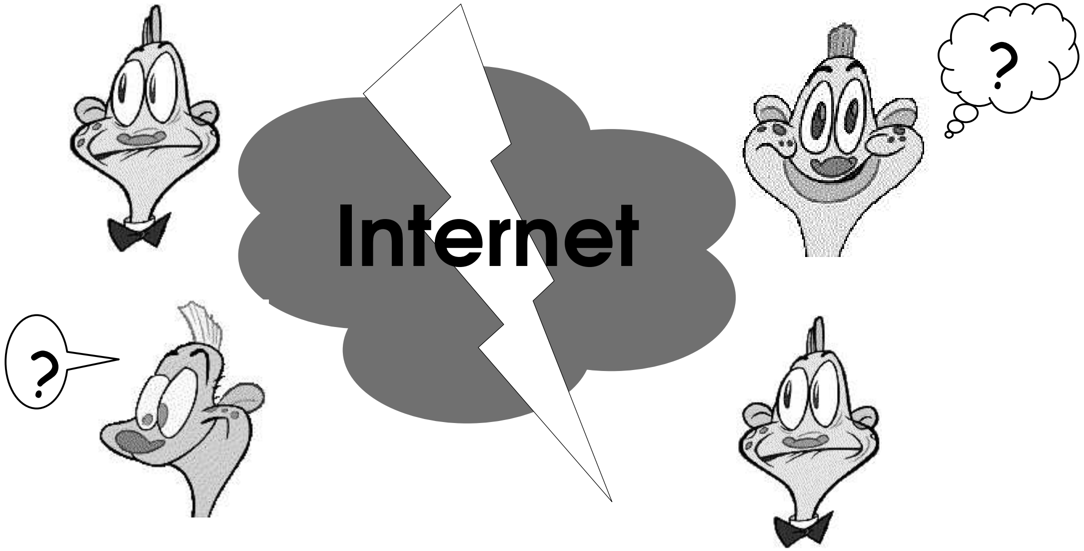
A Game-Theoretic Analysis of TCP

Aditya Akella, Richard Karp,
Christos Papadimitriou, Srinivasan Seshan,
Scott Shenker

