Scalable Performance Signalling and Congestion Avoidance

PhD presentation

Supervisor: Max Mühlhäuser 2nd supervisor: Jon Crowcroft

Abteilung Telekooperation, TU Darmstadt, Nov. 18th 2002

Michael Welzl

michael.welzl@uibk.ac.at

University of Linz -> University of Innsbruck

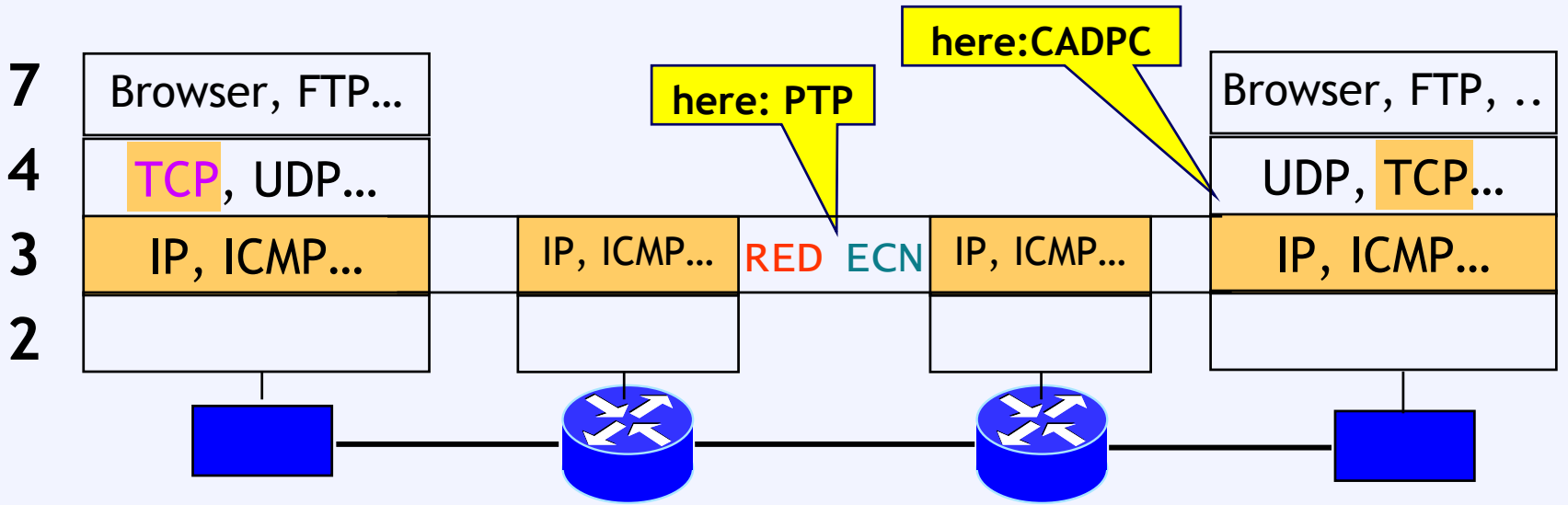
Outline

- Problem identification / motivation
- Proposed solution
- Simulation results
- Conclusion

Internet Congestion Control (CC)

- Necessary to keep the Internet stable
  - prevent congestion collapse
- must be scalable → (bad? old) idea:
  - ***no*** extra work for routers: congestion detected via **packet loss** in TCP

OSI:



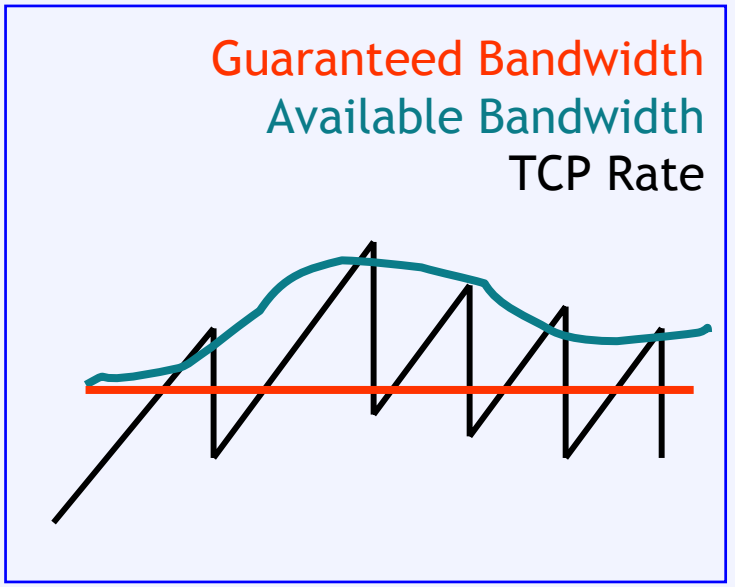
- Later: ***some*** extra work for routers
  - active queue management: RED, actual communication: ECN

Some problems with TCP(-like) CC

- Special links (becoming common!):
  - noisy (wireless) links
  - "long fat pipes" (large bandwidth*delay product)
- Stability issues
  - Fluctuations lead to regular packet drops + reduced throughput
    => problematic for streaming media
  - Stability depends on delay, capacity, load, AQM
- Rate hard to control / trace / predict
  - Load based charging difficult
- **Main reason:** binary congestion notification (E)CN
  - when it occurs, it's (almost) too late

Quality-of-Service ↔ CC

- Separate research areas up to now!!
- Common goals
  - efficiently use available bandwidth
  - provide "good" QoS
- Theory
  - AIMD, self-clocking, ..
- Practice
  - Problem with Multimedia



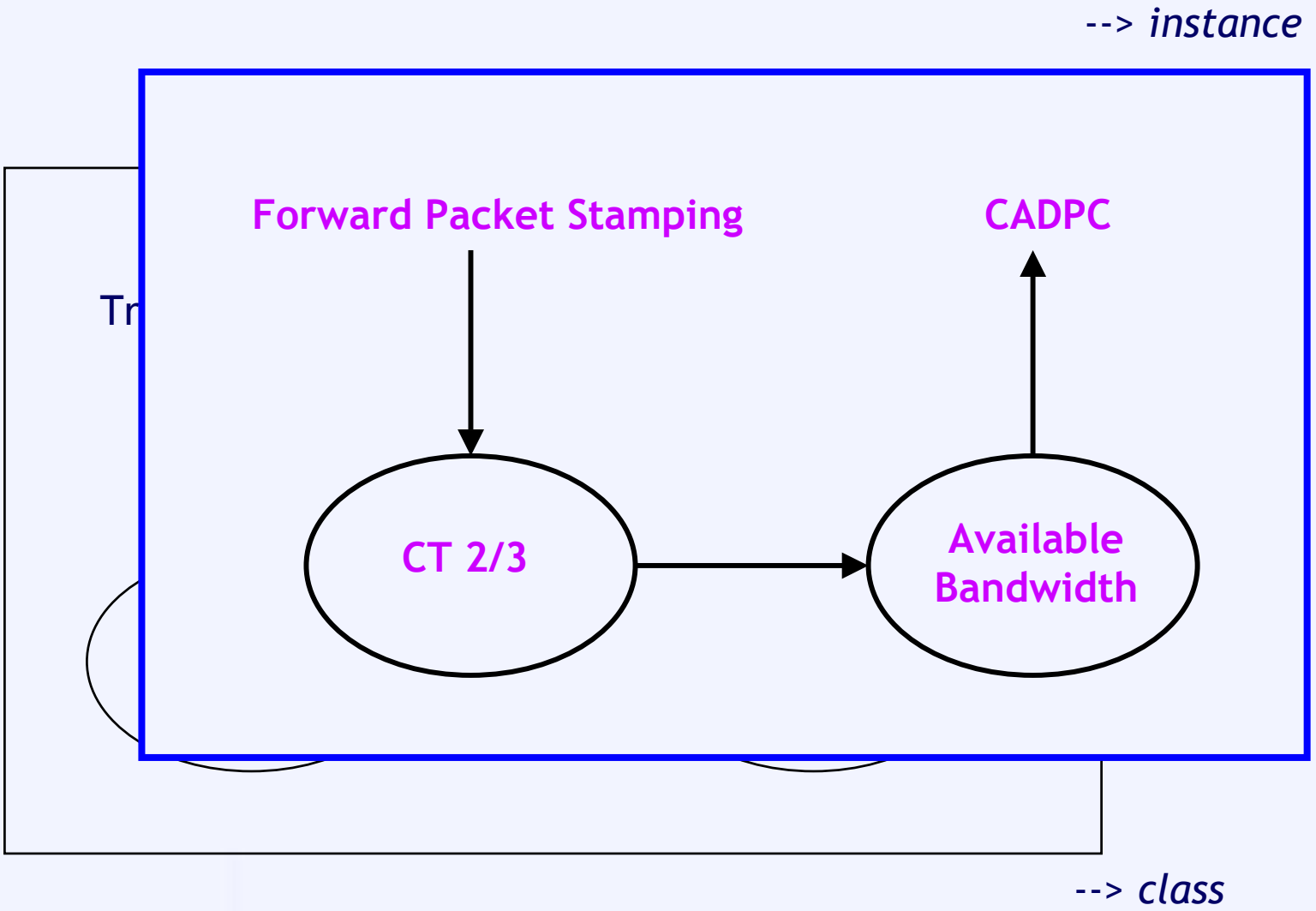
Proposed Solution

- Totally different CC model
  - *only* rely on rare explicit bandwidth (=traffic) signaling
- Assumptions:
  - extra forward signalling for CC = good idea *(≠ common belief!!)*
  - router support
  - mechanism must clearly outperform TCP to justify (even a little!) additional traffic
  - timeouts necessary for loss of signalling packets *rate should not depend on round trip time RTT*

Outline

- Problem identification / motivation
- Proposed solution
  - PTP framework
  - CADPC
- Simulation results
- Conclusion

