Equilibrium and Dynamics of TCP

John Doyle
Steven Low
Caltech

Fernando Paganini
UCLA

April 2001

End-to-end Control

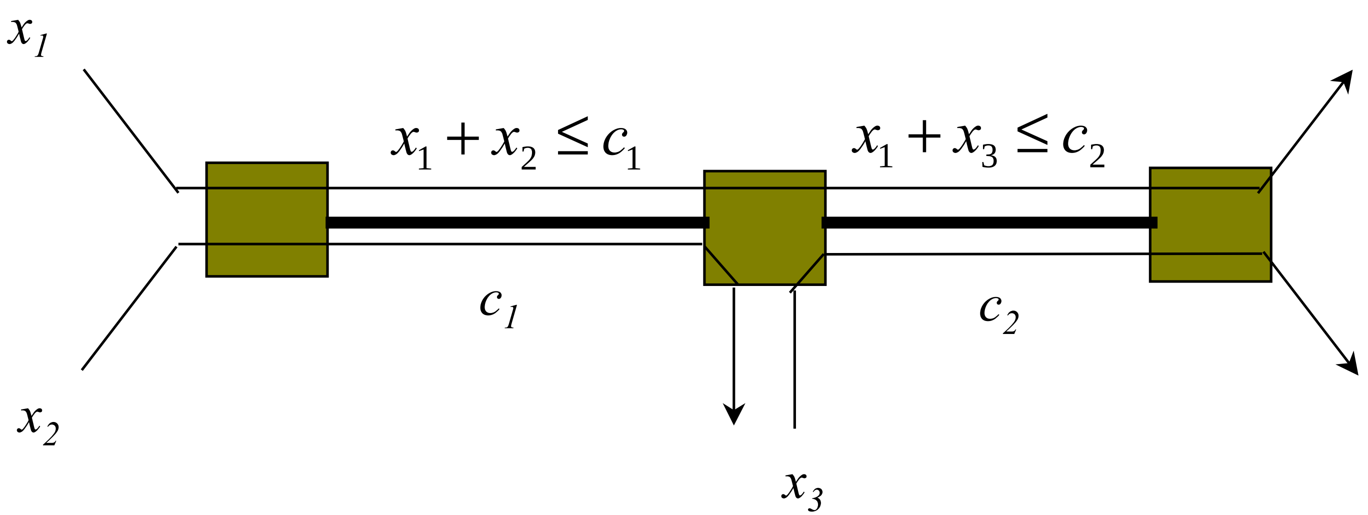
- Heavy tail → Mice-elephants
  - End-to-end: control elephants (not mice!)
  - Yet determines delay & loss for mice
- Outline
  - Duality model
  - Dynamic model
  - REM: Active Queue Management

Flow control

- Interaction of source rates $x_s(t)$ and congestion measures $p_l(t)$
- Duality theory
  - They are primal and dual variables
  - Flow control is optimization process
- Example congestion measure
  - Loss (Reno)
  - Queueing delay (Vegas)
  - Queue length (RED)
  - Price (REM)

Model

- Sources s
  - $L(s)$ - links used by source s
  - $U_s(x_s)$ - utility if source rate = x_s
- Network
  - Links l of capacities c_l



Primal problem

$$\max_{x_s \geq 0} \quad \sum_s U_s(x_s)$$

$$\text{subject to} \quad x^l \leq c_l, \quad \forall l \in L$$

- Assumptions
  - Strictly concave increasing U_s
- Unique optimal rates x_s exist
- Direct solution impractical

Duality Approach

$$\text{Primal:} \quad \max_{x_s \geq 0} \sum_s U_s(x_s) \quad \text{subject to } x^l \leq c_l, \quad \forall l \in L$$

$$\text{Dual:} \quad \min_{p \geq 0} \quad D(p) = \left(\max_{x_s \geq 0} \sum_s U_s(x_s) + \sum_l p_l (c_l - x^l) \right)$$

