# IASTED PDCN 2004 Tutorial:

# Shaping the Future of Internet Congestion Control

**Michael Welzl** http://www.welzl.at

**Distributed and Parallel Systems Group**
**Institute of Computer Science**
**University of Innsbruck**

# Outline

1. Congestion Control: a quick introduction
2. Problems
3. Some proposed enhancements
4. How to design your own congestion control mechanism

**Research procedure:**

1. get to know the field
2. identify problems
3. look at what others have done
4. come up with your own solutions

# Congestion Control

A quick introduction

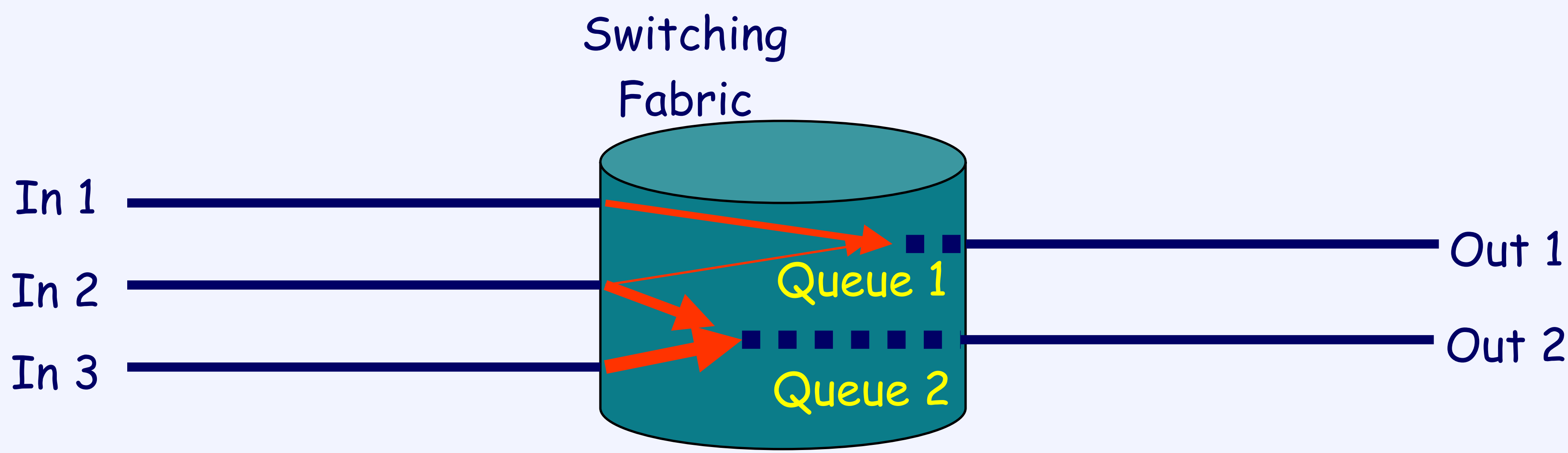
# Problem statement

- Efficient transmission of data streams across the Internet
  - various sources, various destinations, various types of streams
- What is „efficient"?
  - terms: latency, end2end delay, jitter, bandwidth (nominal/available/bottleneck -), throughput, goodput, loss ratio, ..
  - goals: high throughput (bits / second), low delay, jitter, loss ratio

typically bit/s at this level!

- Note: Internet = TCP/IP based world-wide network
  - no assumptions about lower layers!
  - ignore CSMA/CD, CSMA/CA, token ring, baseband encoding, frame overhead, switches, etc. etc. !

# A simple router model

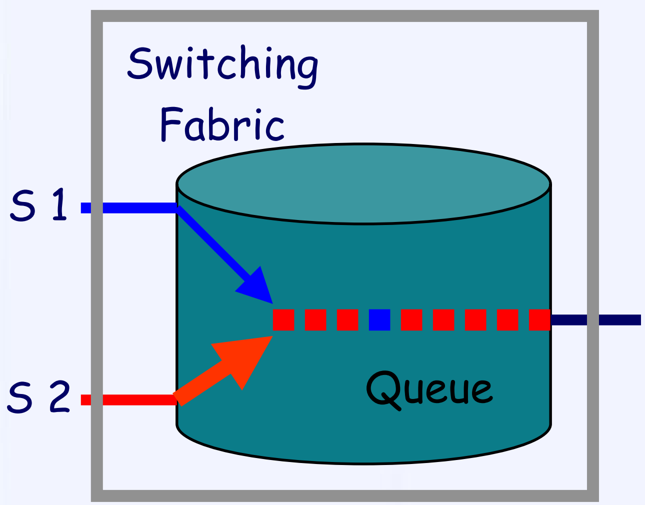


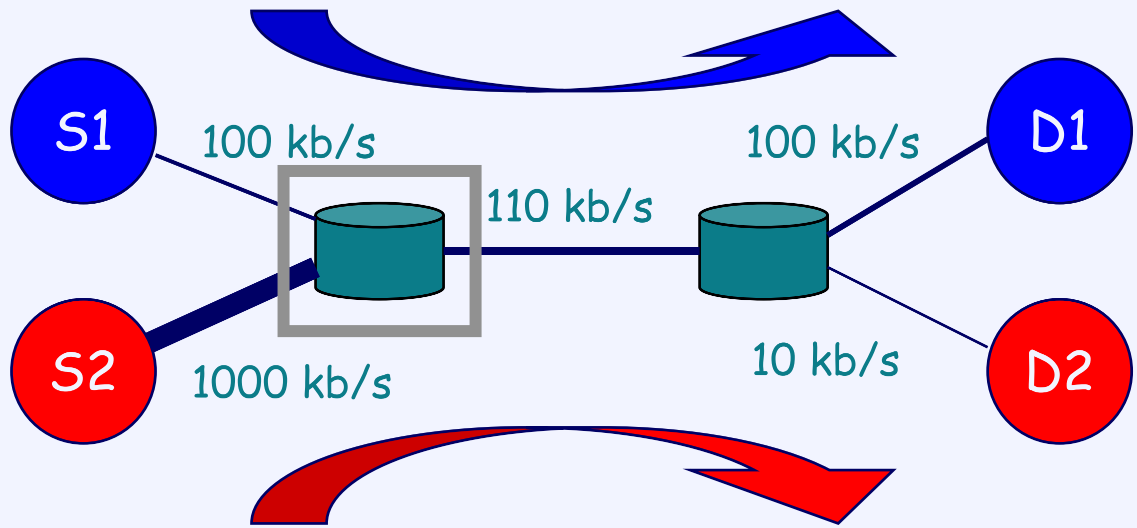

- Switching fabric forwards a packet (dest. addr.)
  if no special treatment necessary: fast path (hardware)

- Queues grow when traffic bursts arrive
  - low delay = small queues, low jitter = minor queue fluctuations
- Packets are dropped when queues overflow ("DropTail queueing")
  - low loss ratio = small queues

# The congestion problem

- Congestion control necessary
- adding fast links does **<u>not</u>** help!