PTP: the framework

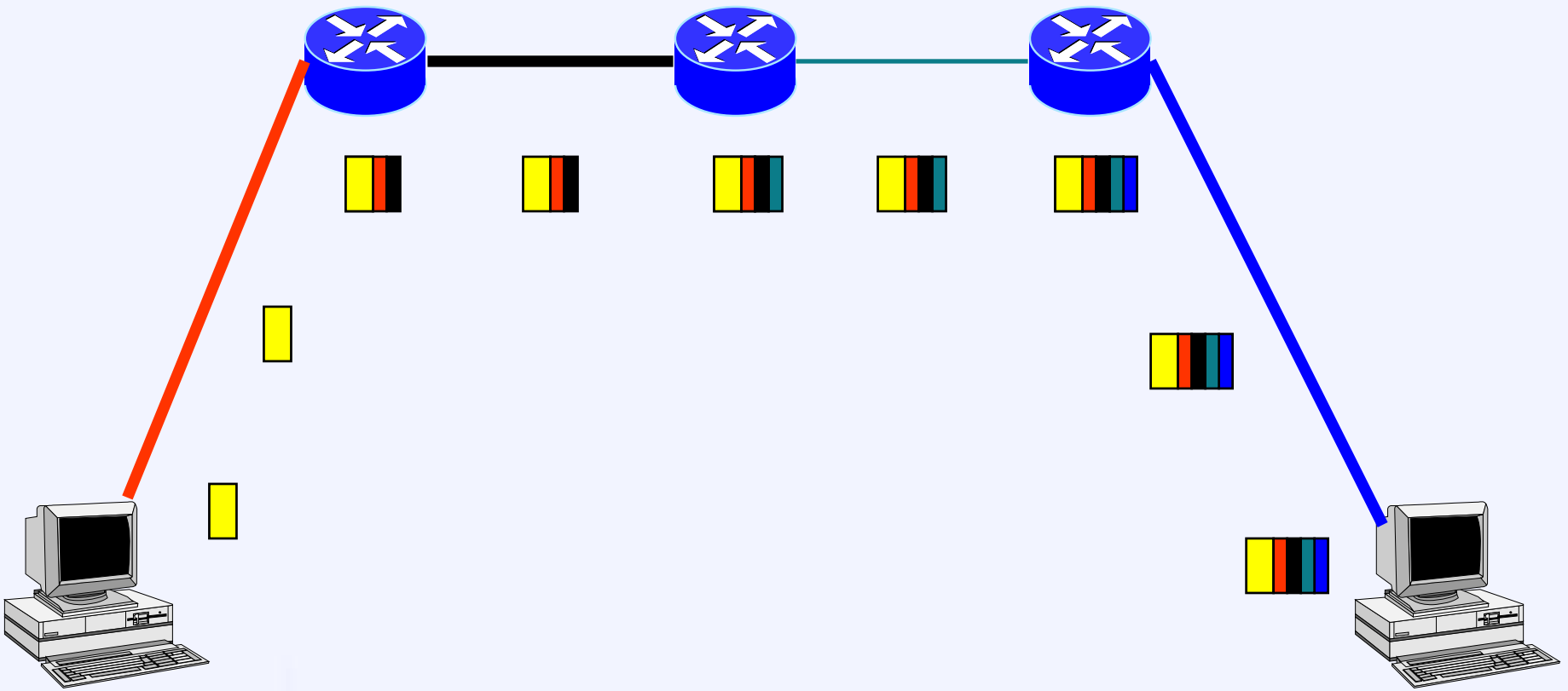


PTP: the signalling protocol

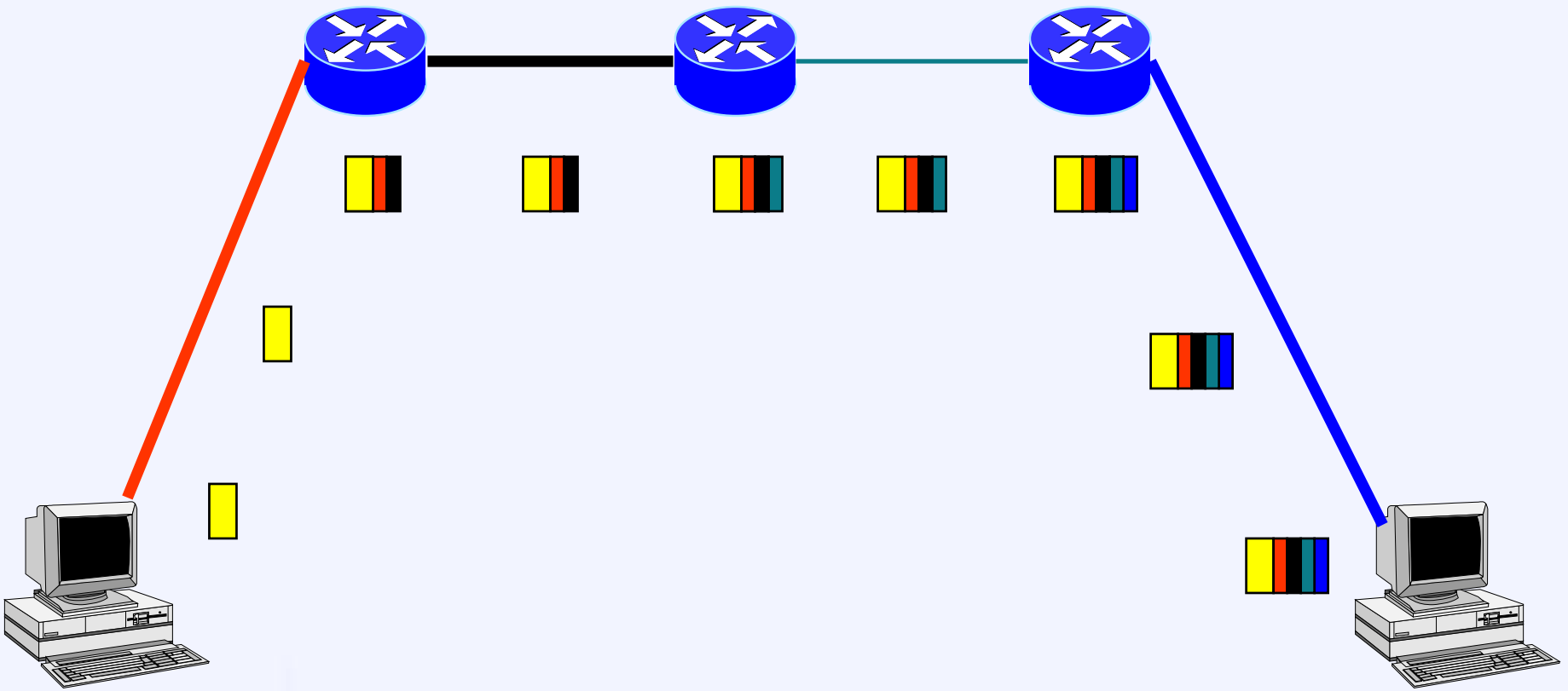
- quasi "generic ECN" - to carry performance information (standardised Content Types, e.g. queue length, ..)
  - resembles ATM ABR and XCP
- **Stateless + simple -> scalable!**
  - all calculations @ end nodes
- Only every 2nd router needed for *full* functionality
- e.g. Available Bandwidth Determination:
  - nominal bandwidth ("ifSpeed") + 2* (address + traffic counter ("if(In/Out)Octets") + timestamp) = available bandwidth
- **two modes:**
  - Forward Packet Stamping
  - Direct Reply (not for available bandwidth (byte counters))

No problems w/ wireless links unless combined with packet loss!

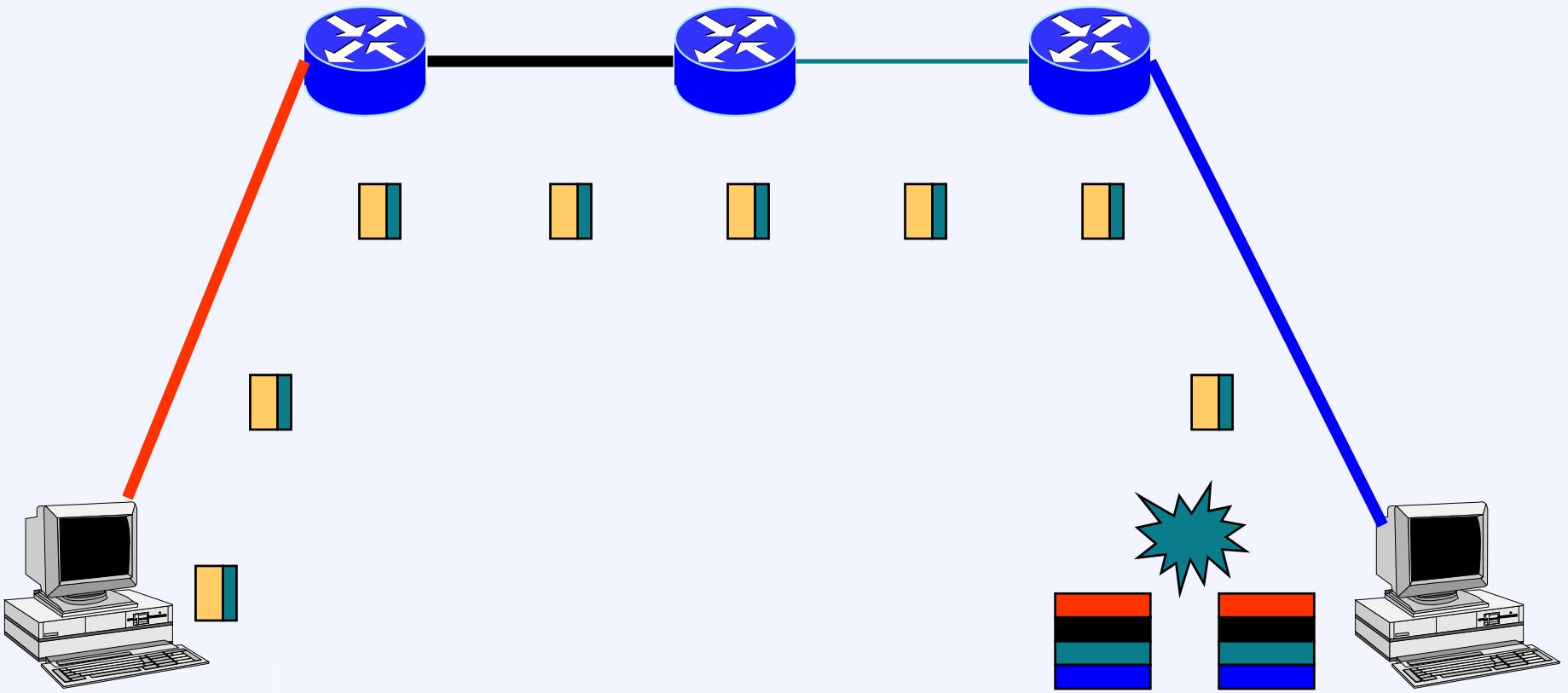
Forward Packet Stamping: how it works



Forward Packet Stamping: how it works



Forward Packet Stamping: how it works



Outline

- Problem identification / motivation
- Proposed solution
  - PTP framework
  - CADPC
- Simulation results
- Conclusion

CADPC: a new CC *mechanism*

- "Congestion Avoidance with Distributed Proportional Control"
  fully distributed convergence to max-min fairness

- Idea:
  - relate user's current rate to the state of the system (also in LDA+)
    In the Chiu-Jain-diagram, if the rate increase factor is indirectly proportional to the user's current rate, the rates will equalise.