Primal-dual algorithm:

$$x(t+1) = F(p(t), x(t))$$

$$p(t+1) = G(p(t), x(t))$$

Duality Model of TCP

- Source algorithm iterates on rates
- Link algorithm iterates on prices
- With different utility functions

Primal-dual algorithm:

$$x(t+1) = F(p(t),\ x(t))$$

$$p(t+1) = G(p(t),\ x(t))$$

Reno, Vegas → $x(t+1) = F(p(t),\ x(t))$

DropTail, RED, REM → $p(t+1) = G(p(t),\ x(t))$

Active Queue Management

	$p_l(t)$	$G(p(t), x(t))$
DropTail	loss	$[1 - c_l/x^l(t)]^+$ (?)
RED	queue	$[p_l(t) + x^l(t) - c_l]^+$
Vegas	delay	$[p_l(t) + x^l(t)/c_l - 1]^+$
REM	price	$[p_l(t) + \gamma(\alpha_l\, b_l(t) + x^l(t) - c_l)]^+$

$x(t+1) = F(\, p(t), x(t)\,)$ ← Reno, Vegas

$p(t+1) = G(\, p(t), x(t)\,)$ ← DropTail, RED, REM

Duality model of TCP

- Reno utility function

$$U_s^{reno}(x_s) = \frac{\sqrt{2}}{D_s} \tan^{-1}\left(\frac{x_s D_s}{2}\right)$$

- Vegas utility function

$$U_s^{vegas}(x_s) = \alpha_s d_s \log x_s$$

Example

- Basic algorithm

source : $x_s(t) = U_s^{'-1}(q_s(t))$

link: $p_l(t) = \gamma(y_l(t) - c_l)$

Theorem (ToN'99)

Converge to optimal rates in asynchronous environment, provided γ sufficiently small

TCP are smoothed versions of source algorithm

(but AQM is generally not the link algorithm above)

Duality model

- Reno utility function

$$U_s^{reno}(x_s) = \frac{\sqrt{2}}{D_s}\tan^{-1}\left(\frac{x_s D_s}{2}\right)$$

- Vegas utility function

$$U_s^{vegas}(x_s) = \alpha_s d_s \log x_s$$

- Reno/DropTail, Reno/RED, Reno/REM

$$x_s(t+1) = \left[x_s(t) + \frac{m(p(t))}{2}(\bar{x}_s^2(t) - x_s^2(t))\right]^+$$

$U_s'^{-1}(p(t))$

Summary

- Flow control problem

$$\max_{x \geq 0} \quad \sum_s U_s(x_s)$$

$$\text{subject to} \quad Rx \leq c$$

- Primal-dual algorithm

$$\dot{x}(t) = F(p(t), x(t))$$ ← Reno, Vegas

$$\dot{p}(t) = G(p(t), x(t))$$ ← DropTail, RED, REM

- Major TCP schemes
  - Maximize aggregate source utility
  - With different utility functions

Outline

- Duality model
- Dynamic model
- REM: Active Queue Management

Duality model - AIMD

AI | MD

$$\dot{x}_i = x_i(t-\tau_i)(1-q_i(t))\frac{1}{\tau_i^2 x_i(t)} - x_i(t-\tau_i)q_i(t)\frac{x_i(t)}{2}$$

source rate

end-to-end prob

$$q_i(t) = \sum_l m_l(t-\tau_{li}^b)$$

Linear model - AIMD

AI MD

$$\dot{x}_i = x_i(t-\tau_i)(1-q_i(t))\frac{1}{\tau_i^2 x_i(t)} - x_i(t-\tau_i)q_i(t)\frac{x_i(t)}{2}$$