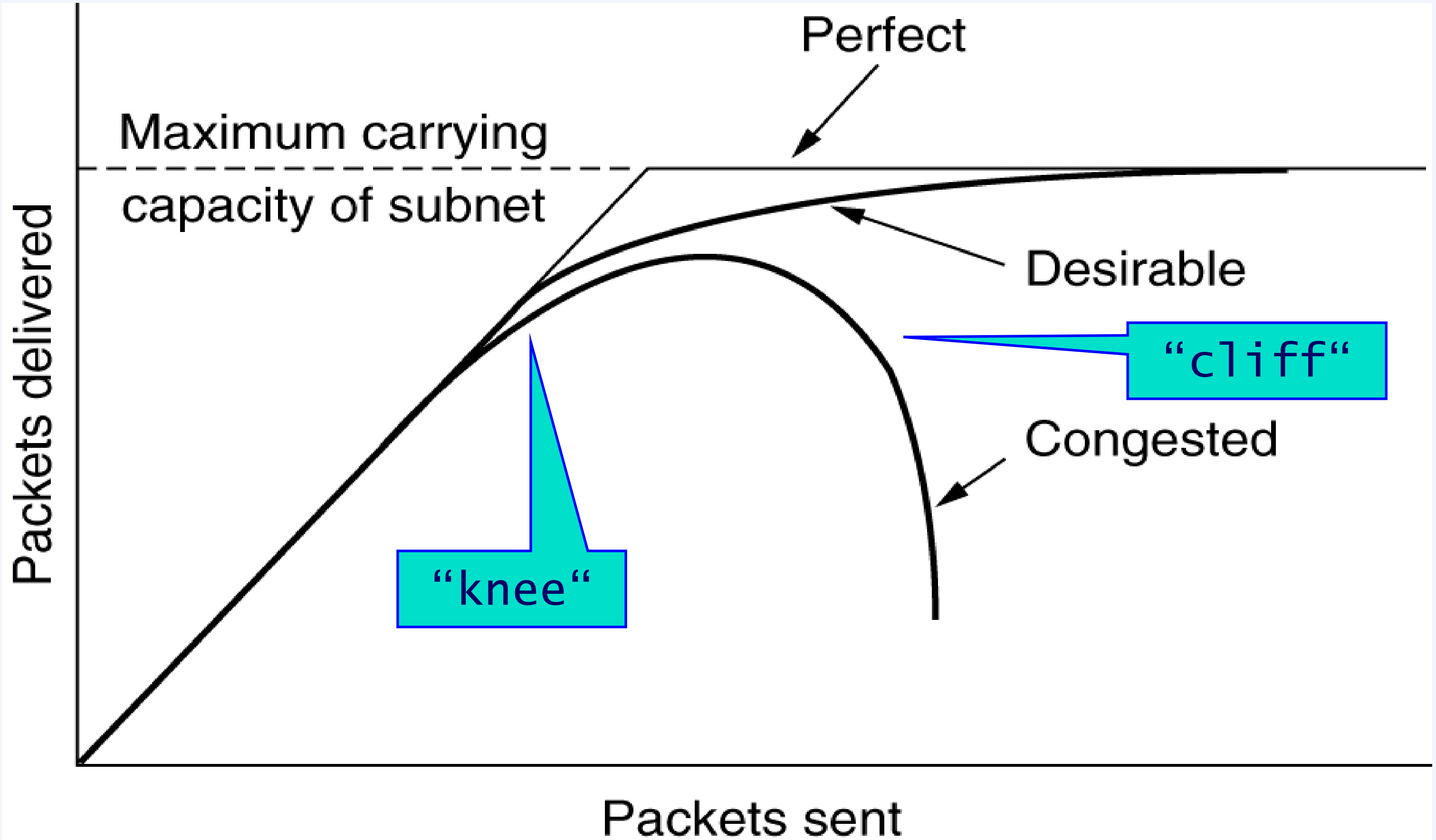



total throughput w/o cc.: 20kb/s

total throughput w/ cc.: 110kb/s

# Congestion collapse




Goal: operation at the "knee"

# Internet congestion control: History

- **1968/69:** dawn of the Internet
- **1986:** first congestion collapse
- **1988:** "Congestion Avoidance and Control" (Jacobson)
  Combined congestion/flow control for TCP

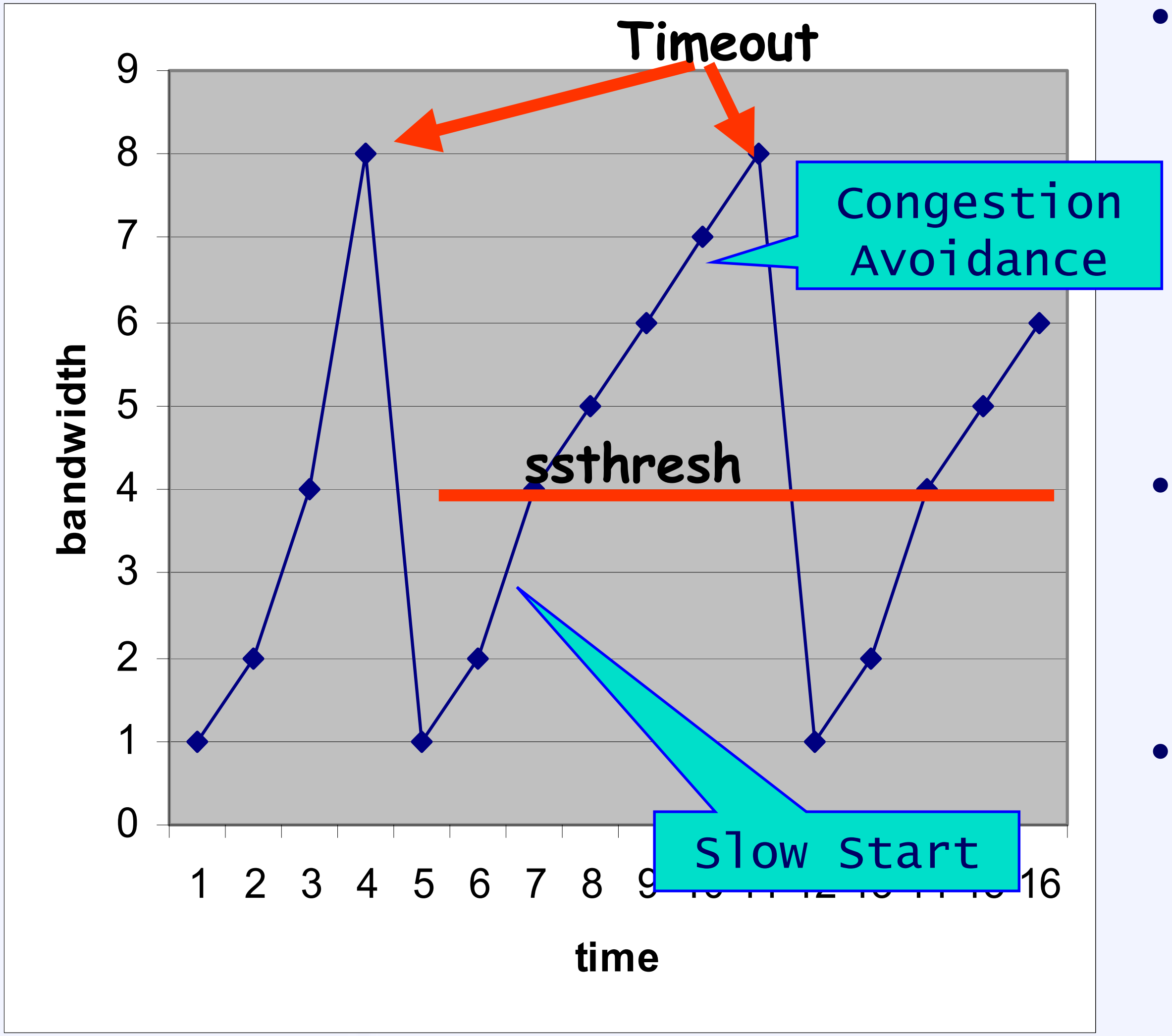
- Goal: stability - in equilibrum, no packet is sent into the network until an old packet leaves
  - ack clocking, "conservation of packets" principle
  - made possible through window based stop+go - behaviour

- Superposition of stable systems = stable →
  network based on TCP with congestion control = stable

# TCP Congestion Control /1: Tahoe, 1988

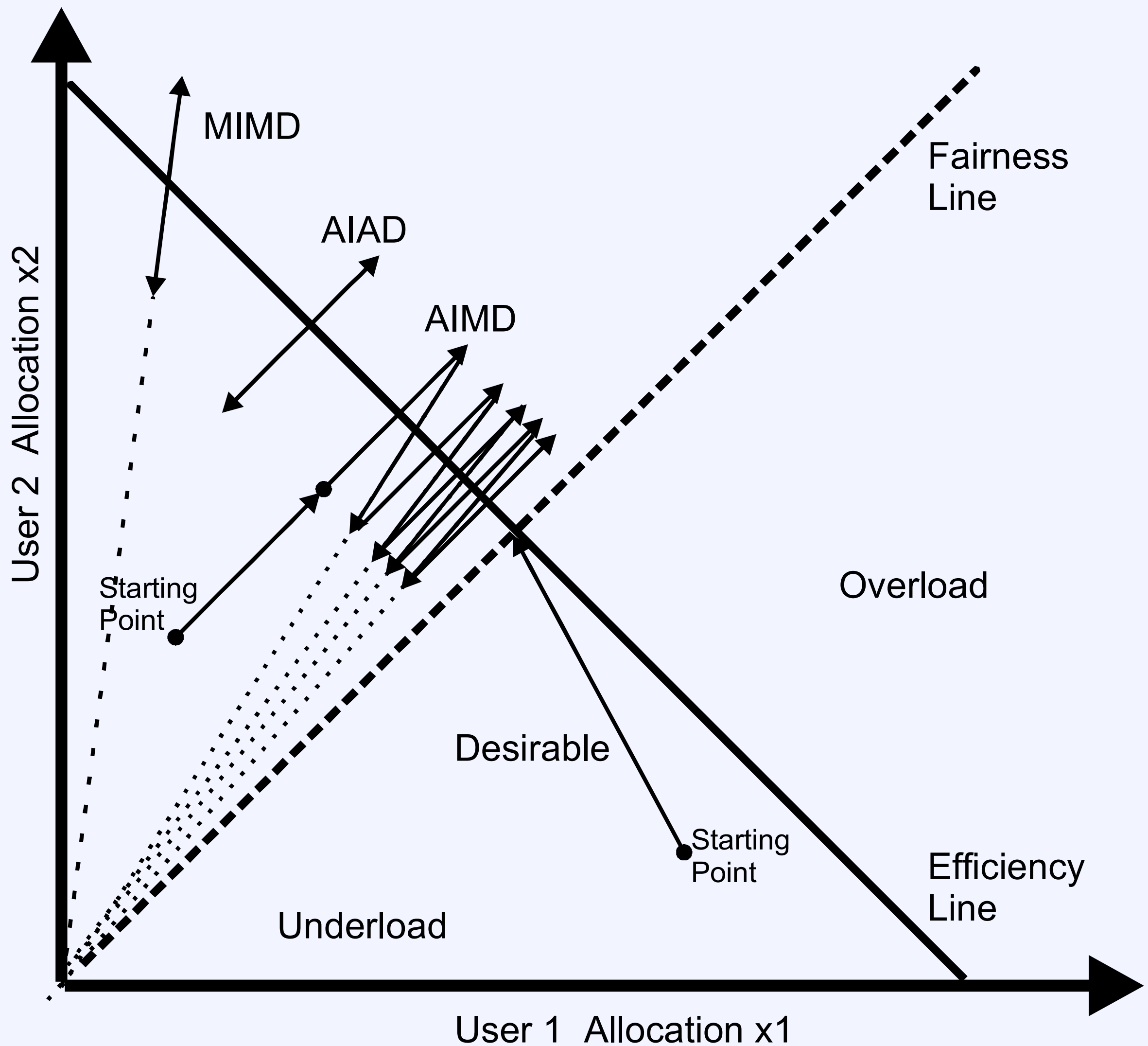
- Distinguish:
  - flow control: protect receiver against overload
    (receiver "grants" a certain amount of data ("receiver window") )
  - congestion control: protect network against overload
    ("congestion window" (cwnd) limits the rate: min(cwnd,rwnd) used! )
- Flow/Congestion Control combined in TCP. Several algorithms:
- (window unit: SMSS = Sender Maximum Segment Size, usually adjusted to Path MTU; init cwnd<=2 (*SMSS), ssthresh = usually 64k)
  - Slow Start: for each ack received, increase cwnd by 1
    (exponential growth) until cwnd >= ssthresh
  - Congestion Avoidance: each RTT, increase cwnd by SMSS*SMSS/cwnd
    (linear growth - "additive increase")