- From:
  - erx = 1 + d* rup = 1 + rup * (1 - traffic/r0)

relate
traffic : target rate

- To:
  - erx = 1 + d * (1 - myRate/d)

relate
user's rate : available bandwidth

- Final formula (normalised with Bottleneck capacity):
  x(t+1) = x(t)a(1-x(t)-traffic)+x(t)

CADPC, contd.

- Only depends on old rate, smoothness factor and traffic
  - does not depend on RTT
    - Feedback packets can be delayed => scalable
    - reasonable choice: 4 x RTT
- Control realises logistic growth
  - Asymptotically stable in simplified fluid model with synchronous RTTs
  - Smooth convergence to a steady rate
- Initial convergence can be slow (depending on bg traffic)
  - startup enhancement: temporarily sacrifice stability

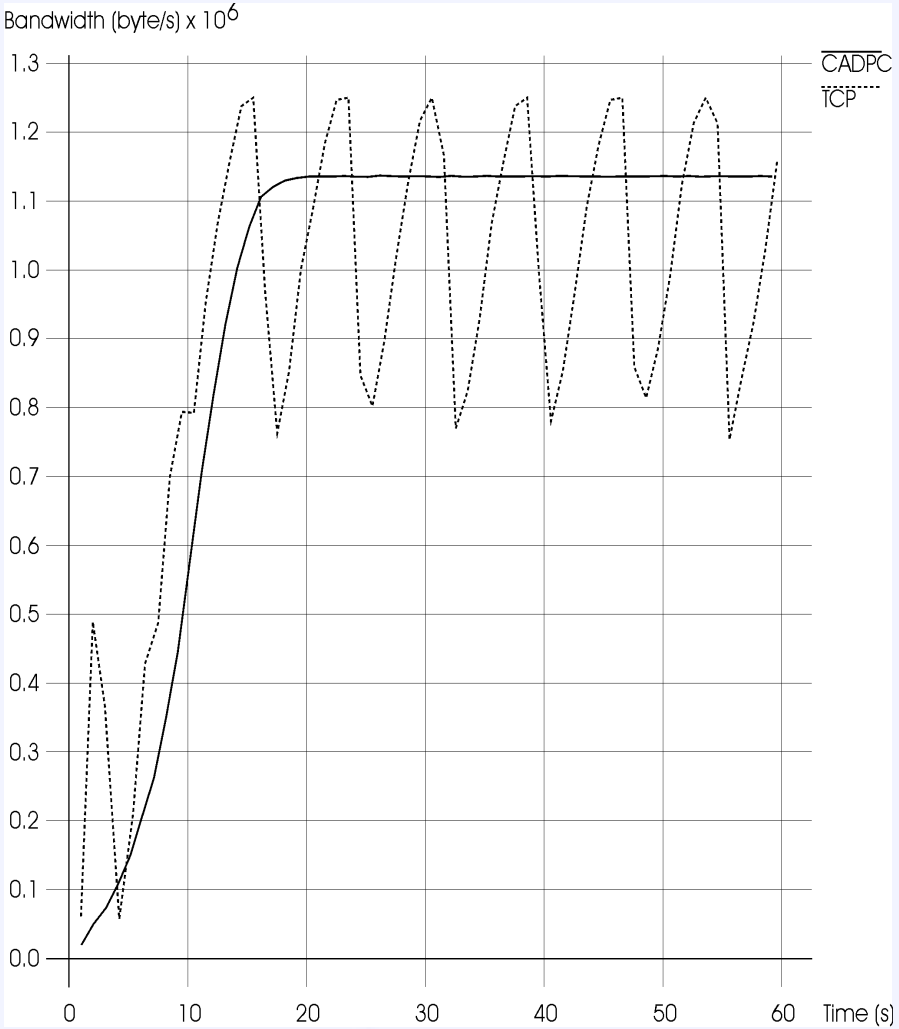
Outline

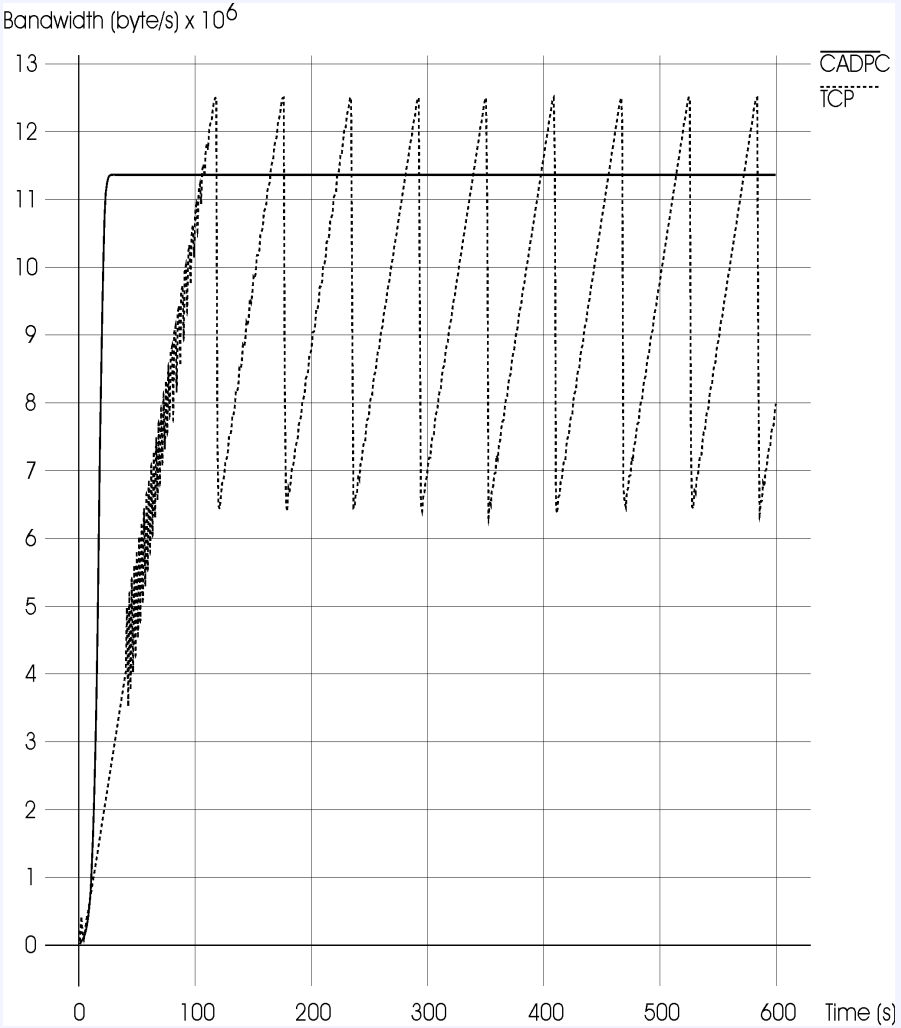
- Problem identification / motivation
- Proposed solution
- Simulation results
  - Dynamic behaviour
  - Long-term performance evaluation
- Conclusion

Unless otherwise mentioned...

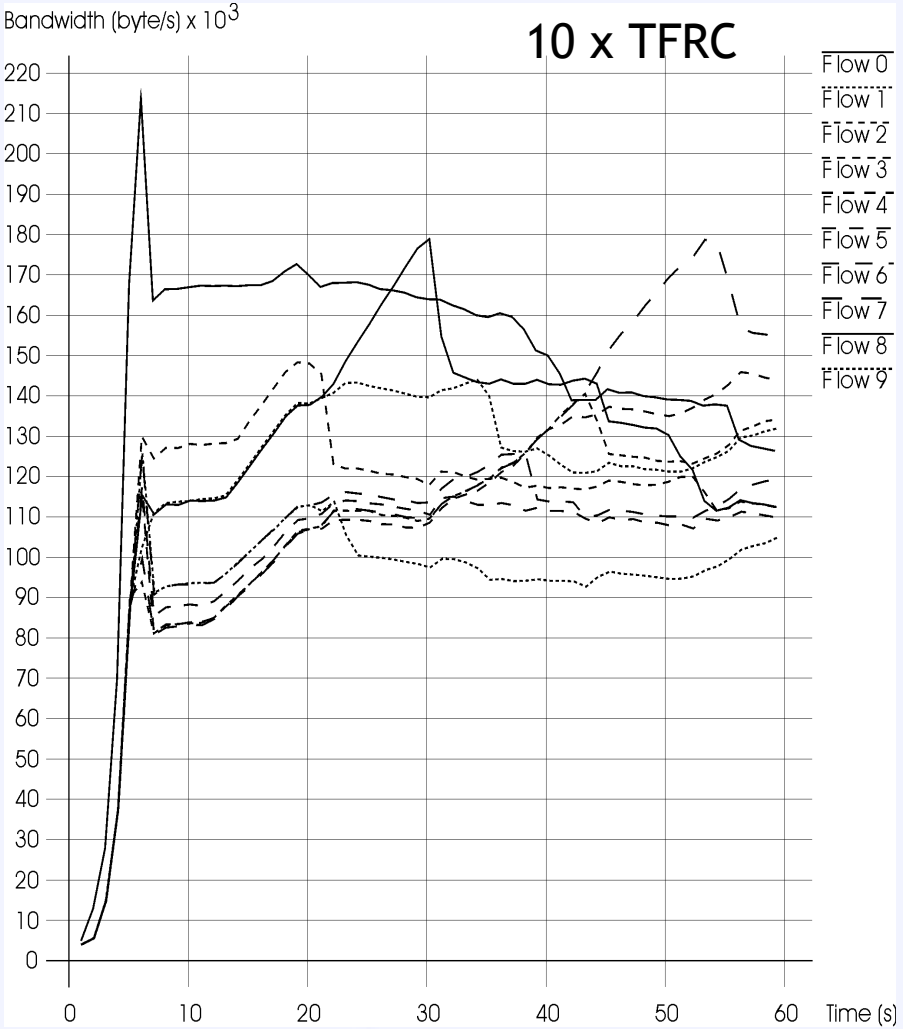
Topology	**Dumbbell**
CADPC update frequency	**4 RTTs**
CADPC smoothness factor a	**0.5**
Packet Sizes	1000 bytes
Bottleneck Link Bandwidth	10 Mbit/s
Bandwidth of all other links	1000 Mbit/s
Link delay	50 ms each
Queueing discipline	Drop-tail
Duration	Long-term: 160 seconds
All flows start after	0 seconds
Flow type	Greedy, long-lived
TCP flavour	TCP Reno
Bottleneck Queue Length	50 packets