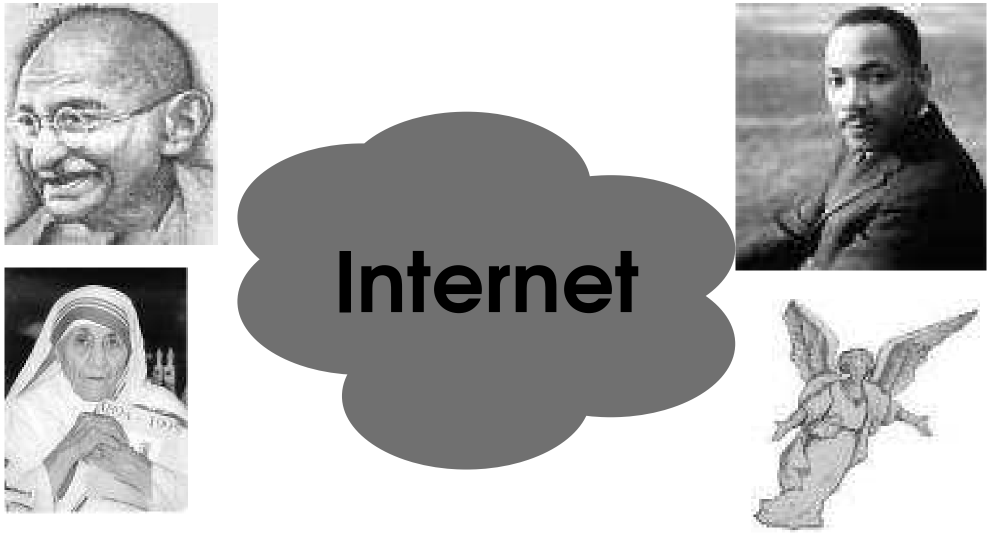
CMU/ICSI/UC Berkeley

1980s: Network Collapse



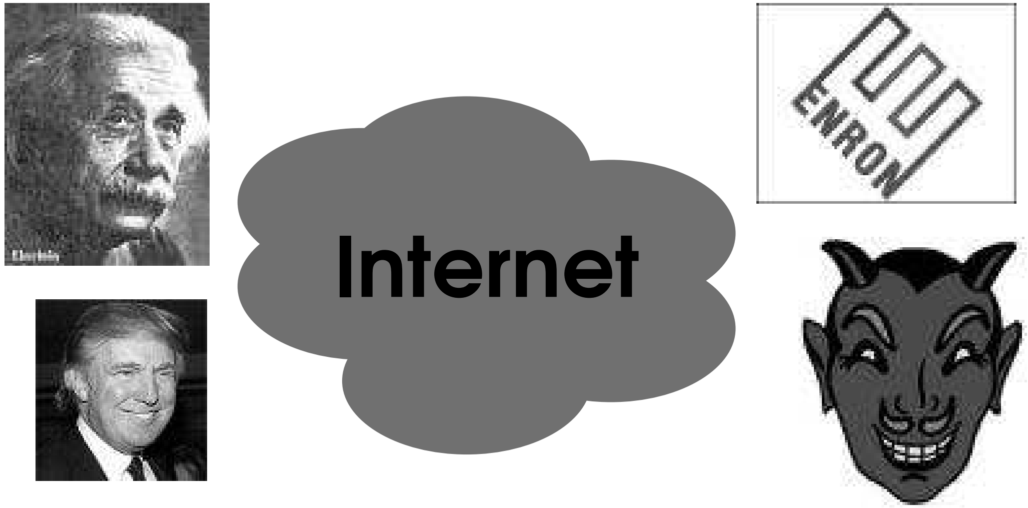
In the 80s's, naïve behavior caused the network to collapse

Salvation!



Socially responsible congestion control implemented at end-points was given credit for saving the Internet

Salvation?



Can the network survive with greedy (but intelligent) behavior?

Aditya Akella (aditya@cs.cmu.edu)

Why Bother?

- If greed is bad, today's Internet is stable because –
    - End-points are consciously social and/or
    - It is hard to modify end-hosts to be greedy

We may need mechanisms to monitor, dissuade aggressive behavior

- If not, we need no such mechanism
    - Can rely on end-point behavior for efficient operation

Outline

- The TCP Game
- Results for the TCP Game
- Mechanisms for Nash Equilibrium

The TCP Game

- *TCP Game*
  - Flows attempt to maximize application throughput
    - Flows modify their AIMD parameters (α, β)
    - Must still provide reliability
- What happens at *Nash Equilibrium*?
  - No flow can gain in throughput by unilaterally changing its parameter choice

The TCP Game

- Analyze a simplified version of this Game for…
  - Parameters at the Nash Equilibrium
  - Efficiency at the Nash Equilibrium
    - Link Goodput and per-flow Loss rate
- Study symmetric Nash Equilibria only

Factors Affecting the Nash Equilibrium

(1) End-point's loss recovery mechanism

- Reno vs. SACK (primitive vs. modern)
- Depends on TCP implementation

(2) Loss assignment at routers

- Bursty loss assignment vs. randomized uniform losses

(3) Congestion control parameters of the flows

- How flows are allowed to vary their parameters
- Under complete control of end-point

End-points show greed by adjusting factor (3) alone.
Factors (1), (2) are part of the environment.