# TCP Congestion Control /2




- If a packet or ack is lost (timeout, roughly 4*rtt), set cwnd = 1, ssthresh = current bandwidth / 2 ("multiplicative decrease") - exponential backoff
- Several timers, based on RTT; good estimation is crucial!
- Later additions: *(TCP Reno, 1990)* Fast retransmit / fast recovery (notify sender of loss via duplicate acks)

# Background: AIMD



# Active Queue Management

- Today, TCP behaviour dominates the Internet (WWW, ..)
- (somewhat old) example backbone measurement: 98% TCP traffic
- **1993:** Random Early Detection ("Discard", "Drop") (RED)
  (now that end nodes back off as packets are dropped,
  drop packets earlier to *avoid* queue overflows)
- Another goal: add randomization to avoid traffic phase effects!
- ***Qavg = (1 - Wq) x Qavg + Qinst x Wq***
  (Qavg = average occupancy, Qinst = instantaneous occupancy,
  Wq = weight - hard to tune, determines how aggressive RED behaves)

# Active Queue Management /2

- Based on exponentially weighted moving average (EWMA) of instantaneous queue occupancy = low pass filter
  - recalculated every time a packet arrives

- *Qavg* below threshold *min_th*: Nothing happens
- *Qavg* above threshold *min_th*: Drop probability rises linearly
- *Qavg* above threshold *max_th*: Drop packets

- RED expects all flows to behave like TCP - but is it fair?

- Variants: drop from front, drop based on instantaneous queue occupancy, drop arbitrary packets, drop based on priorities...

# Explicit Congestion Notification (ECN)

- **1999:** Explicit Congestion Notification (ECN)
  Instead of dropping, set a bit

- End systems are expected to act as if packet was dropped
  $\Rightarrow$ actual communication between end nodes and the network!

- ATM and Frame Relay: not only ECN but also BECN

- Internet BECN: often proposed and regularly discussed (ICMP SQ), but very unlikely - several reasons

- Very popular among researchers - lots of ideas to exploit the bit!

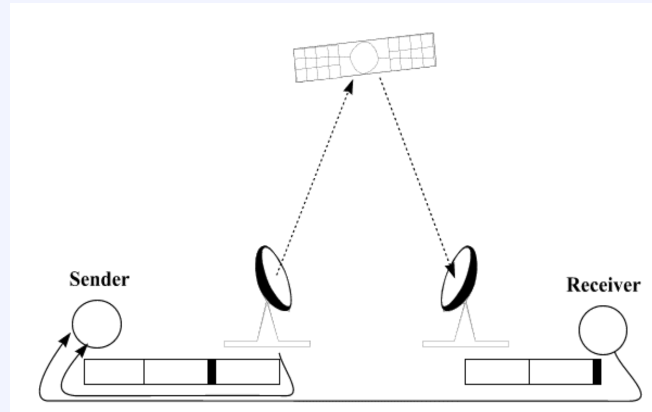
- ECN cannot totally replace loss measurements!

# Problems

( = potential research topics )

# TCP in heterogeneous environments

- TCP over noisy links: problems with „packet loss = congestion"
  - was bad idea in times of error-prone networks
  - seems reasonable in times of fibre networks
  - really bad for wireless links!




- TCP over "long fat pipes": large bandwidth*delay product
  - long time to reach equilibrium, MD = problematic!

- TCP in highly asymmetric networks: (e.g. direct satellite last mile)
  - incoming throughput (high capacity link) limited by rate of outgoing ACKs („ACK compression")

# Fairness

- ATM ABR: Max-Min-fairness
  - "A (..) allocation of rates is max-min fair iff an increase of any rate (..) must be at the cost of a decrease of some already smaller rate."
  - One resource: mathematical definition satisfies "general" understanding of fairness - resource is divided equally among competitors
  - *Usually* requires knowledge of flows in routers (switches) - scalability problem!

- Internet:
  - TCP dominant, but does not satisfy Max-Min-fairness criterion!
  - Ack-clocked - flows with shorter RTT react sooner (slow start, ..) and achieve better results
  - Therefore, Internet definition of fairness: TCP-friendliness

    "A flow is TCP-compatible (TCP-friendly) if, in steady state, it uses no more bandwidth than a conformant TCP running under comparable conditions."

# Issues with TCP-friendliness

- TCP regularly increases the queue length and causes loss
  ⇒ detect congestion when it is already (ECN: almost) too late!
  - possible to have more throughput with smaller queues and less loss
    ... but: exceed rate of TCP under similar conditions ⇒ not TCP-friendly!

- What if I send more than TCP *in the absence* of competing TCP's?
  - can such a mechanism exist?
  - yes! TCP itself, with max. window size = bandwidth * RTT
  - Does this mean that TCP is not TCP-friendly?

- Details missing from the definition:
  - parameters + version of „conformant TCP"
  - duration! short TCP flows are different than long ones

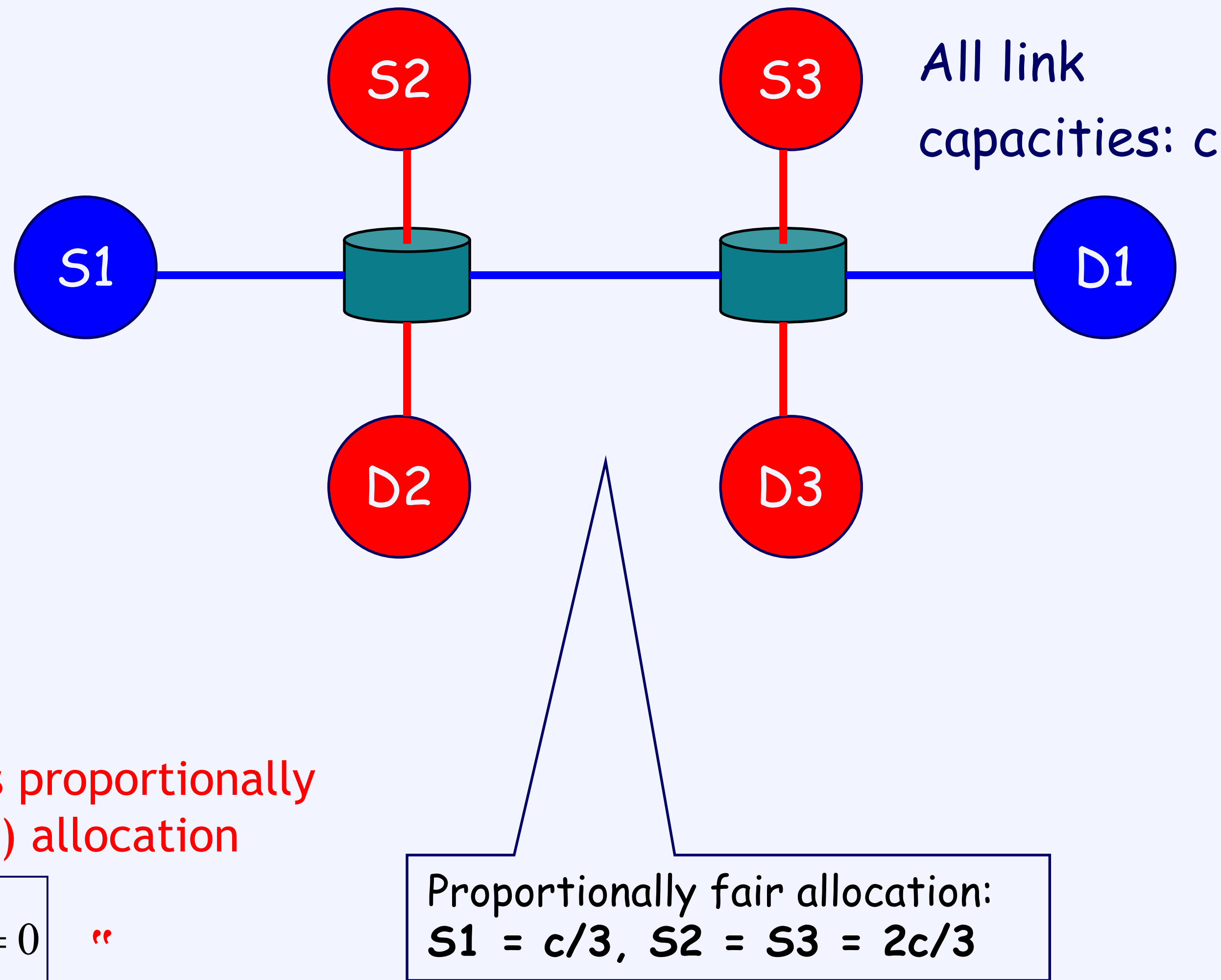
Does TCP-friendliness hinder research?