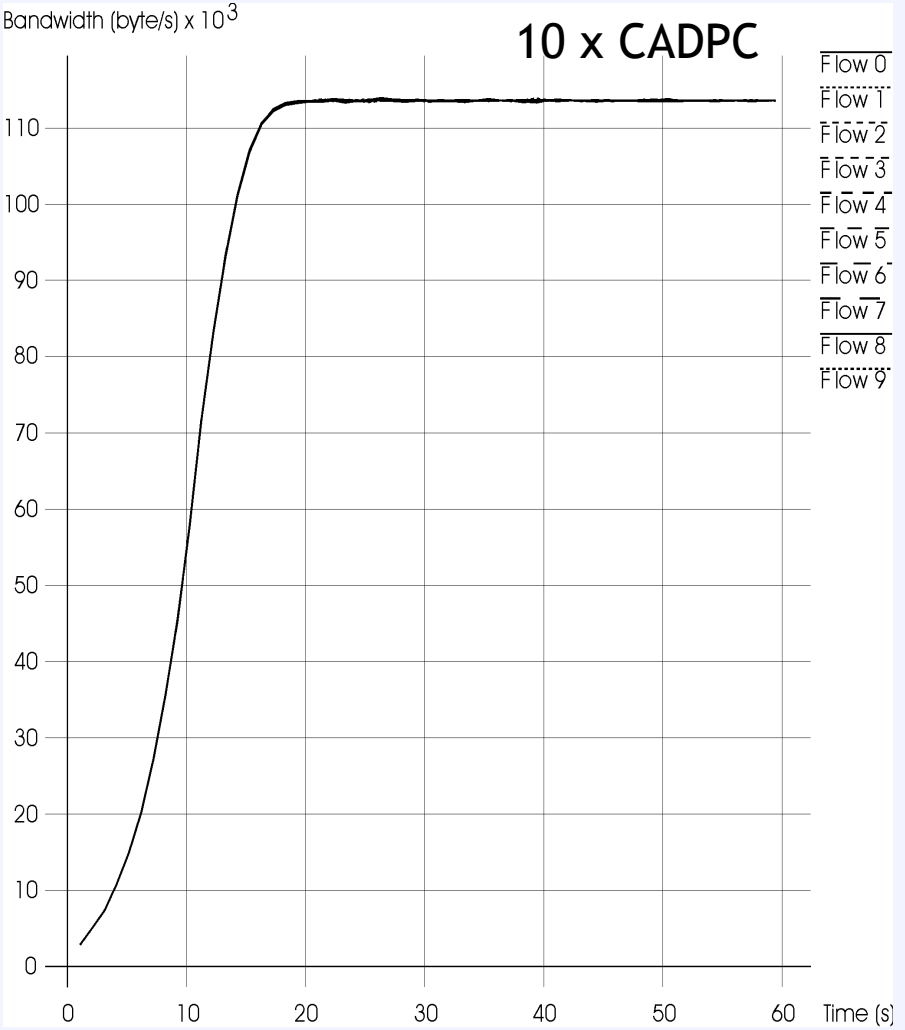
CADPC vs. TCP


Bandwidth (byte/s) x 10^6
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50
60
Time (s)
CADPC
TCP

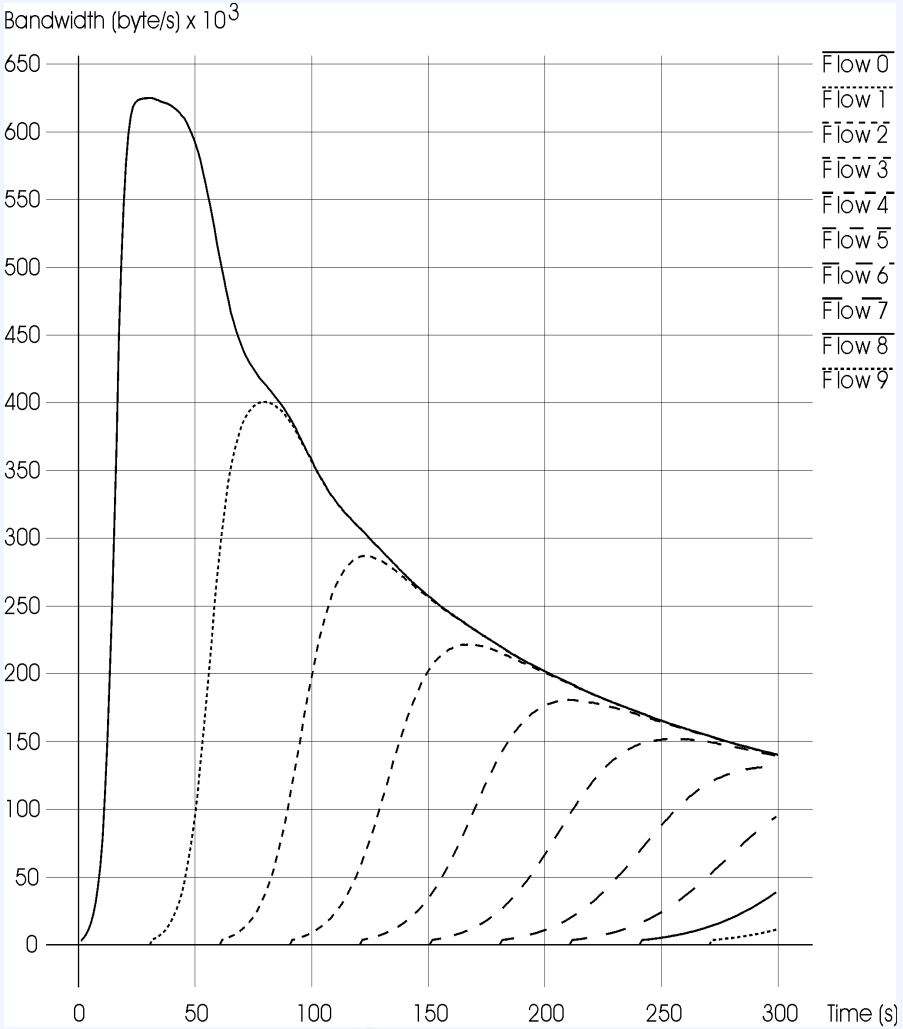

Bandwidth (byte/s) x 10^6
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0
100
200
300
400
500
600
Time (s)
CADPC
TCP

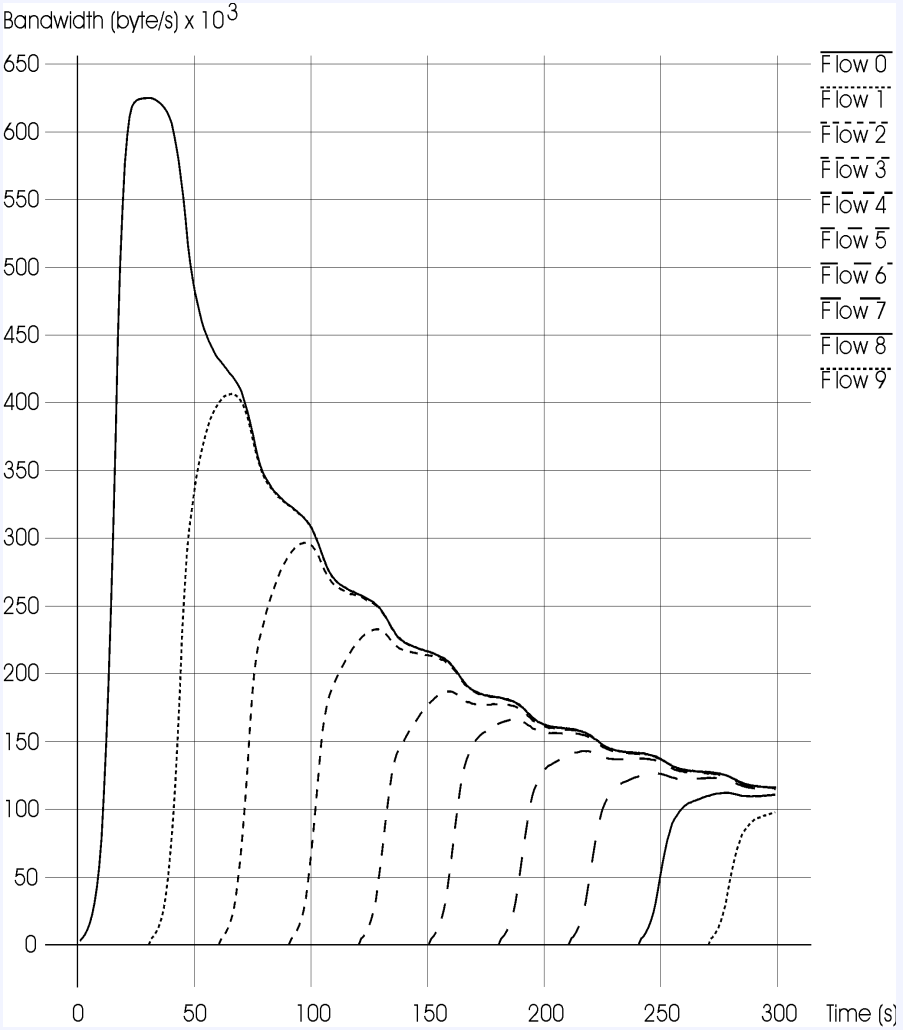
Smoothness


Bandwidth (byte/s) x 10^3
10 x TFRC
Flow 0
Flow 1
Flow 2
Flow 3
Flow 4
Flow 5
Flow 6
Flow 7
Flow 8
Flow 9
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
Time (s)