Linearize around equilibrium

$$\dot{x}_i = -x_i q_i x_i(t) - \frac{1}{\tau_i^2 q_i} q_i(t)$$

In Laplace domain

$$x_i(s) = -\frac{1}{\tau_i^2 q_i}\frac{1}{s + x_i q_i} q_i(s)$$

Duality model - AQM

$$\dot{\lambda}_l = y_l(t) - c_l$$

$$p_l(t) = m_l(\lambda_l(t))$$

congestion measure

marking prob

Linear model - AQM

$$\dot{\lambda}_l = y_l(t) - c_l$$

$$p_l(t) = m_l(\lambda_l(t))$$

Aggregate rate

$$y_l(t) = \sum_i x_i(t - \tau_i^f)$$

Linearize around equilibrium

$$\dot{\lambda}_l = y_l(t)$$

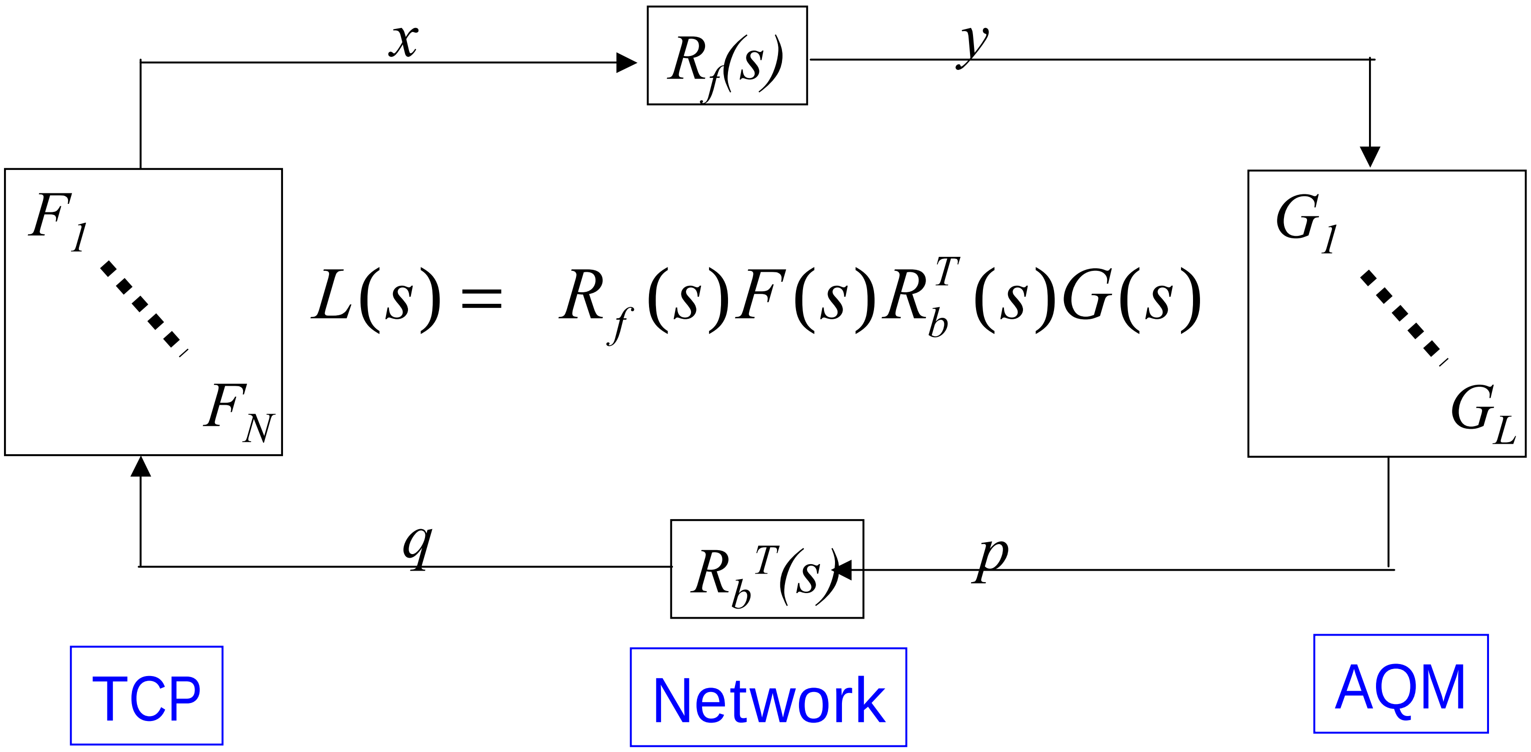
$$\dot{p}_l = m_l^{'}(\lambda_l)\, y_l(t)$$