- TCP-friendliness = compatibility of new mechanisms with old mechanism
  - there was research since the 80's! e.g. new knowledge about network measurements

- TCP rate depends on RTT - how does this relate to „fairness"?

# Proportional Fairness

F. Kelly: Network should solve a global optimization problem (maximize log utility function)




Max-Min-fairness suboptimal:
S1 = S2 = S3 = c/2

Proportional fairness:
"An allocation of rates $\vec{x}$ is proportionally fair iff, for any other (..) allocation $\vec{y}$, we have: $\boxed{\sum_{s=1}^{S} \frac{y_s - x_s}{x_s} <= 0}$ "

*(__roughly__ approximated by AIMD!)*

# Congestion pricing

- Basic idea: higher sending rate = more congestion = higher price
  - Idea: charge more when ECN flag is set

- „Smart" Market idea:
  - each packet bids for capacity
  - out of $n$ bids, $m$ highest bids can be accepted
  - price: „marginal cost" (highest bid of unaccepted packets)
    $\Rightarrow$ leads to market equilibrium

- „Smart" Market not practical (bidding per packet), but shows:
  - balance of demand and supply leads to market equilibrium
  - stable system just based on economics suffices for congestion control

- Problem: link cannot know bids in advance $\Rightarrow$ no QoS guarantees

# End2end real-time data transfer

- Assumption: no special service available at application level
  - (Definition of Internet *"real-time"* softer than usual)

- Different requirements:
  - reliable service may not be needed (no retransmission)
  - Timely transmission important

- Different treatment:
  - no retransmission / waiting for ACKs
  - no sliding window (stop + go behaviour not suitable)
- but:
  - some kind of flow control still needed
  - synchronization necessary
  - often: Multicast

# Multimedia adaptation

- Mistake:
  - adaptation schemes assume arbitrary data stream scalability
- Problem:
  - Data streams show fluctuations (example: MPEG I-, B-, P-frames)
- Solution:
  - Special CBR design for communication - H.261 designed for ISDN
  - not always feasible
- Problem:
  - compression usually not deterministic - size depends on content!
  - real-life distance learning example:
    40kbps enough for streaming video (Smartboard) + audio (speech), but speech suffers dramatically if teacher visible

# Congestion Control and Quality of Service

- Quality of Service (QoS): provide differentiated quality based on $$$

...modest QoS with adaptive multimedia? - still unresolved!

- Questions:
  - is a fluid low-quality video better than a bucking high-quality video?
  - and what about audio?

- Scalable QoS = no per-flow guarantees
  - standard architecture: Differentiated Services (DiffServ) places users in flow aggregates
  - congestion control still necessary within an aggregate
    ⇒ per-flow QoS depends on congestion control mechanism!
  - How does a congestion control mechanism interact with QoS elements?
    e.g. TCP known to be a bad match for token bucket

# Special types of traffic

- Grid: predictable traffic pattern
  - This is totally new to the Internet!
    - Web: users create traffic
    - FTP download: starts ... ends
    - Streaming video: either CBR or depends on content! (head movement, ..)
- Special requirements and properties
  - may require delay bounds or minimum bandwidth
  - mixture of sporadic (RPC type) messages and bulk data transfer
- Related: signaling traffic
  - usually not a large amount of data
  - to date, no serious efforts for tailored congestion control
    (SCTP = transport protocol designed for signaling; congestion control = similar to TCP)

# Some reasons for TCP stability

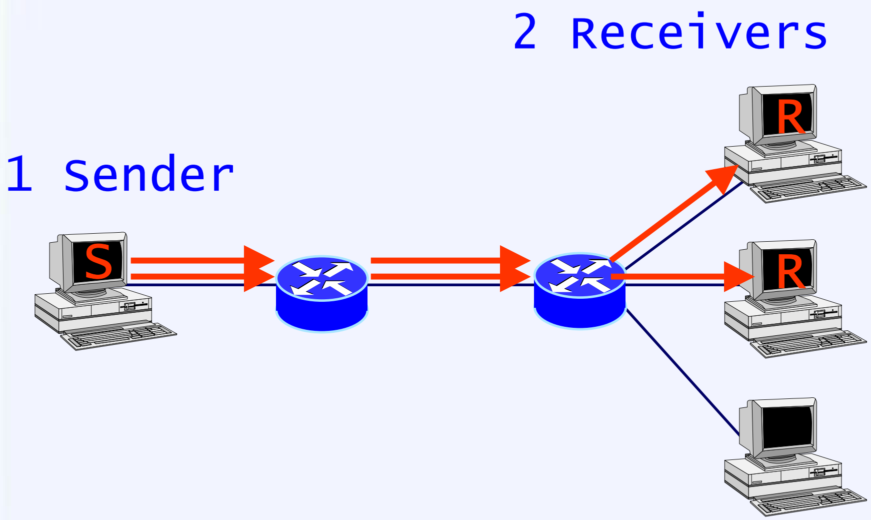
"Congestion Avoidance and Control", Van Jacobson, SIGCOMM'88:

- Exponential backoff:
  "For a transport endpoint embedded in a network of unknown topology and with an unknown, unknowable and constantly changing population of competing conversations, only one scheme has any hope of working - exponential backoff - but a proof of this is beyond the scope of this paper."
- Conservation of packets:
  "The physics of flow predicts that systems with this property should be robust in the face of congestion."
- Additive Increase, Multiplicative Decrease:
  Not explicitely cited as a stability reason **in the paper**!
  - ...but in 1000's of other papers!

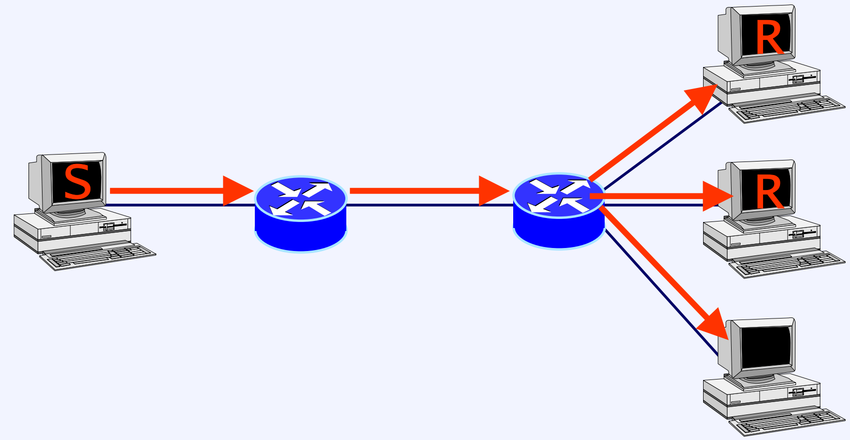
# “Proofs“ of TCP stability

- AIMD:
  Chiu/Jain: diagram + algebraic proof of homogeneous RTT case
- steady-state TCP model: window size ~ 1/sqrt(p)
  (p = packet loss)
- Johari/Tan, Massoulié, ..:
  - local stability, neglect details of TCP behaviour (fluid flow model, ..)
  - assumption:
    “queueing delays will eventually become small relative to propagation delays“
- Steven Low:
  - Duality model (based on utility function / F. Kelly, ..):
    Stability depends on delay, capacity, load and AQM !

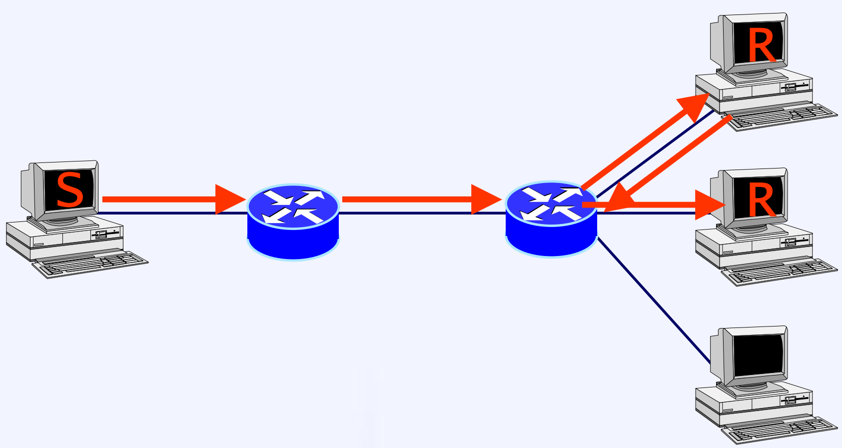
# Unicast / Broadcast / (overlay) Multicast



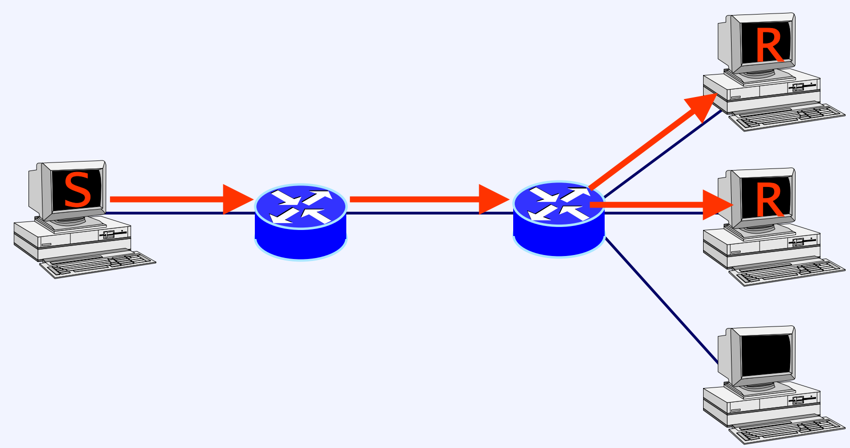

Unicast




Broadcast




Overlay Multicast




IP Multicast

# Multicast issues

- Required for applications with multiple receivers only
  - video conferences, real-time data stream transmission, ..
    $\Rightarrow$ different data streams than web surfing, ftp downloads etc!