Outline

- The TCP Game
- Results for the TCP Game
- Mechanisms for Nash Equilibrium

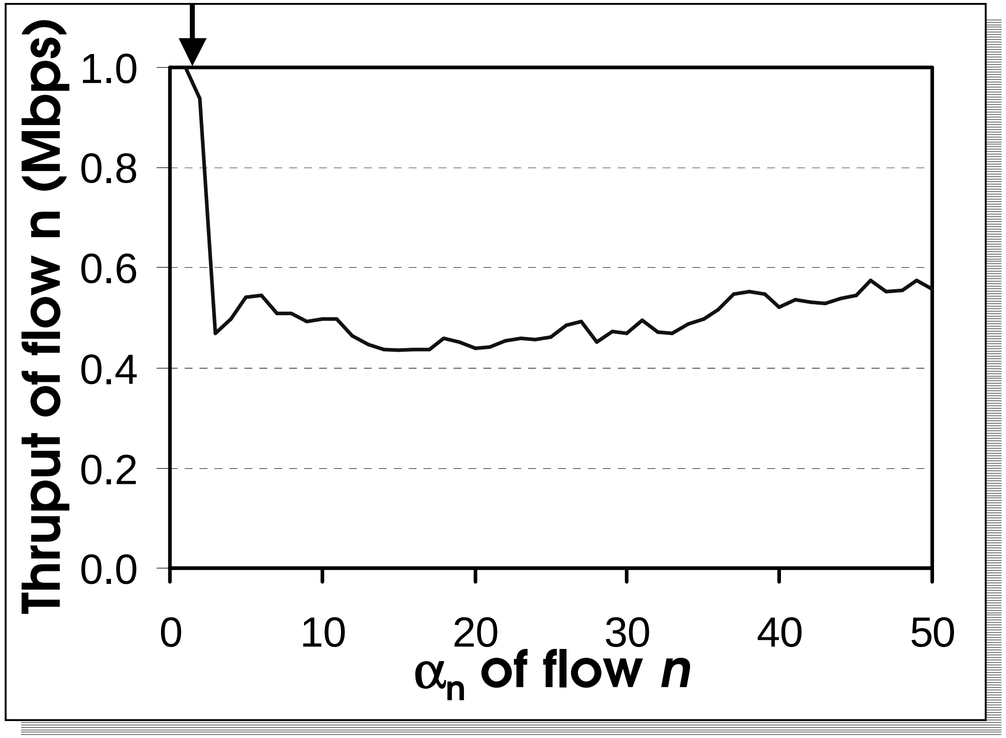
Results – Road-map

- First, consider FIFO droptail buffers
    - Most wide-spread in today's Internet
    - Efficiency at Nash Equilibrium for Tahoe, Reno, SACK-style loss recovery
- Then, discuss RED buffers briefly
    - As above
- Put the results together

FIFO Droptail Buffering

- Droptail buffers punish bursty behavior
    - Unintentionally so
    - Observed by designers of RED AQM
    - Flows with bursty transmission incur losses proportional to their burstiness
        - AIMD flows incur losses in bursts of size ~ α (AI parameter)