In Laplace domain

$$p_l(s) = m_l^{'}(\lambda_l)\, y_l(s)$$

Linear Model - Network

Multi-link multi-source model



Closed loop system is stable if and only if

$$\det (I + L(s)) = 0$$

for no s in closed RHP

Single link (Reno/RED)

queue | RED averaging | TCP | RTT

$$L(s) = \frac{1}{2N^2} \frac{\tau}{\tau s + 1} \frac{\rho \alpha c}{s + \alpha c} \frac{c^3 \tau^2}{\frac{c\tau^2}{N} s + 2} e^{-\tau s}$$

N = #identical sources
τ = round trip time
ρ = slope of RED
α = RED queue weight
αc = link capacity

- TCP unstable as
  - Delay increases, or
  - Capacity increases !
- Stable only when window size is small

…… is scalable control possible?

Control objectives

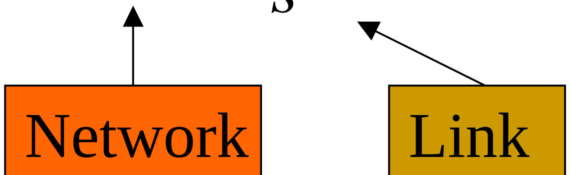
- Equilibrium
  - High utilization
  - Low loss and delay
- Dynamic
  - Stability scalable to any delay, capacity, load, topology
  - Delay invariant
  - This does not hold for current TCP

Restriction: decentralized control at links+sources

Delay compensation

- Equilibrium
  - High utilization: itegrator at links
  - Low loss & delay: REM, PI (HMTG01a)
- Stability
  - Integrator+network delay always unstable for high τ!

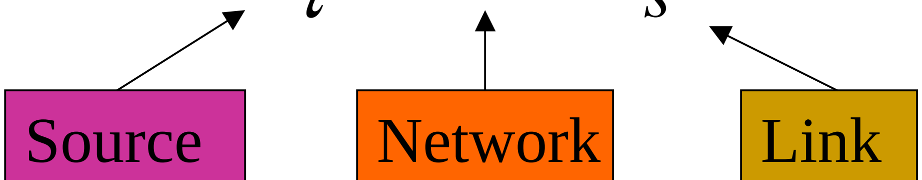
$$e^{-\tau s} \qquad \frac{1}{s}$$



Delay compensation

- Equilibrium
  - High utilization: itegrator at links
  - Low loss & delay: REM, PI (HMTG01a)
- Stability
  - Integrator+network delay always unstable for high τ!
- Delay invariance
  - Scale down gain by τ (known at sources)
  - This is self-clocking!

$$\frac{\alpha}{\tau} \quad e^{-\tau s} \quad \frac{1}{s}$$



Compensation for capacity

- Capacity invariance
  - Speed of adaptation combined with delay gives instability
- Simplest solution

$$\dot{p} = \frac{1}{c}(y - c)$$

Scalable control

Theorem (Paganini, Doyle, Low 2001)

Provided R is full rank, feedback loop is stable for arbitrary delay, capacity, and topology

Outline

- Duality model
- Dynamic model
- REM: Active Queue Management

Active queue management

- Idea: provide congestion information by probabilistically marking packets
- Issues
  - How to measure congestion (p and G)?
  - How to embed congestion measure?
  - How to feed back congestion info?