- Issues:
  - group management
    - protocol required to join / leave group dynamically: Internet Group Management Protocol (IGMP)
    - state in routers: hard / soft (lost unless refreshed)?
    - who initiates / controls group membership?
  - congestion control
    - scalability (ACK implosion)
    - dealing with heterogeneity of receiver groups
    - fairness

- Multicast congestion control mechanism classification:
  - sender- vs. receiver-based, single-rate vs. multi-rate (layered),
  - reliable vs. unreliable, end-to-end vs. network-supported

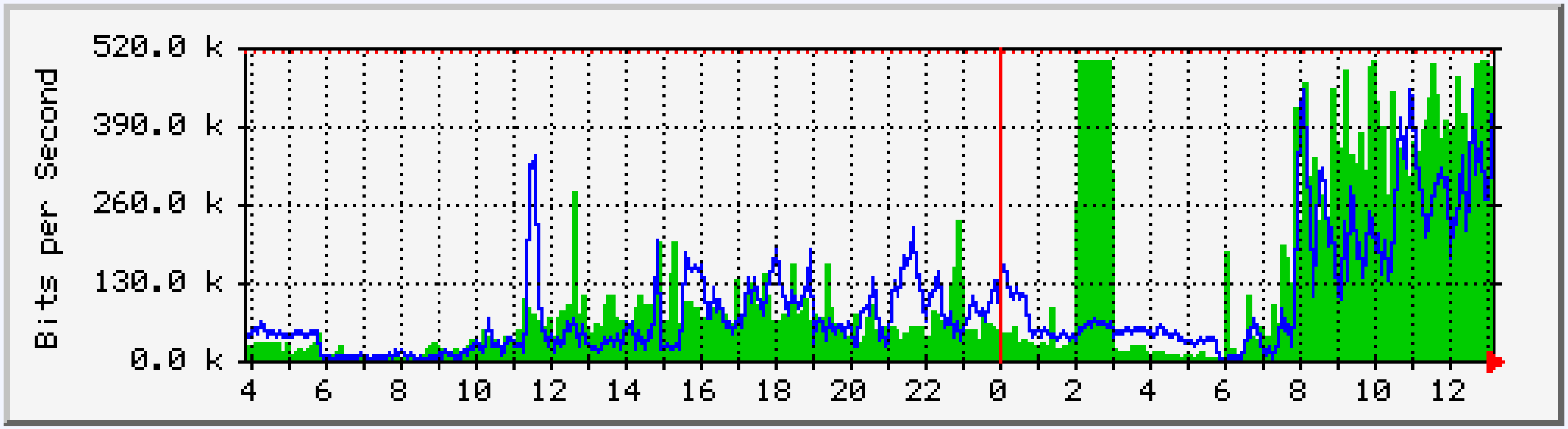
depends on content!

# Some proposed enhancements

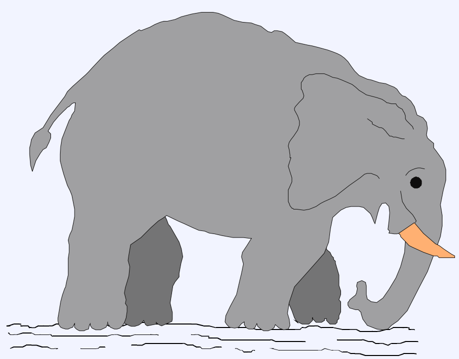
Research by others

# Internet traffic characteristics

- MRTG trace (based on SNMP, accessing traffic counters in MIB)




http://www.switch.ch/lan/stat/peerings/linkeunet.html, 11. 10. 99, 13:05



# Internet traffic characteristics /2

- Traditional traffic modelling: queuing theory notion: traffic follows poisson distribution

Still true for user arrival !

- Internet traffic is bursty - intuitive reasons:
  - TCP is bursty by nature: congestion avoidance, payload vs. acks…
  - ACK compression can cause payload bursts due to ACK-clocking *(later!)*
  - various packet sizes
  - Bursts from queues aggregate as traffic traverses the net
  - Burstiness of one flow affects other adaptive flows

# Internet traffic characteristics /3

- Overlapping of independent on-off sources leads to distribution with heavy-tailed autocorrelation function
- Long-range dependance: "peaks sit on ripples which sit on waves"
- No "flattening" towards a mean as you zoom out - same structures may be found at different time scales, hence self similar
- characteristics modeled with time series (fARIMA models) or wavelets
  - Measurement of the "degree of self similarity": Hurst parameter
  - $\Rightarrow$ model approximation involves Hurst parameter estimation
- TCP known to propagate bottleneck self-similarity to end system
  - possibility: use model to predict traffic instead of guessing
  - question: scalability (what if everybody does this?)

# How to be TCP-friendly

- TCP-friendliness can be achieved by emulating the behaviour of TCP (or the desired parts of it)

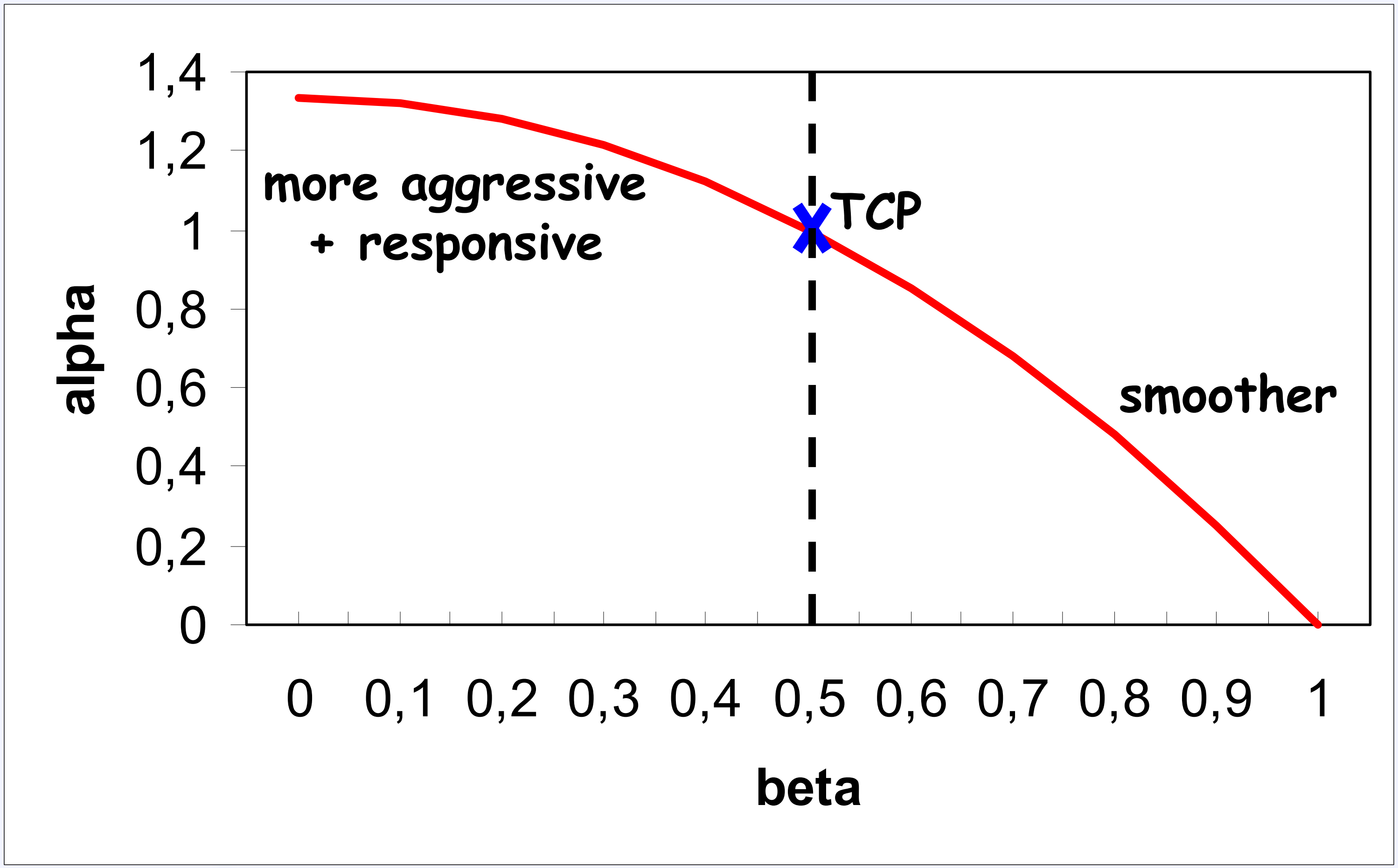
- Simplified TCP: AIMD (additive incr. $\alpha$ , multiplicative decr. $\beta$)
  - $0 < \alpha$ , $0 < \beta < 1$ -> stable and fair congestion control
  - $\alpha = 4 \times (1 - \beta^2) / 3$ -> TCP-friendly congestion control (GAIMD)
  - $\alpha = 1$, $\beta = 1/2$ -> TCP

- AIMD mechanisms for multimedia applications: RAP, LDA+

- Different approaches:
  - TCP Emulation At Receivers (TEAR)
    TCP calculations (cwnd calculation, fast recovery, ...) moved to receiver, do not ack every packet, smooth sending rate
  - Binomial congestion control: generalization of GAIMD with nonlinear control
  - CYRF framework: generalization of binomial congestion control

# GAIMD congestion control

Relationship between $\alpha$ and $\beta$ for TCP-friendliness:



# Equation based congestion control

- Based on TCP steady-state response function
  - gives upper bound for transmission rate T (bytes/sec):

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)}$$

s: packet size
R: rtt
$t_{RTO}$: TCP retransmit timeout
p: steady-state loss event rate (the difficult part!)