Results for FIFO Droptail Buffers – A Sample



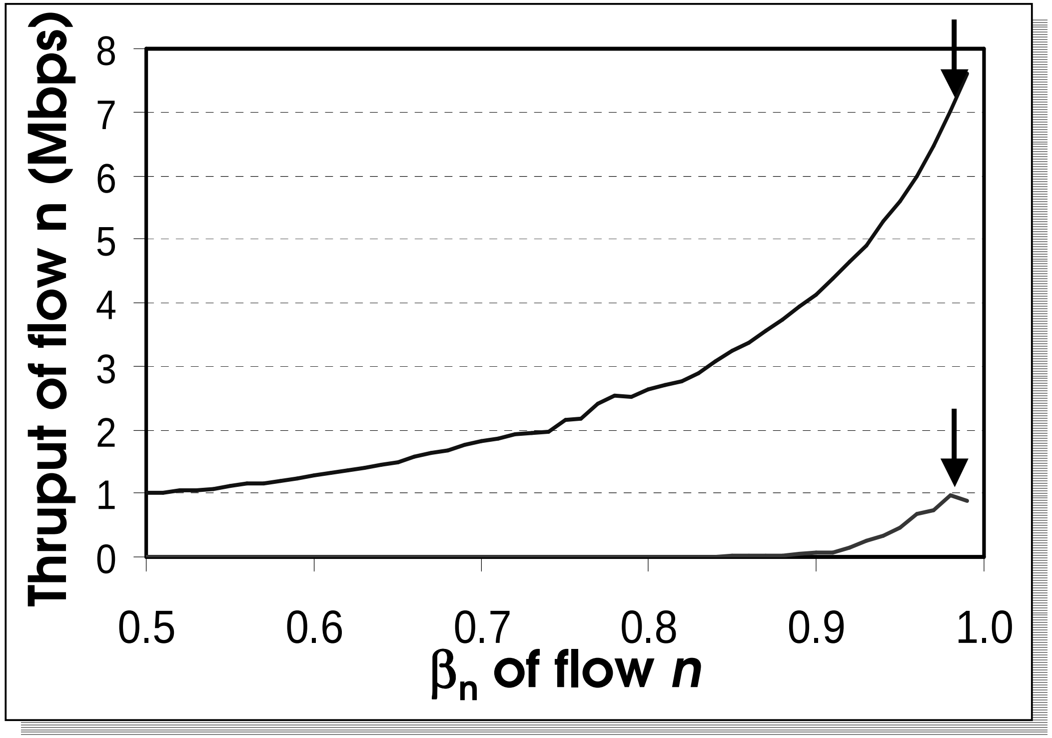
Reno-style loss recovery (flows vary α, keeping $\beta=0.5$)

α_E	Link Goodput	Per-flow Loss rate
1	100%	0.15%

- Greedy flows don't gain by using large α's
  - Flows observe burst losses
  - Reno's severe reaction (time-outs) kicks in

Aditya Akella (aditya@cs.cmu.edu)

Results for FIFO Droptail Buffers – A Sample



Reno-style loss recovery (flows vary β, keeping $\alpha=1$)

β_E	Link Goodput	Per-flow Loss rate
0.98	100%	1.5%

- Greedy flows gain by using $\beta \rightarrow 1$
  - No burst losses since $\alpha=1$

Results for FIFO Droptail Buffers – A Sample

Reno-style loss recovery
(flows vary α, β together)

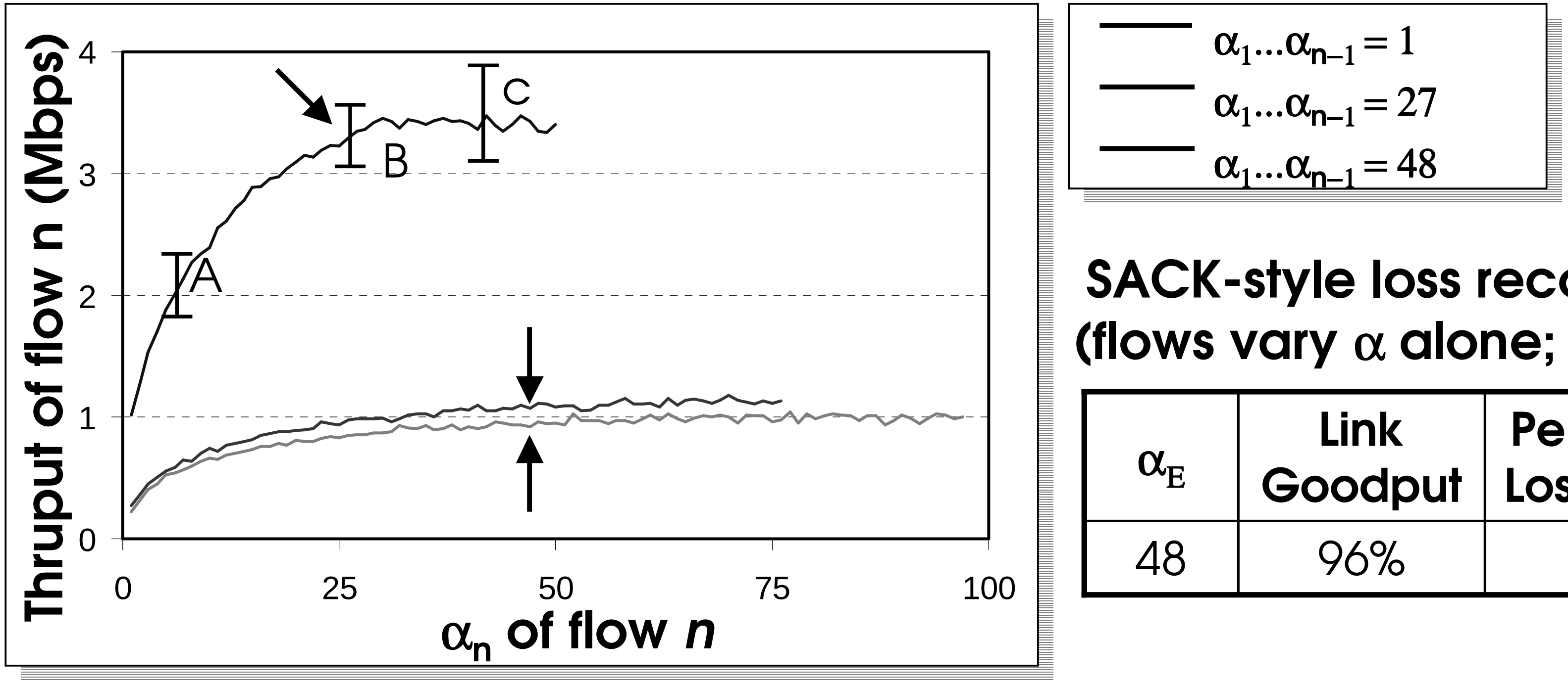
(α_E, β_E)	**Link Goodput**	**Per-flow Loss rate**
(1, 0.98)	**100%**	**1.5%**