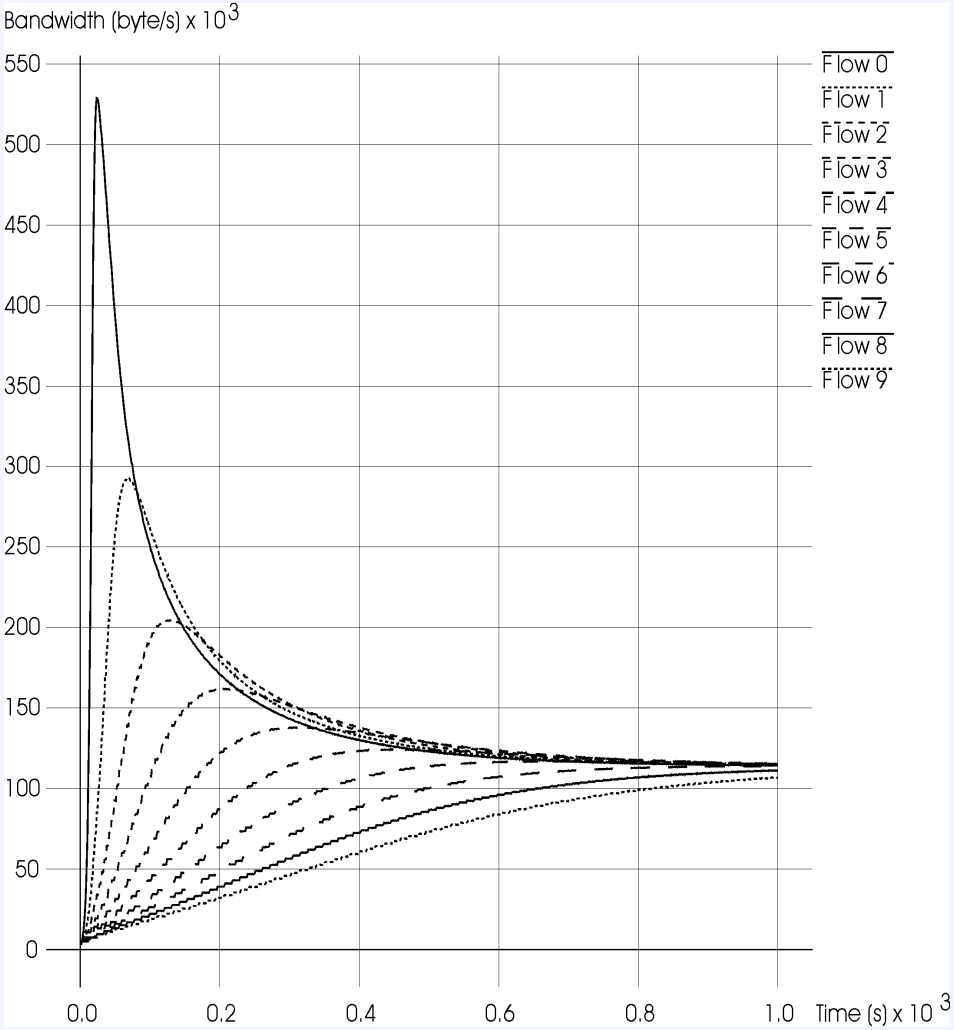

Bandwidth (byte/s) x 10^3
10 x CADPC
Flow 0
Flow 1
Flow 2
Flow 3
Flow 4
Flow 5
Flow 6
Flow 7
Flow 8
Flow 9
110
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
Time (s)