- well known example: **TFRC** - TCP-friendly rate control protocol
  - smooth sending rate
- Extension: TFMCC - TCP-friendly multicast congestion control

# Not-so-TCP-friendly solutions

- Overcome rate fluctuations:
  limit encodings (e.g. 2 or 3 qualities), let user decide
- Cross-media-adaptation:
  choose from video, audio, single pictures, text
  (e.g. MPEG7)
- Limit by bottleneck bandwidth
  - often: "last mile" - e.g. RealMedia
  - better: determine actual bottleneck via packet pair approach

Network stability: <u>some</u> adaptation better than none!

- If wireless link involved: small packets, UDP Lite
- Another possibility: send more (do FEC) in response to packet loss
  - (very network-unfriendly behaviour, but may yield less data loss)

# Some thoughts

- How TCP-friendly are 8 web browsers?
  - Congestion Manager: congestion control for all flows in OS core
  - MulTCP: Emulate multiple TCP's to provide differentiated services
- How TCP-friendly are short-lived flows? (web-traffic, ..)
- How to convince Internet multimedia app. programmers to implement TCP-friendly congestion control?
- Solution: Datagram Congestion Control Protocol (DCCP)
  - Well-defined framework for (TCP-friendly) congestion control
  - Sender app chooses an appropriate congestion control mechanism
  - Core OS implementation of mechanisms
  - Lots of additional features: nonces, partial / separate checksums (distinguish: corruption ⇔ congestion), ...

# Heterogeneous environments

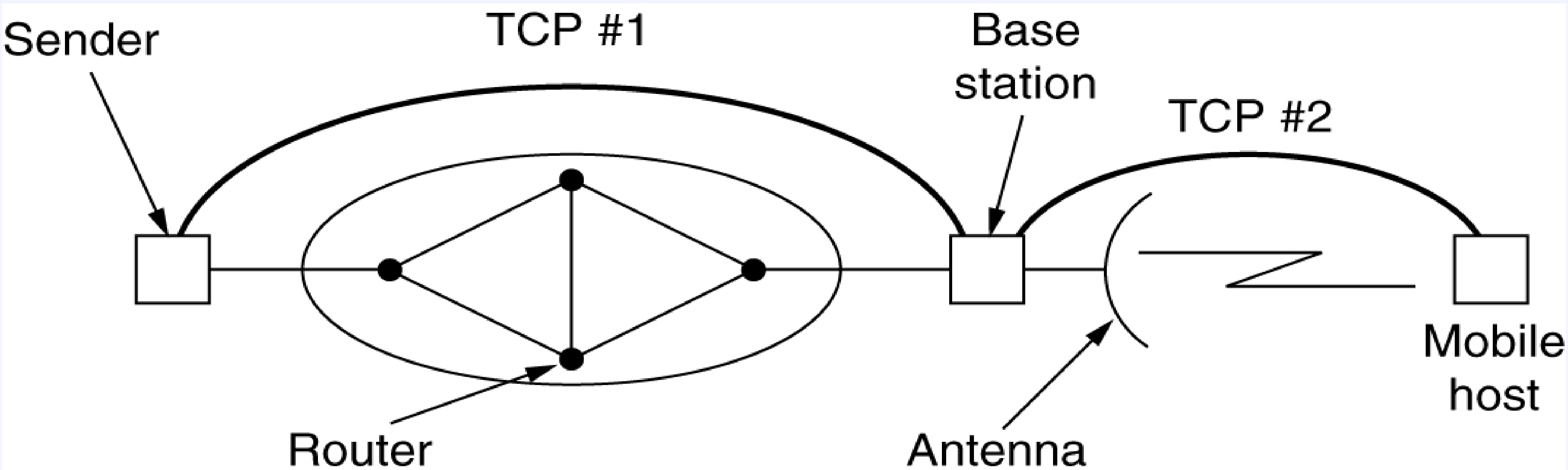
TCP over high speed links:

- larger initial window / window scaling option, TCP SACK
- Scalable TCP: increase/decrease functions changed
  (probing times proportional to rtt but not rate)
- HighSpeed TCP (merged with Scalable TCP):
  response function less drastic in high bandwidth environments *only*
- Quick-Start: query routers for initial sending rate with IP options
  draft only - seems to be abandoned?!

TCP over asymmetric links:

- ACK suppression, ACK compaction, TCP header compression

# TCP over noisy (wireless) links

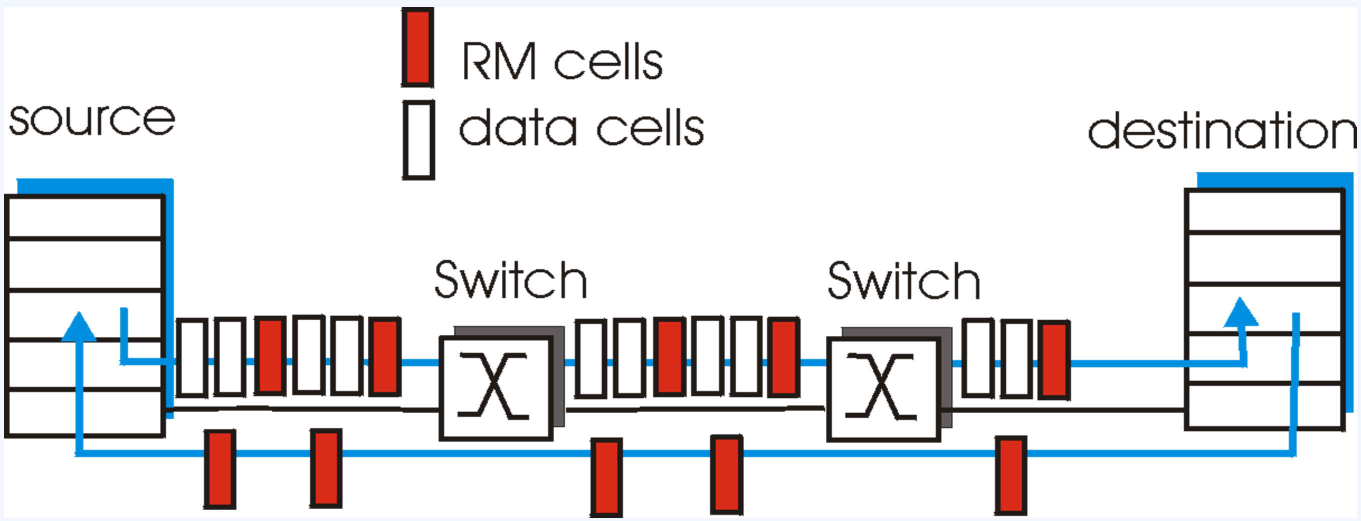



- Various possible enhancements:
  - split connection at base station
  - monitor connection at base station, buffer + interfere ("Snoop TCP")

- Note: ECN is not affected by link noise!

# Beyond ECN

- ATM: Explicit Rate Feedback (part of Available Bit Rate (ABR) service)
  RM (resource management) cells:
  - sent by sender, interspersed with data cells; bits in RM cell set by switches
    - NI bit: no increase in rate (mild congestion), (EF)CI bit: like Internet ECN
    - two-byte ER (explicit rate) field: may be lowered by congested switch
    - sender' send rate thus minimum supportable rate on path!




- Problem: ATM failed (scalability? too much complexity in switches?)
- Experimental Internet approaches:
  - Multilevel ECN (two bits), eXpress Control Protocol (XCP), CADPC/PTP (my own)

# Other TCP enhancements

- FAST TCP
  - Variant based on window and delay
  - Delay allows for earlier adaptation (awareness of growing queue)
  - Proven to be stable
  - Commercially announced + patent protected, by Steven Low's CalTech group
  - another delay-based example: TCP Vegas
- TCP Westwood
  - different response function (proportional to rate)
  - proven to be stable
- Lots of experimental Active Queue Management schemes out there
  - Adaptive RED, BLUE, REM, RIO etc. etc.