- Nash Equilibrium is efficient!
  - Goodput is high and loss rate is low
  - Greedy behavior might work out
- But unfair
  - Since $\beta \rightarrow 1$ (AIAD)

RED Buffering

- RED buffers “spread” losses uniformly across flows
  - Identical loss %-age across flows irrespective of parameters used
  - Greater greed of a few flows causes a small increase in overall loss rate
  - Bursty flows do not experience burst losses, unlike droptail buffers

Results for RED Buffers – A Sample



SACK-style loss recovery (flows vary α alone; $\beta=0.5$)

α_E	Link Goodput	Per-flow Loss rate
48	96%	5%

- Aggression is always good
    - TCP SACK ➔ high loss rate doesn't affect goodput
    - RED ➔ greater aggression will cause minor increase in overall loss rate

Aditya Akella (aditya@cs.cmu.edu)

Results for RED Buffers – A Sample

SACK-style loss recovery

Flows vary α,β together

(α_E, β_E)	**Link Goodput**	**Per-flow Loss rate**
(23, 0.98)	96%	5.70%

- Nash Equilibrium is inefficient
  - Parameter setting is very aggressive
    - Loss rate is high
  - Potential congestion collapse!

Aditya Akella (aditya@cs.cmu.edu)

Results for the TCP Game – A Summary

	Droptail	**RED**
Tahoe	$\alpha_E \rightarrow$ high, $\beta_E \rightarrow 1$, **Inefficient**	$\alpha_E \rightarrow$ high, $\beta_E \rightarrow 1$, **Inefficient**
Reno	$\alpha_E = 1$, $\beta_E \rightarrow 1$, **Efficient, Unfair**	$\alpha_E \rightarrow$ high, $\beta_E \rightarrow 1$, **Inefficient**
SACK	$\alpha_E \rightarrow$ high, $\beta_E \rightarrow 1$, **Inefficient**	$\alpha_E \rightarrow$ high, $\beta_E \rightarrow 1$, **Inefficient**

Aditya Akella (aditya@cs.cmu.edu)

Discussion

Question: Does selfish congestion control endanger network efficiency?

Common Intuition: Yes, since flows would always gain from being more aggressive.