$$x(t+1) = F(\, p(t), x(t) \,)$$ ← Reno, Vegas

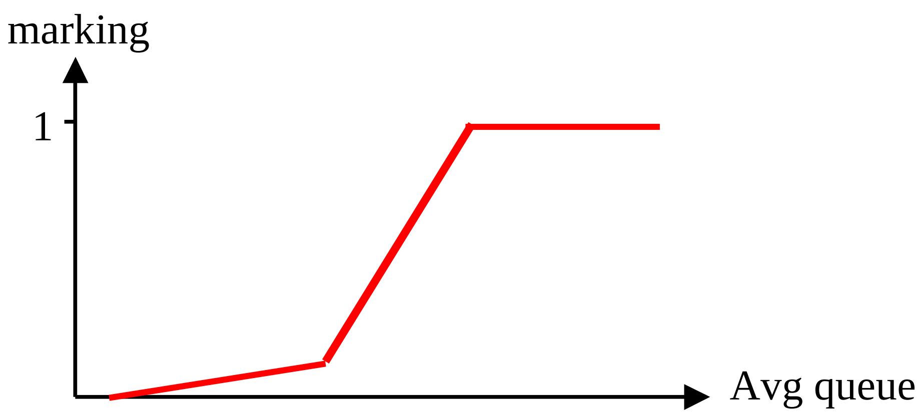
$$p(t+1) = G(\, p(t), x(t) \,)$$ ← DropTail, RED, REM

RED (Floyd & Jacobson 1993)

- Congestion measure: average queue length

$$p_l(t+1) = [p_l(t) + x^l(t) - c_l]^+$$

- Embedding: p-linear probability function



- Feedback: dropping or ECN marking

REM (Athuraliya & Low 2000)

- Congestion measure: price

$$p_l(t+1) = [p_l(t) + \gamma(\alpha_l b_l(t) + x^l(t) - c_l)]^+$$

- Embedding:

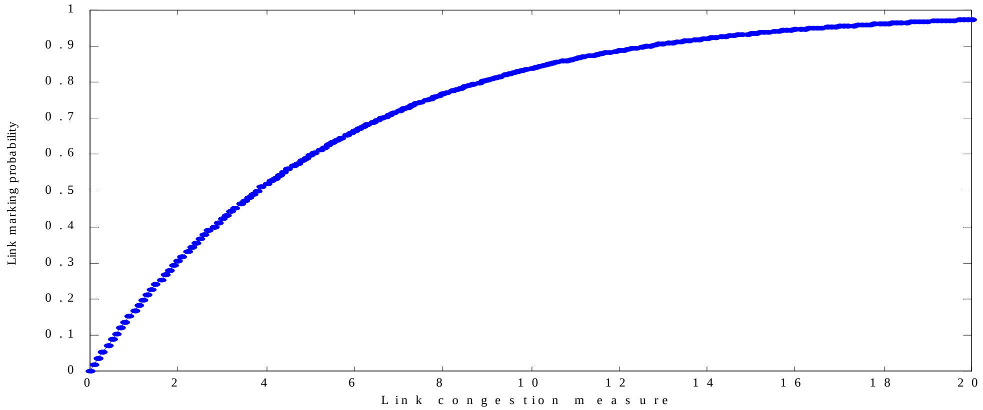
- Feedback: dropping or ECN marking

REM (Athuraliya & Low 2000)

- Congestion measure: price

$$p_l(t+1) = [p_l(t) + \gamma(\alpha_l\, b_l(t) + x^l(t) - c_l)]^+$$

- Embedding: exponential probability function



- Feedback: dropping or ECN marking

Key features

- Clear buffer and match rate

$$p_l(t+1) = [p_l(t) + \gamma(\alpha_l b_l(t) + \hat{x}^l(t) - c_l)]^+$$

Clear buffer   Match rate

- Sum prices

$$1 - \phi^{-p_l(t)} \Rightarrow 1 - \phi^{-p^s(t)}$$

Theorem (Paganini 2000)

Global asymptotic stability for general utility function (in the absence of delay)