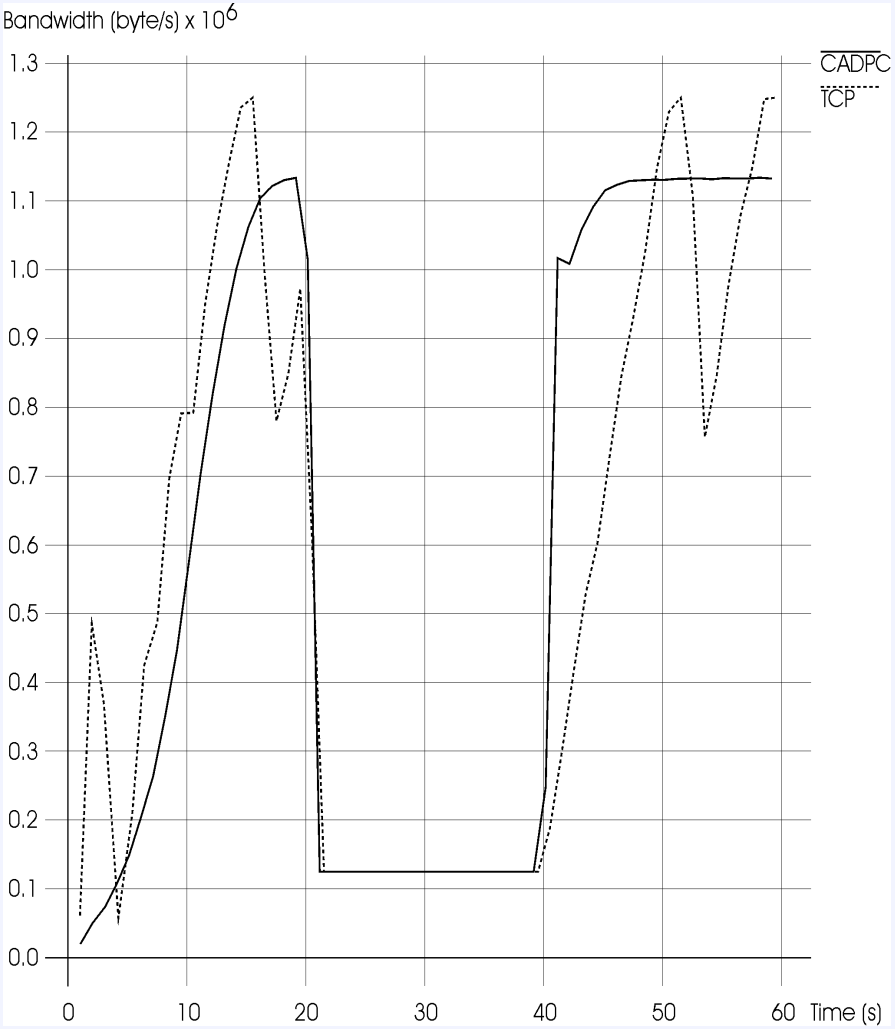
Startup enhancement



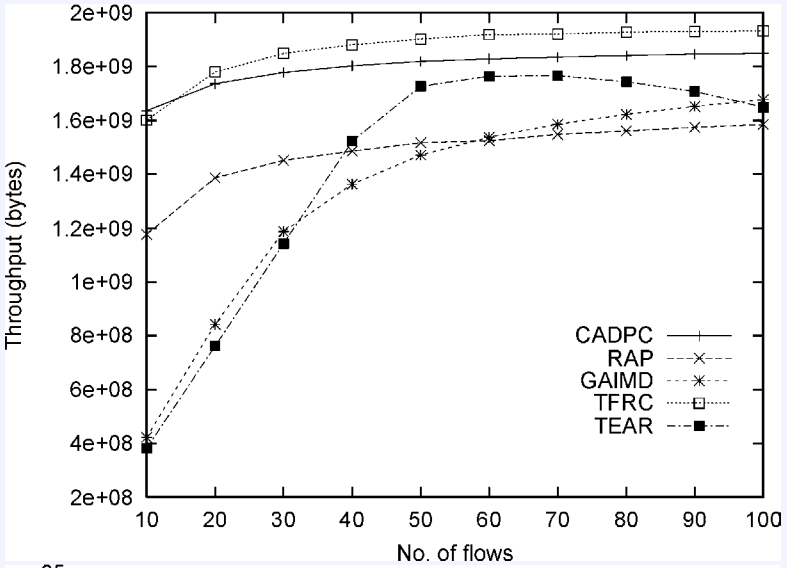


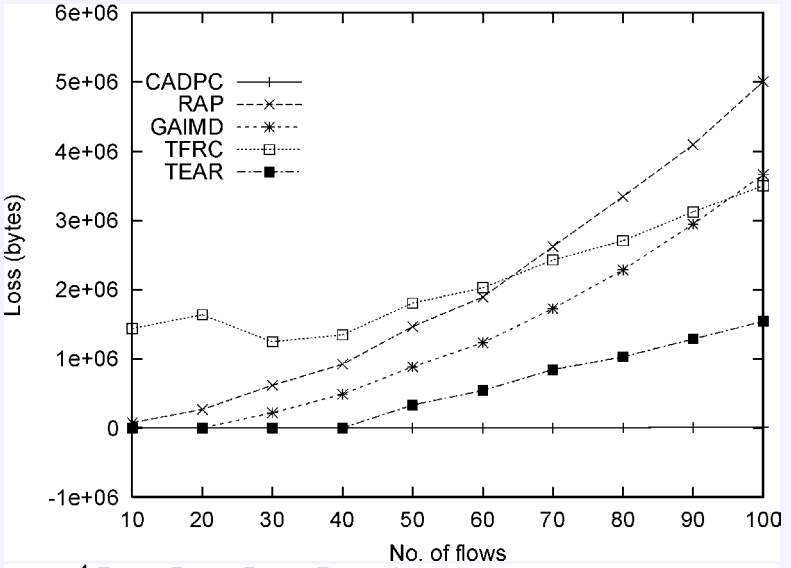
Heterogeneous RTTs + Robustness

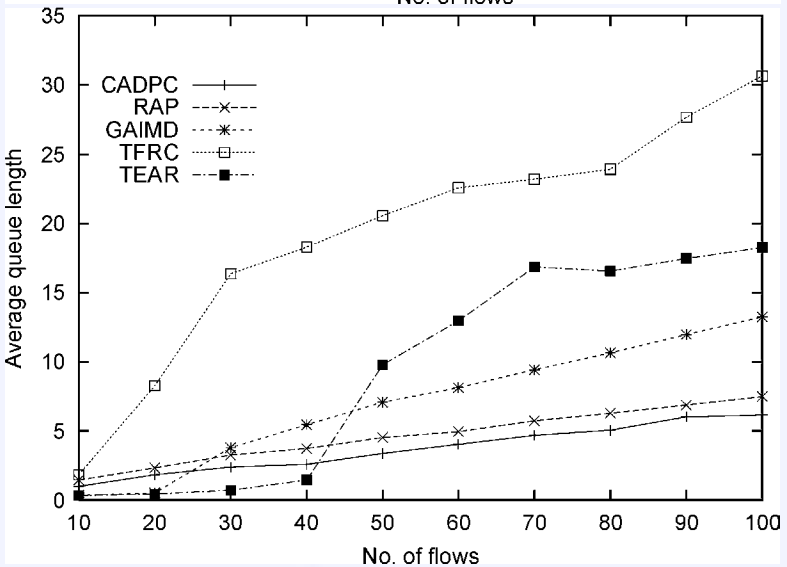


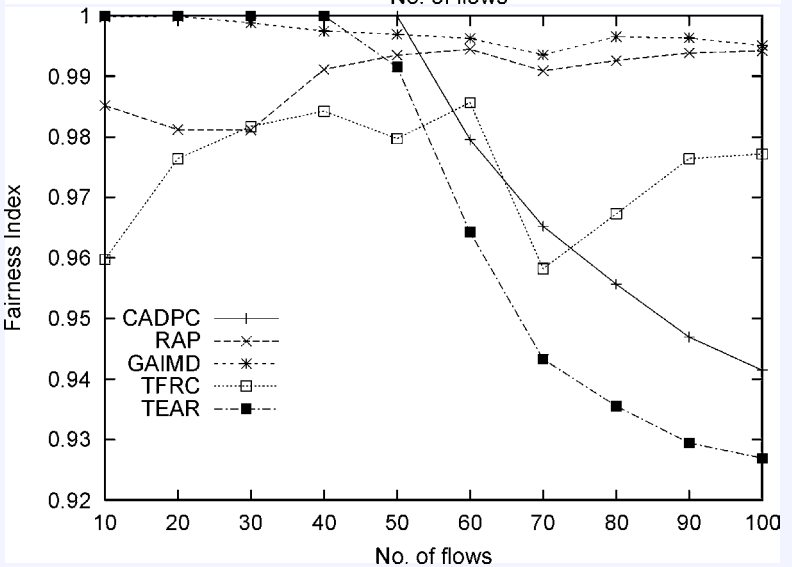


CADPC vs. TCP-friendly CC. mechanisms

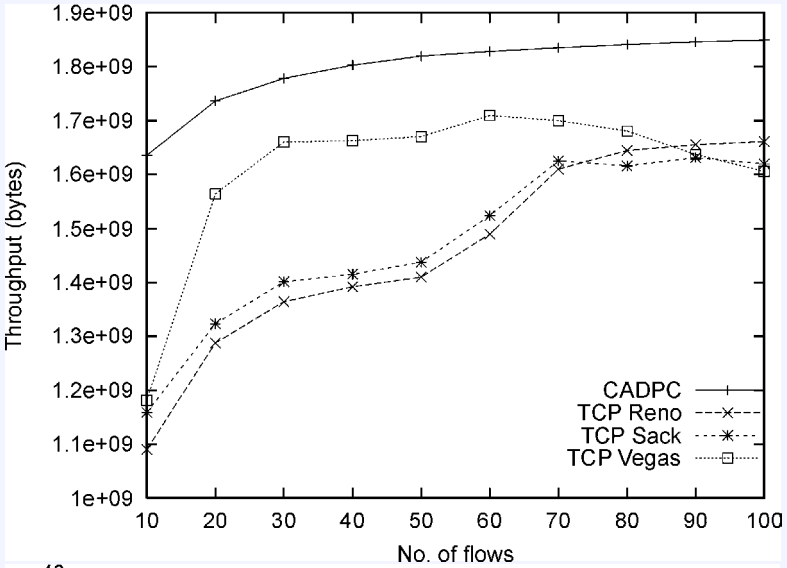


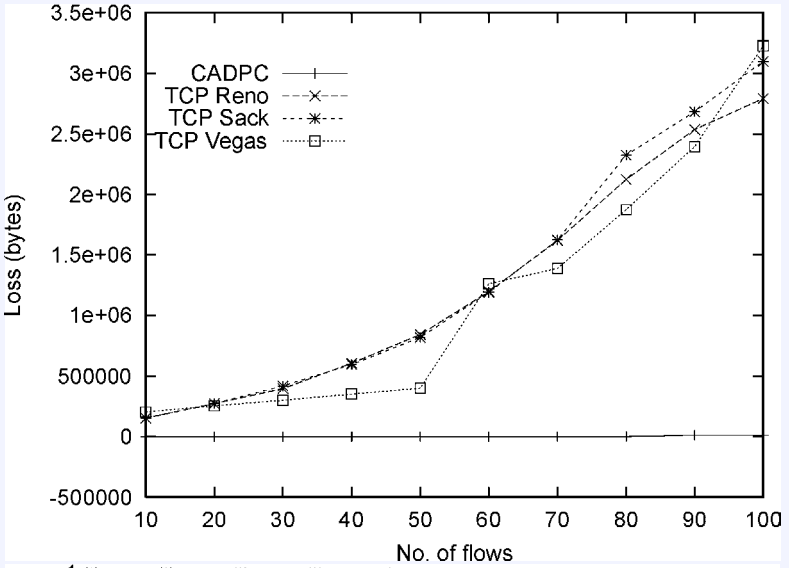


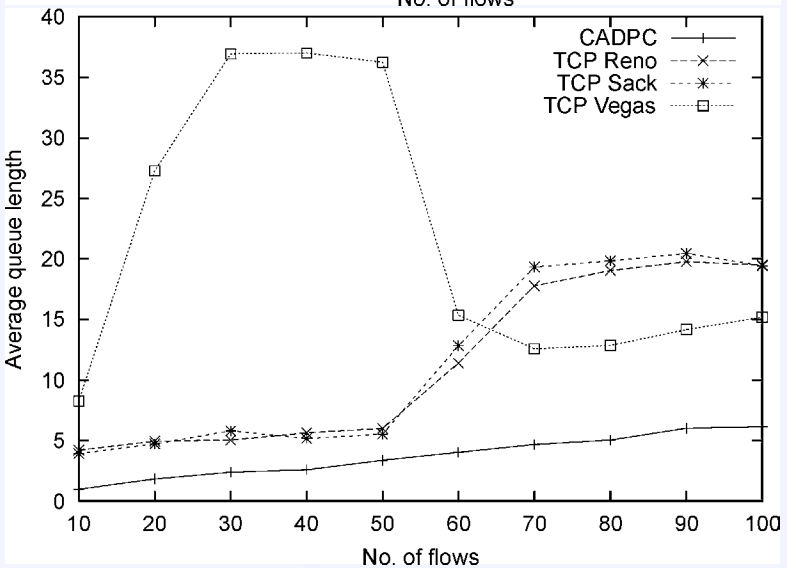


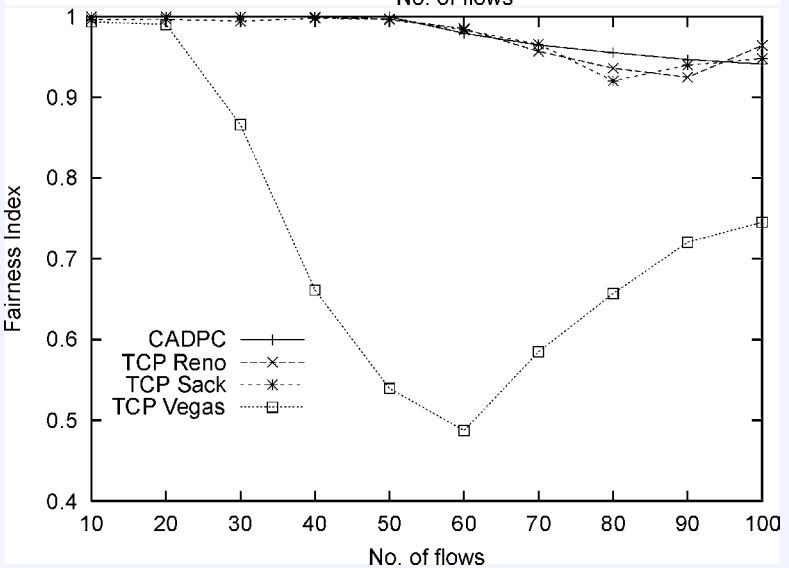


CADPC vs. 3 TCP(+ECN) flavors









Further simulations (in the thesis)

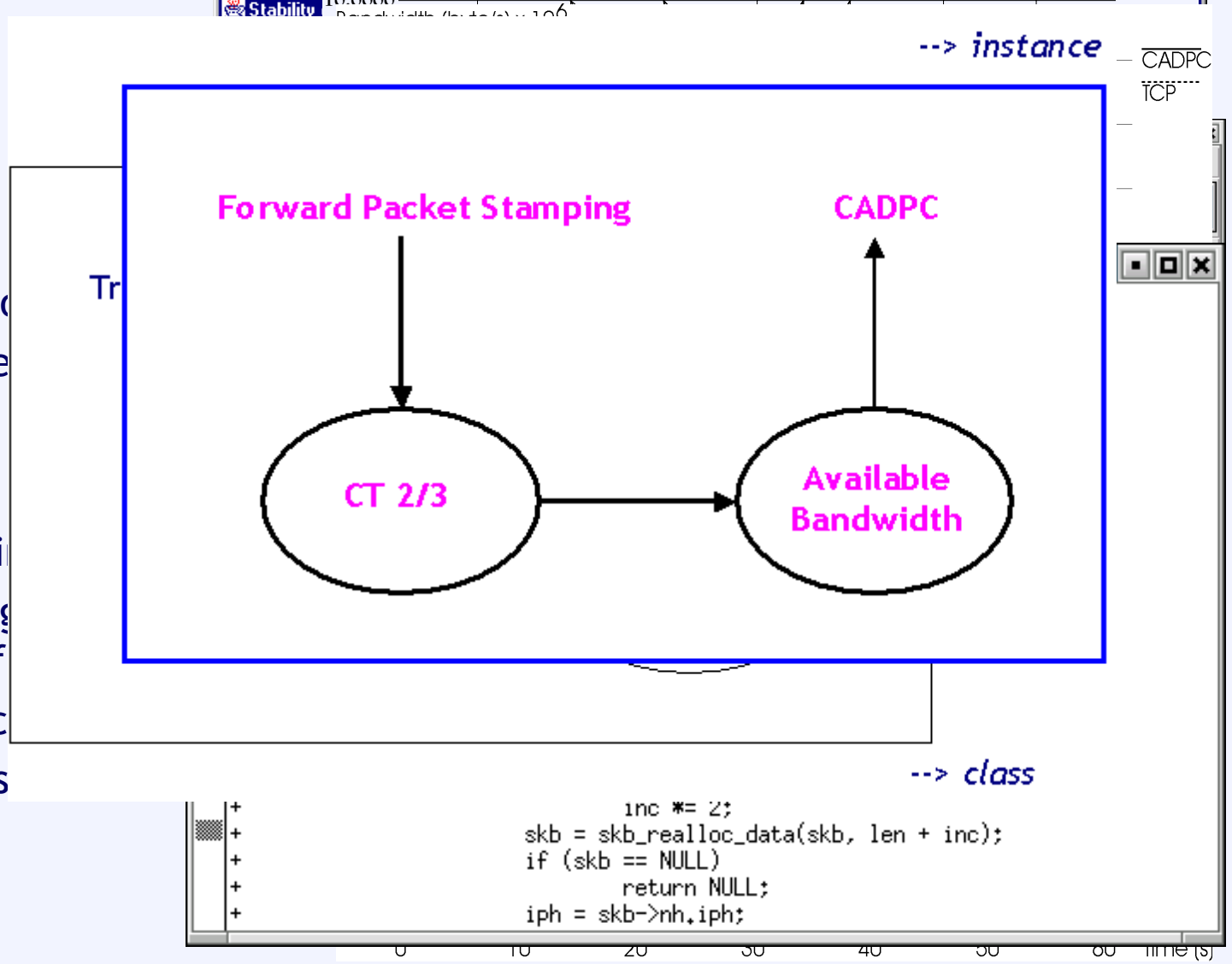
- Dynamic:
  - Dependence on smoothness parameter *a* and packet size
  - Robustness against fast routing changes
  - Effect of mixing converged flows
  - Performance across highly asymmetric connections + noisy links
- Long-term (throughput, loss, average queue length, fairness):
  - Check valid parameter space: bandwidth, no. of flows, packet size
  - Varying bottleneck bandwidth
  - Varying feedback delay
  - RED, REM and AVQ Active Queue Management
  - Behaviour with varying amount of web background traffic
  - Max-min fairness (scenario with 2 bottlenecks)