# How to design your own mechanism

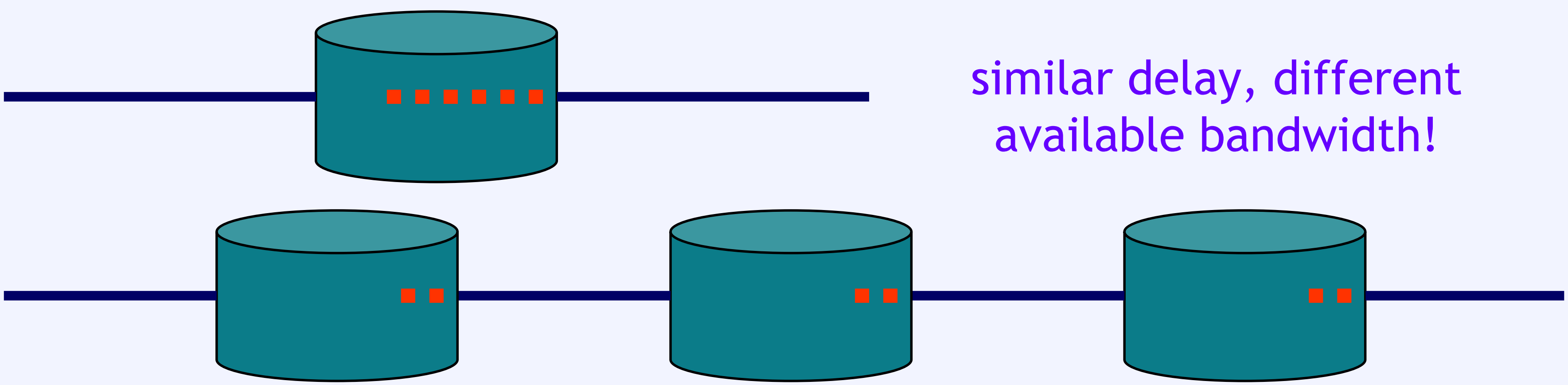
# End systems: Measure...

**throughput** ("goodput")
( mean, fluctuations,
packet loss ratio..)

+ well studied
- leads to misinterpretation of transmission errors

**delay**
( rtt, one way delay,
jitter..)

+ easy to measure
+ independent of transmission errors
- not practical without throughput



similar delay, different available bandwidth!

# ... and change!

- **lower layers**
  - throughput (gap between packets)
    - window based / rate based
    - well studied, many options - **our main interest!**
  - packet size
    - large: recommended
      less overhead
    - small: less impact of transmission errors,
      **smaller latency!**
  - protocol

**Note:** packet size = granularity of throughput measurements

- **content**
  - compression
  - hierarchical encoding
  - FEC

**Common difficulties:**
bandwidth known (depending on content)?
granularity of rate changes

# Measuring the network

- When you measure, you measure the past
  - predictions / estimations with a ?? % chance of success or control theory
- When you measure, you change the system
  - think of unresponsive flows vs. TCP
  - non-intrusiveness really important (e.g., monitor TCP behavior)
- Measurements yield no guarantees
  - Internet traffic = result of user behavior!
- Research carried out in controllable, isolated environments
  - Field trials are a necessary extra when you know that something works

# Possibilities in routers

- Communicate with end systems
  - alter header flags (IP only! should not look at other layers)
  - generate signaling packets: Internet Control Message Protocol (ICMP) (mainly error messages)

- Control packets in queues:
  use queue length or position in queue to
  - communicate („mark")
  - drop, move to other queue etc.

- Problems
  - CPU cycles scarce in (core) Internet routers
  - Scalability! (e.g., no per-flow state)
    example: ICMP Source Quench failed
    (congestion notification in times of congestion)

# My Ph.D. recipe :-)

Underlying thought:

*"TCP always exceeds the available bandwidth in order to detect it (when it is already too late). Wouldn't it be better to ask for the available bandwidth?"*

Process:

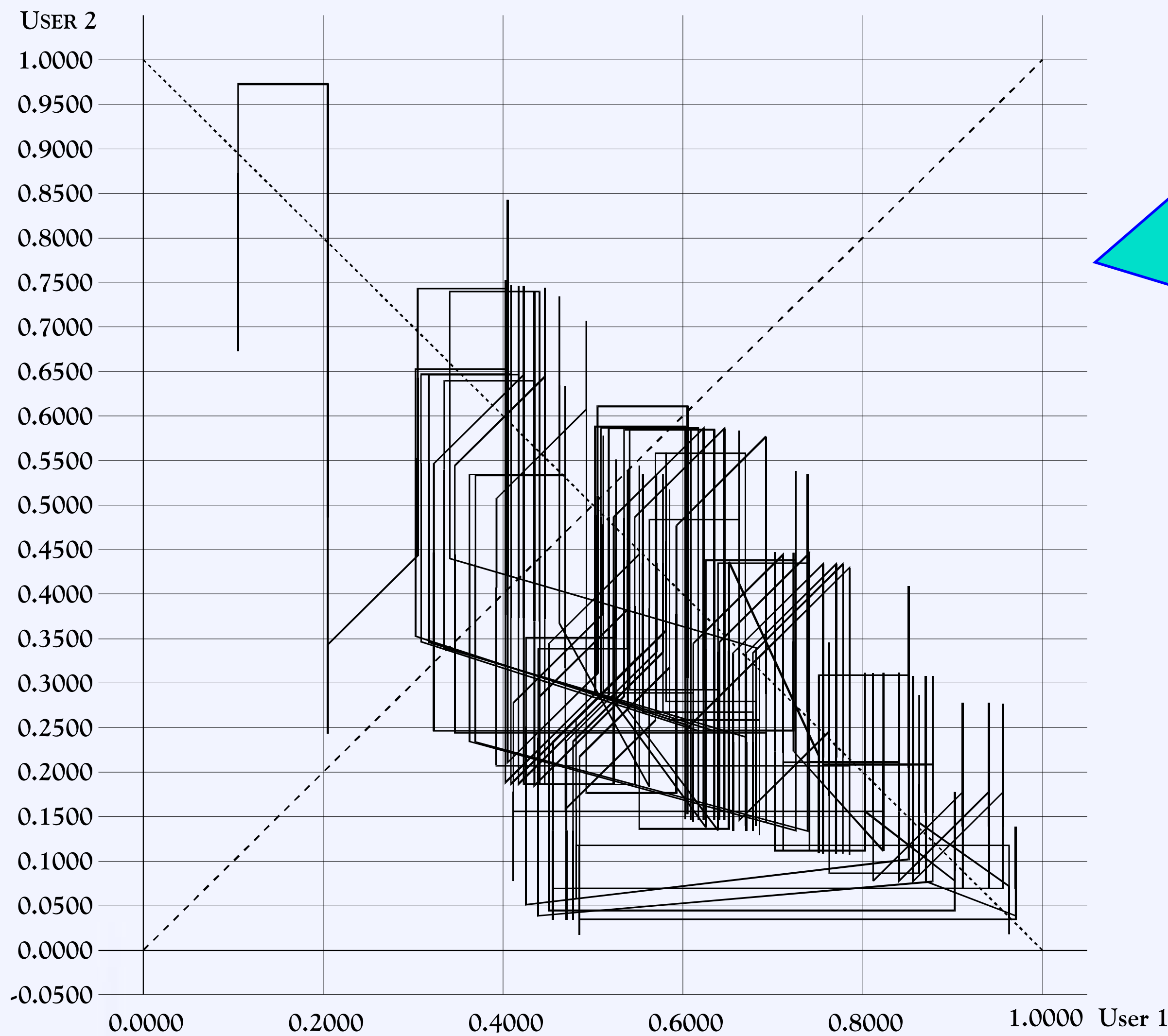
- design such a means: "Performance Transparency Protocol (PTP)"
  - note: must lead to absolutely great results in order to justify router effort!
- find out how to use the (available bandwidth) information ...without being a control theory guru!
  $\Rightarrow$ the tricky part!
- mixture of intuition, maths, simulation, ..

Let's look at this!

# Extended Use of Vector Diagrams

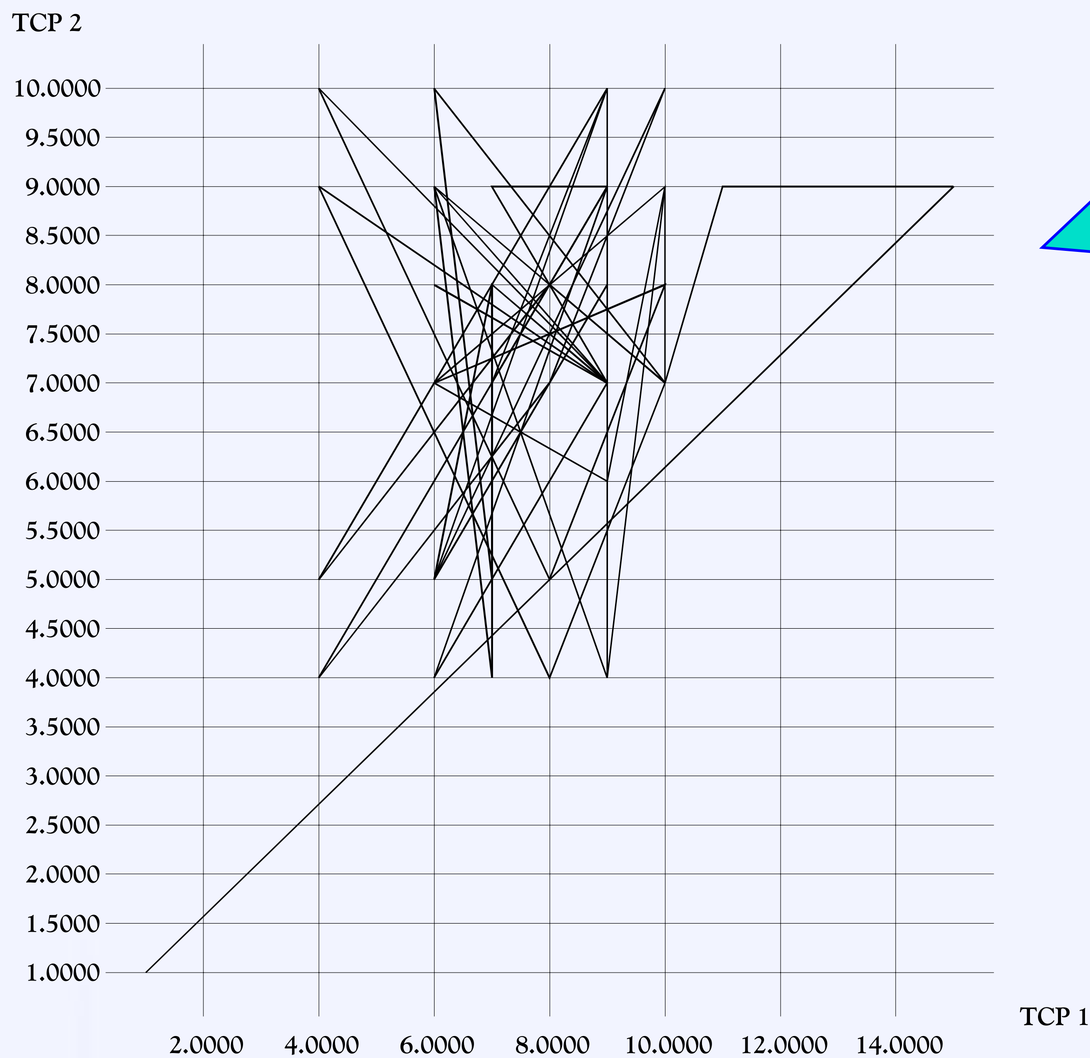
- Problem:
  - Stability analysis complex
  - TCP-like mechanism design difficult
- Solution:
  - Extended use of vector diagrams!
- Analyze actual results (from simulation or real life measurements)
- Instead of just explaining a concept, *design* in the 2D diagram space!
  - Necessary simplifications may even be less dramatic!