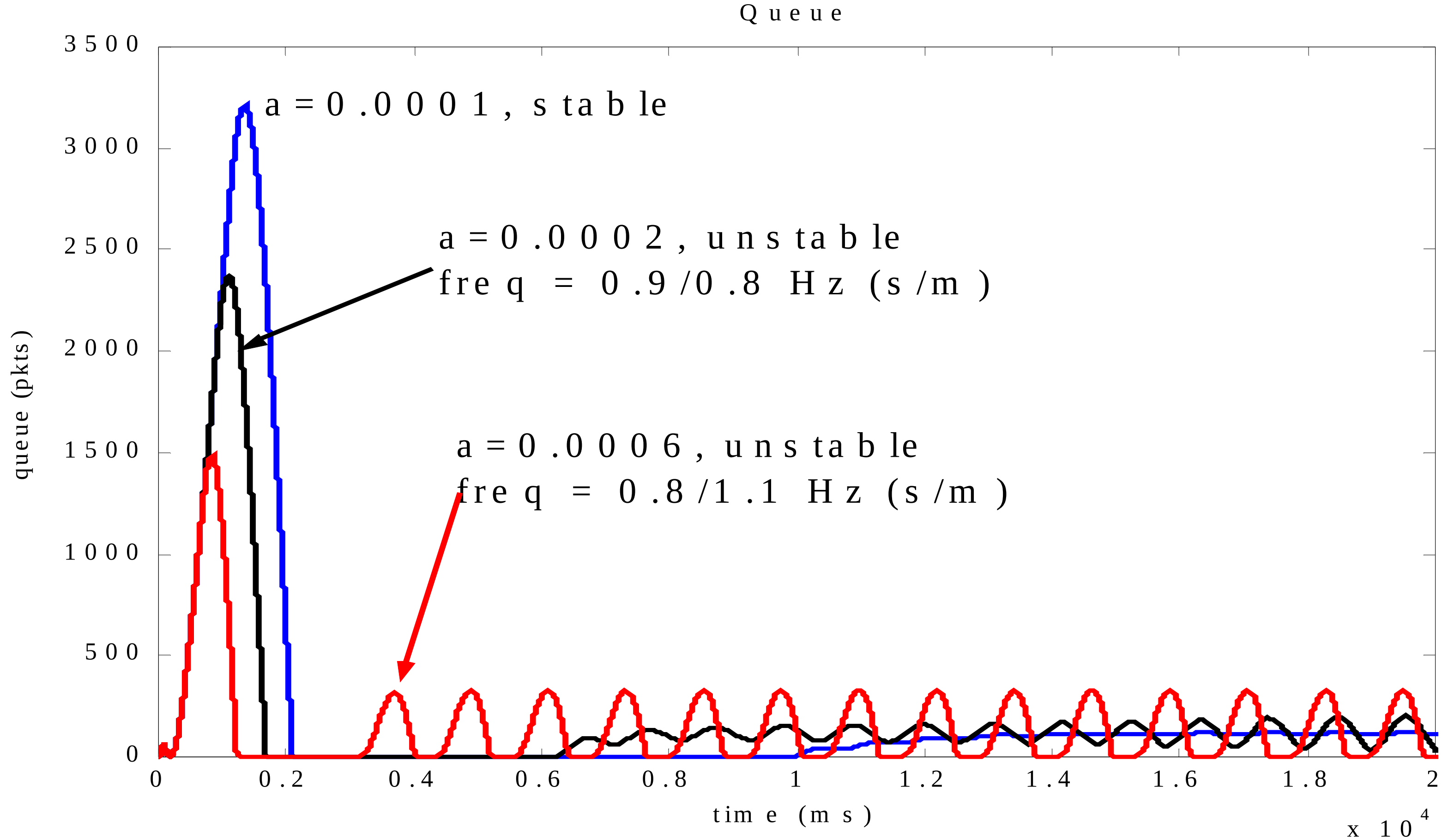
Congestion & performance

	$p_l(t)$	$G(p(t), x(t))$
Reno	loss	$[1 - c_l/x^l(t)]^+$ (?)
Reno/RED	queue	$[p_l(t) + x^l(t) - c_l]^+$
Reno/REM	price	$[p_l(t) + \gamma(\alpha_l b_l(t) + x^l(t) - c_l)]^+$
Vegas	delay	$[p_l(t) + x^l(t)/c_l - 1]^+$

- Decouple congestion & performance measure
  - RED: \`congestion' = \`bad performance'
  - REM: \`congestion' = \`demand exceeds supply'