Our Answer: Not necessarily true!

- In the traditional setting (Reno end-points and droptail routers), network operates fine despite selfish behavior
- Selfish behavior very detrimental with modern loss recovery and queue management schemes

Outline

- The TCP Game

- Results for the TCP Game

- Mechanisms for Nash Equilibrium

Mechanisms for Nash Equilibrium

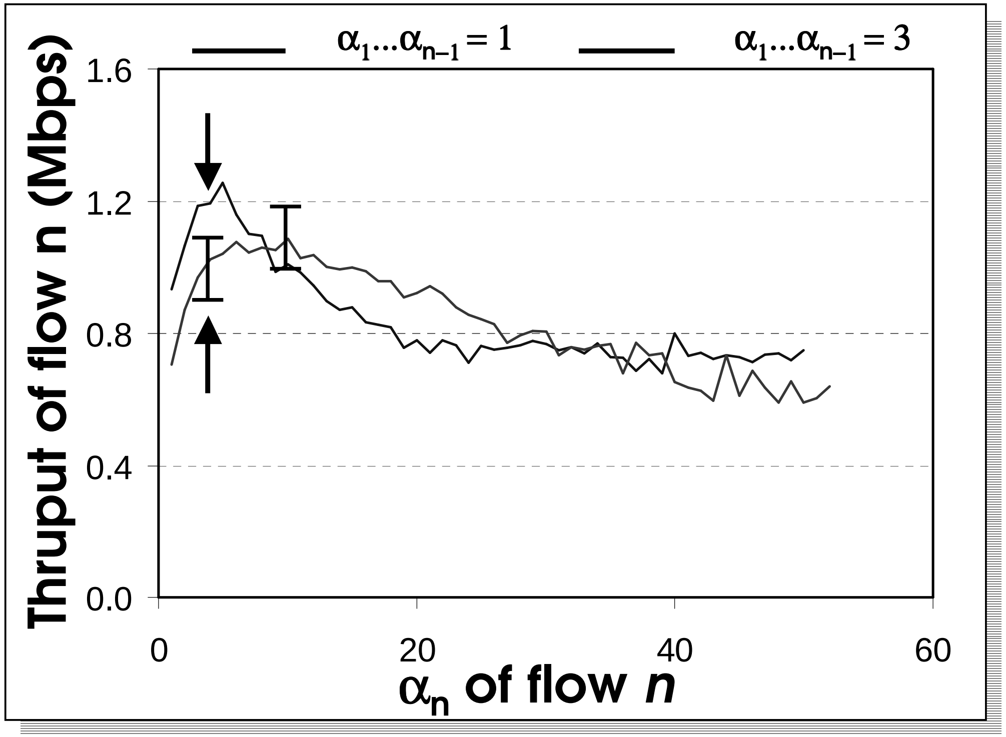
- We need mechanisms to explicitly counter aggressive behavior
- Has been a hot topic in the past
  - Fair Queuing discourages aggressive behavior
    - But needs per-flow state
  - RED-PD, AFD etc. explored lighter mechanisms
    - Aim to ensure *fair* bandwidth allocation

Our requirement is less stringent:

How much preferential dropping is needed to ensure a *reasonable* Nash Equilibrium?