Outline

- Problem identification / motivation
- Proposed solution
- Simulation results
- Conclusion

Tangible Outcomes

- PTP
  - **Framework**
  - **Protocol**
    - *ns* simulat
    - Linux code
- CADPC
  - fluid-model si
    - vector diag analysis of
  - *ns* simulator c
  - simulation res



CADPC Advantages

- high utilisation
- close to zero loss
- small bottleneck queue length
- very smooth rate
- fully distributed precise max-min-fairness

some say it's impossible :)

- rapid response to bandwidth changes (e.g. from routing)
- provable asymptotic stability (synchronous RTTs, fluid model)

CADPC Advantages /2

- Useful for asymmetric links
- Useful for noisy (wireless) links + "long fat pipes"
- Useful for QoS and load-based charging

Disadvantages

- Requires router support

but: sticks to KISS principle

- Requires traffic isolation because…
  - not tcp-friendly
  - slowly responsive: bad results with web traffic

but: see future work...

IMHO: considerable step towards better congestion control

...based on drastic departure from existing approaches

Future work

- So far, ...
  - only max-min-fairness supported
  - only greedy sources simulated

- Gradual deployment ideas:

  - CADPC / PTP within a DiffServ class (QoS "in the small"):
    "we offer QoS + provide router support,
    you use CADPC and obtain a good result
    [and we can calculate your rate, too]"

  - If CADPC works with non-greedy senders:
    edge2edge PTP signalling
    - PTP supported traffic engineering (TCP over CADPC)
    - CADPC <=> TCP translation at edge routers?

The End ...

- Related publications
- PTP ns code
- PTP Linux code (router kernel patch + end system implementation)
- Future updates: thesis, CADPC code, ..

http://fullspeed.to/ptp

Additional information

The end2end argument

This thesis and the end2end argument

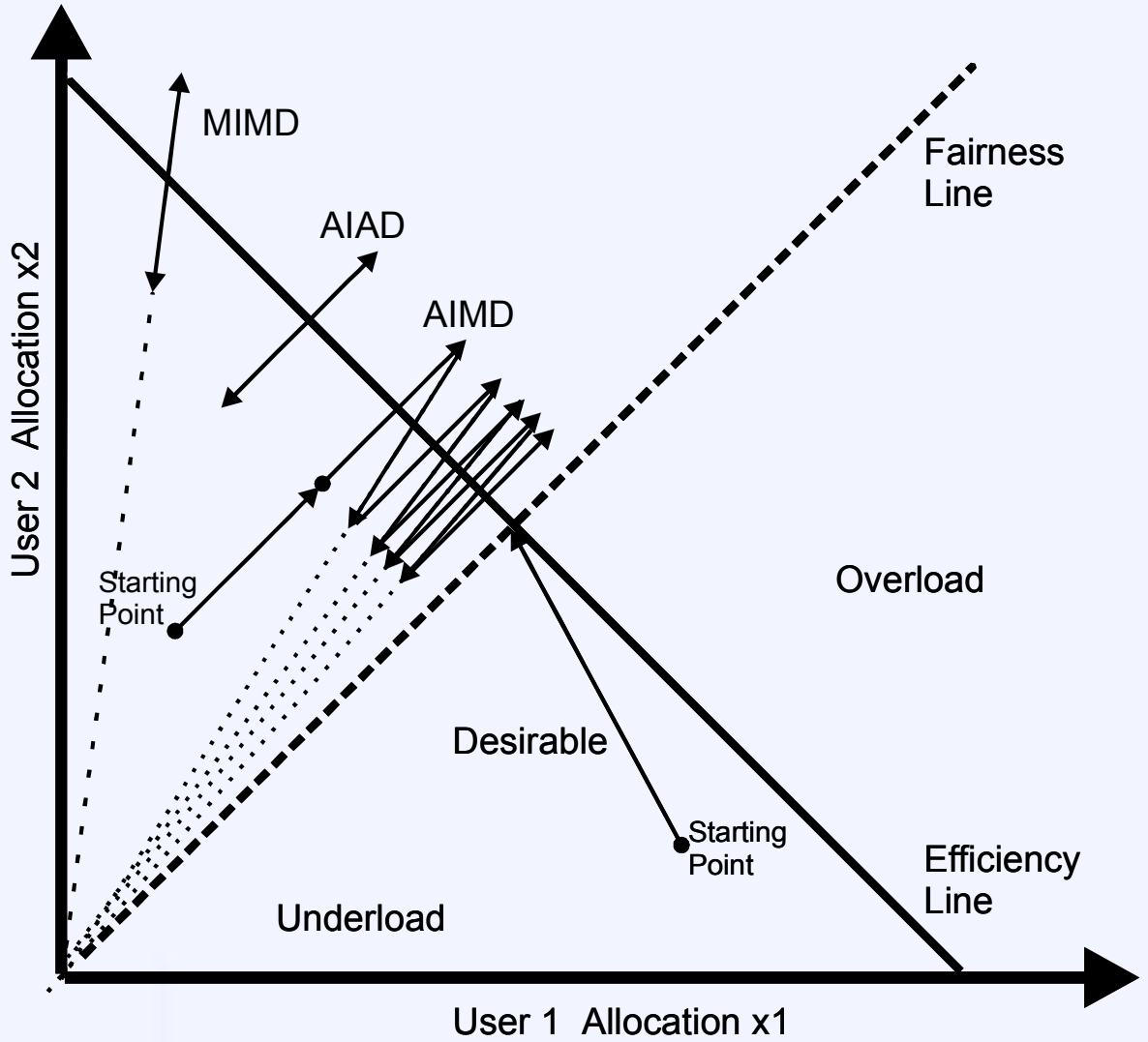
- „Lower-level layers, which support many independent applications, should provide only resources of broad utility across applications, while providing to applications usable means for effective sharing of resources and resolution of resource conflicts. (network transparency)"

  [Active Networking and End-To-End Arguments, Comment by David P. Reed, Jerome H. Saltzer, and David D. Clark]

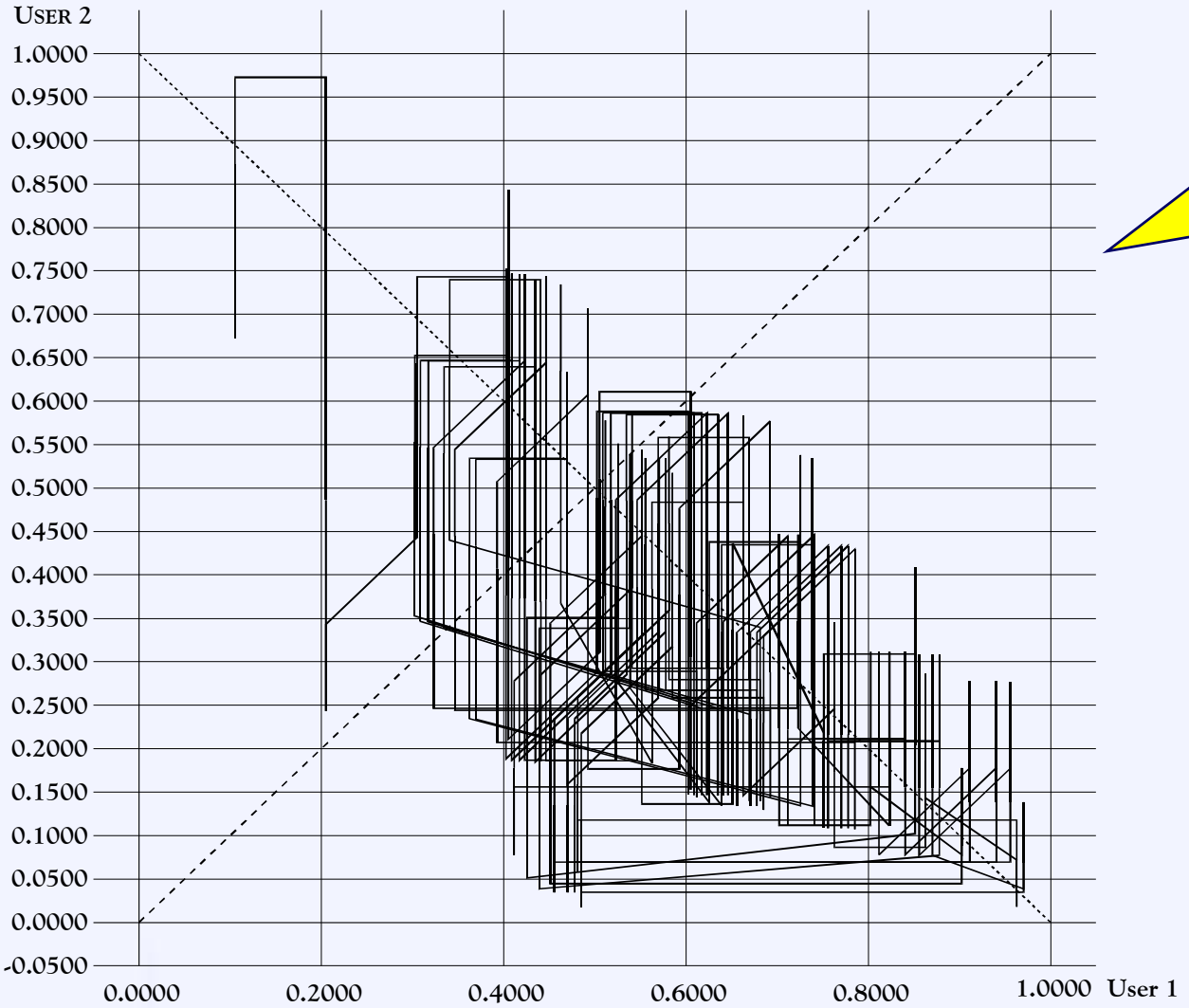
- Goals of this thesis:
  - provide such means
  - use it!