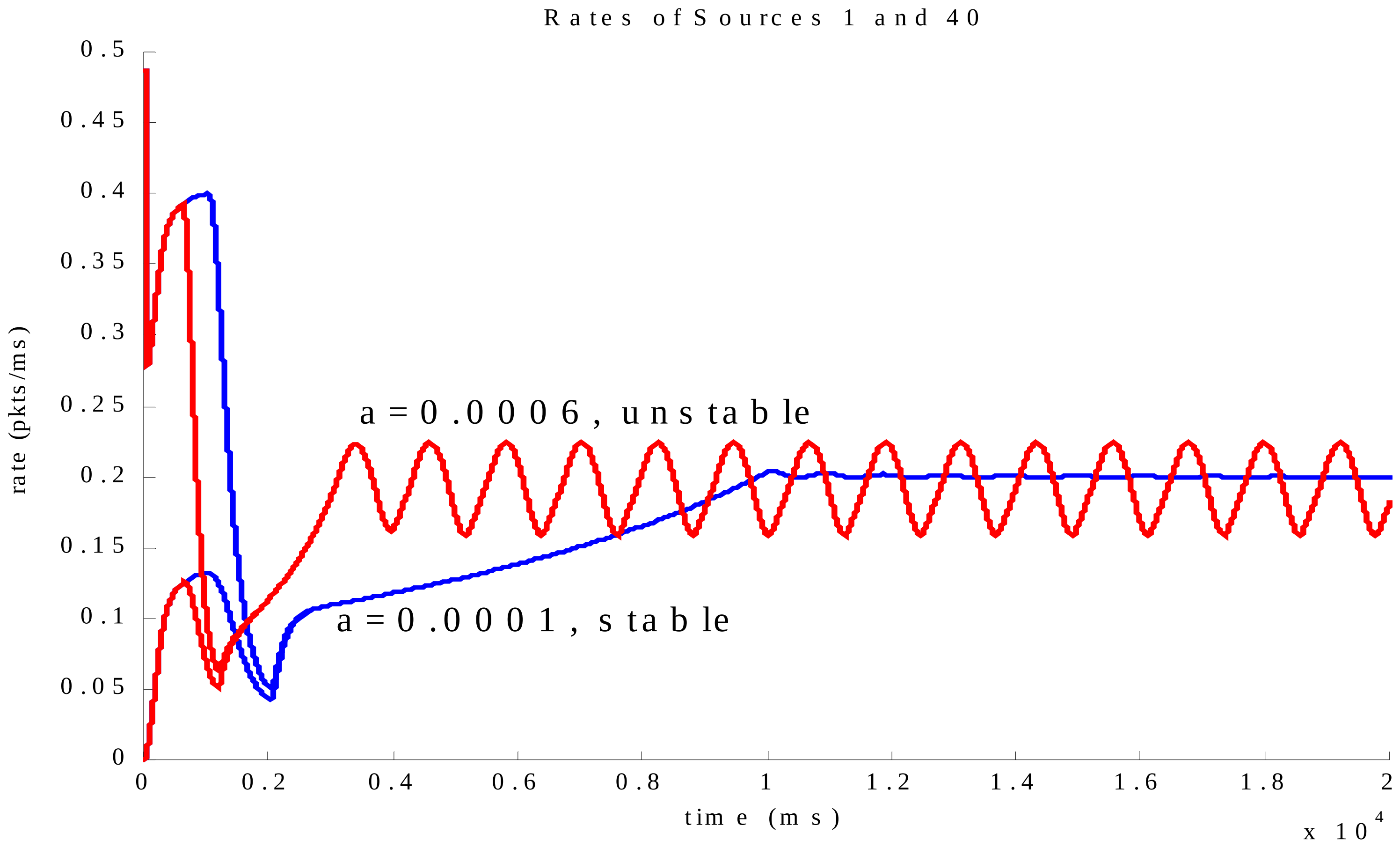
    But performance remains good!

Nonlinear Fluid Simulation



- Oscillation of Reno/RED-gentle is not just due to AIMD

Nonlinear Fluid Simulation



- Corresponding source rates
- Average (i.e. fluid) rates synchronized