CHOKe+ -- A Simple, Stateless Scheme

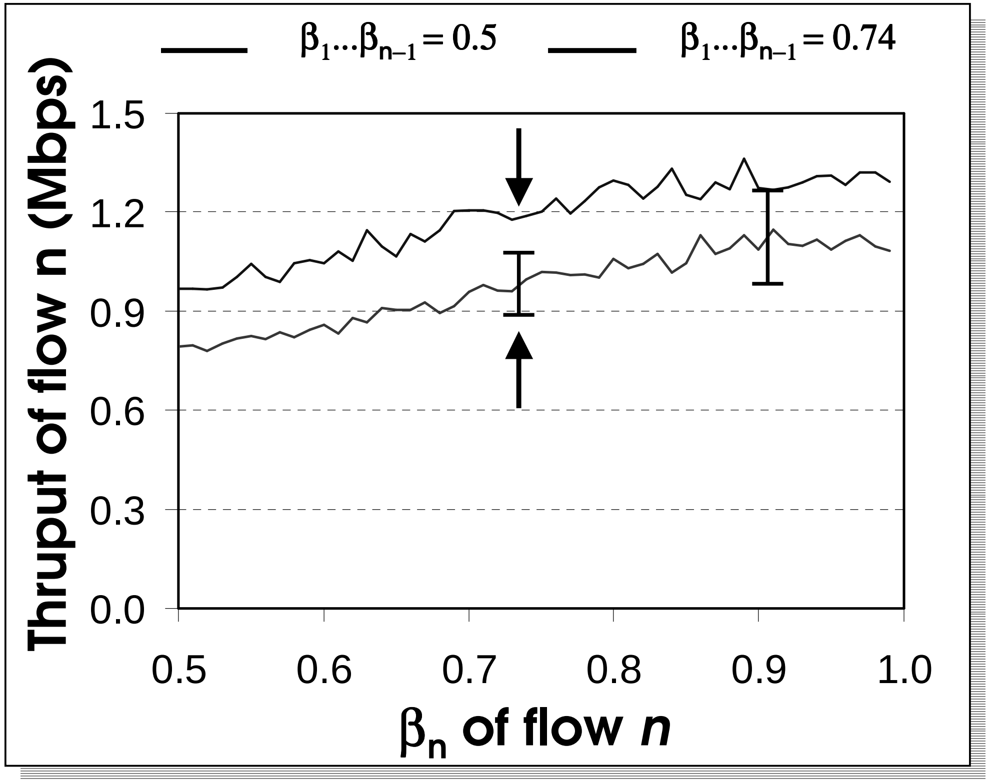
- A small modification to RED is enough
- CHOKe+
  - Simple, stateless
  - Provides just the right amount of punishment to aggressive flows
  - Makes marginal advantage from greed insignificant
  - E.g. SACK flows varying α



α_E	Goodput	Loss rate
3	97%	2.74%

CHOKe+ (Cont.)

- $\beta \rightarrow 1$ at Nash Equilibrium in all cases
  - $\beta < 1$ impossible to ensure without Fair Queuing
- But, CHOKe+ encourages $\beta < 1$
  - Makes aggressive β a risky choice
  - With SACK flows $\beta = 0.74$ at Nash Equilibrium



β_E	**Goodput**	**Loss rate**
0.74	100%	2.42%

Summary

- Greedy congestion control may not always lead to inefficient operation
  - Traditional Reno host-droptail router setting
- Unfortunately, greedy behavior is bad in most other situations
- Fortunately, it is possible to ensure a desirable Nash Equilibrium via simple, stateless mechanisms

Back-up

- Back-up
  - Back-up
    - Back-up

CHOKe+

- CHOKe would have worked
  - But, enforces too high a drop rate
  - Underutilization at low levels of multiplexing
  - CHOKe+ fixes this problem

The CHOKe+ Algorithm

- For each incoming packet P
  - Pick *k* packets at random from queue
  - Let *m* be # packets from the same flow as P
  - Let $0 <= \gamma_2 < \gamma_1 <= 1$ be constants
  - If $m > \gamma_1 k$, P and the *m* packets are dropped
  - Else if $\gamma_2 k <= m < \gamma_1 k$, drop P and the *m* packets only if RED were to drop P
  - Else just drop P according to RED

Aditya Akella (aditya@cs.cmu.edu)

Why AIMD?

- Analysis is more generic than meets the eye
  - Conclusions hold for other congestion control schemes
  - Burstiness is a property of probing
- Widely employed

Why not Change Loss Recovery?

- Historical evaluation
- Very difficult to change
  - Sometimes need bilateral (protocol) support
  - Needs many implementation changes
  - Many design decisions were influenced by system requirements