AIMD Background

AIMD in Theory (equal RTTs)

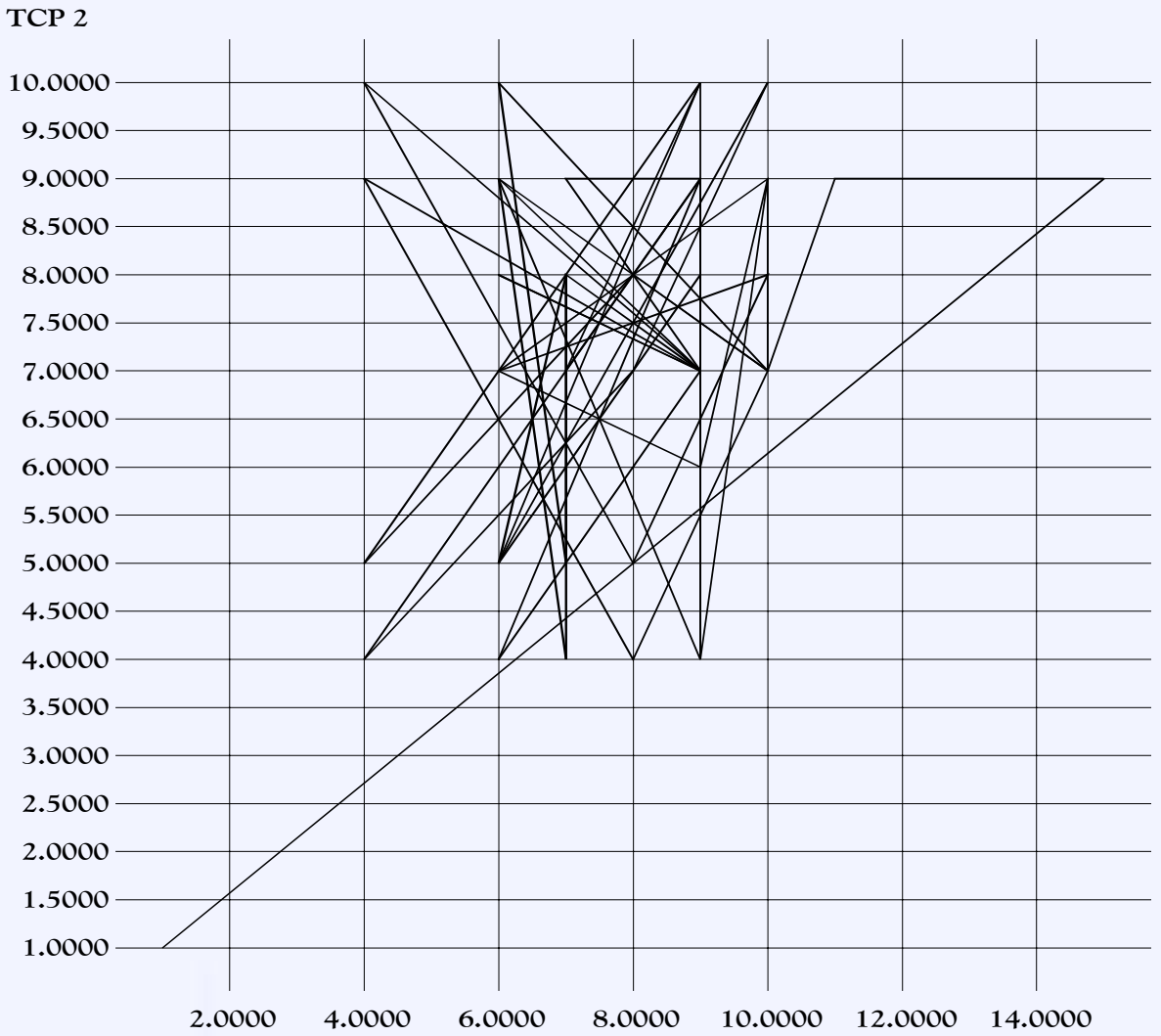


AIMD / asynchronous RTTs



- fluid model
- RTT: 7 vs. 2
- AI=0.1, MD=0.5
- Simul. time=175

AIMD in practise (TCP)



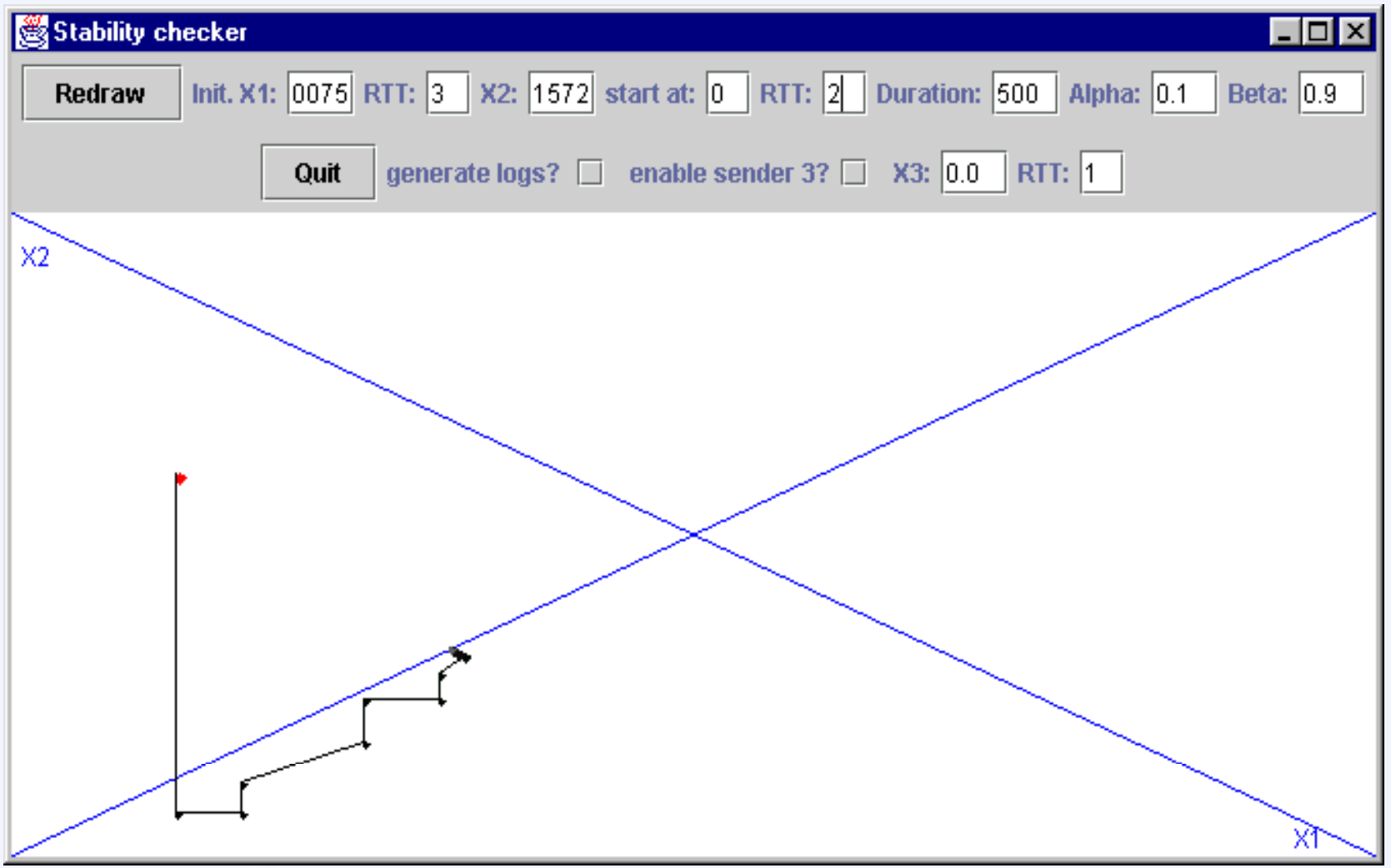
- *ns* simulator
- TCP Reno
- "equal" RTT
- 1 bottleneck link

CADPC Design

Endpoint Mechanism Design Algorithm(tm)

- find useful (closely related) ATM ABR mechanism
- start with simplifications, then expand the model
- A new mechanism must work for 2 users, equal RTT
  - simple analysis similar to Chiu/Jain (diagram + math)
- it must also work with heterogeneous RTTs
  - simulate using a simple Diagram Based Simulator(tm)
- it must also work with more users and in more realistic scenarios
  - simulate with ns

CADPC vector diagram analysis

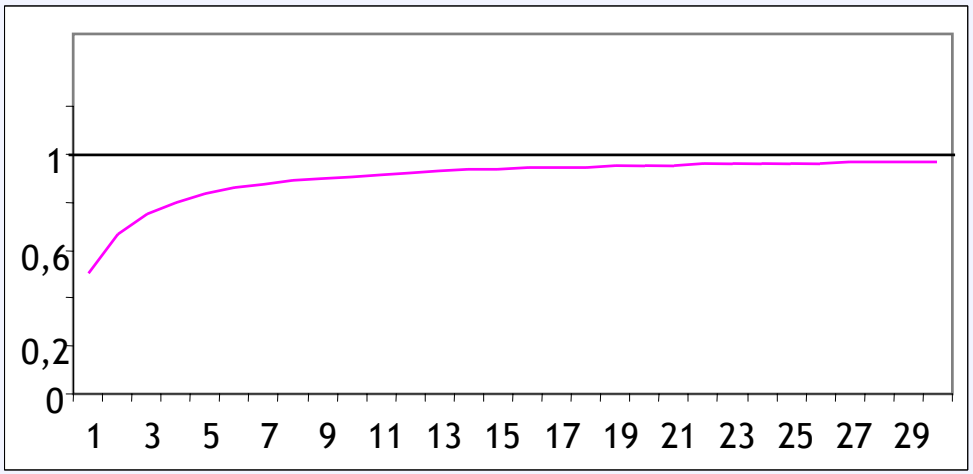


CADPC synchronous case fluid analysis

- **Final formula per user:**

  $x(t+1) = x(t)a(1-x(t)-traffic)+x(t)$

  x(t) ... normalised rate at time t
  a... smoothness factor
  (should be 0 < a <= 1)
  traffic (normalised) ... from PTP



- Converges to: n/(n+1)

- Continuous-time version of synchronous case *(traffic=nx)*:
  logistic growth $x'(t) = x(t)a(1-x(t)/c)$ *[c = 1/(1+n)]*
  asymptotically stable equilibrium point: c