# How Stable is AIMD / async. RTT?




Fluid model
(no queues, ..)
RTT: 7 vs. 2
AI=0.1, MD=0.5
Sim. time=175

# Is AIMD distorted in TCP?

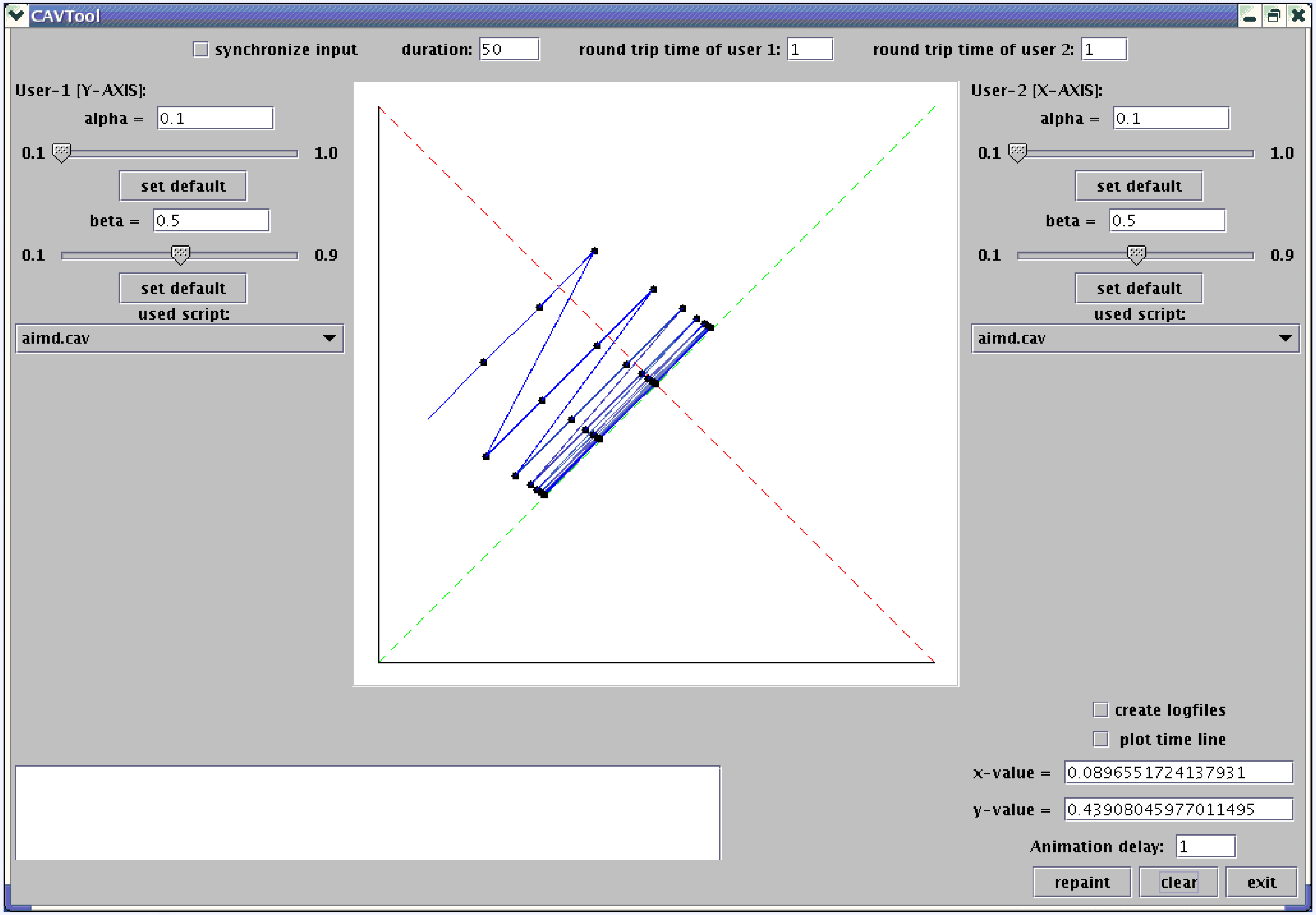



- ns-2 simulator
- TCP Tahoe
- equal RTTs
- 1 bottleneck link

# Various other Possibilities

- Analyze real life data
- Analyze different mechanisms
  - more complex feedback: ATM ABR
  - queueing behaviour: AQM
  - ...
- Perform analysis in vector diagram space
  - plot "distance from optimality" / time development
  - plot time of convergence / user 1 allocation
  - ...
- ... vector diagram aided design!

# Interactive Vector Diagram Aided Design



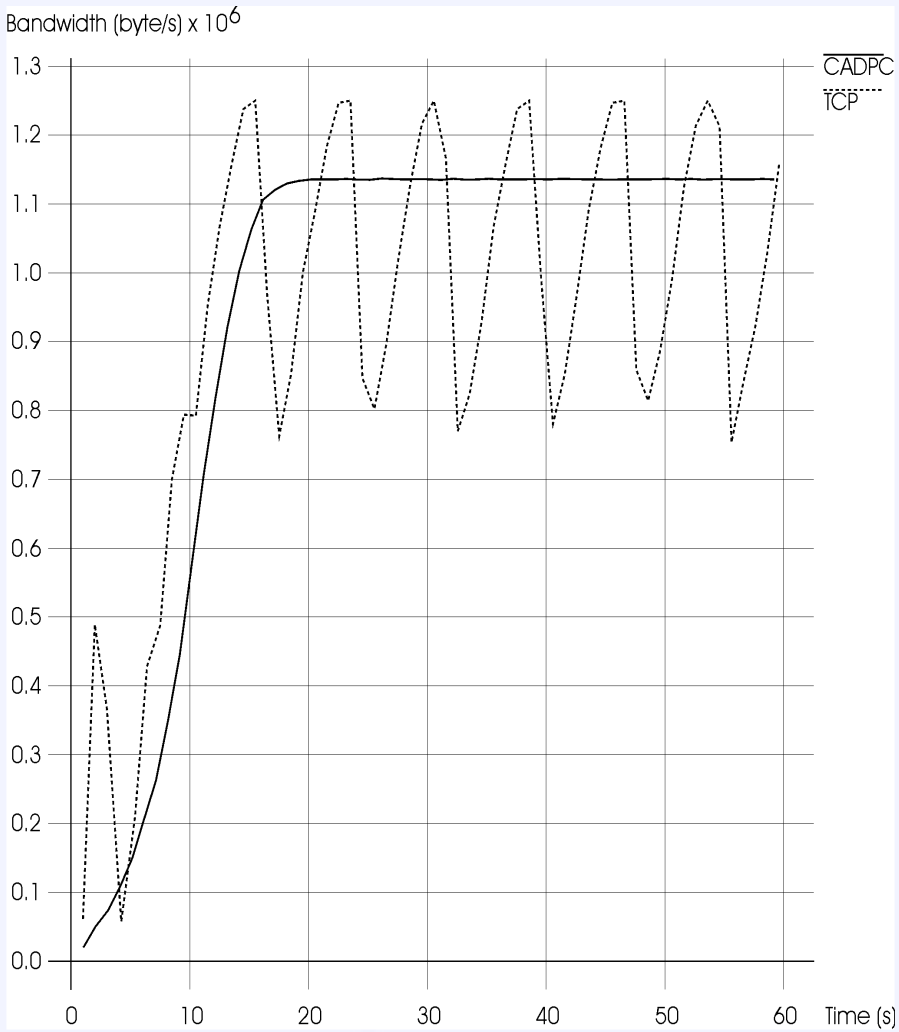
# A foolproof Ph.D. thesis recipe

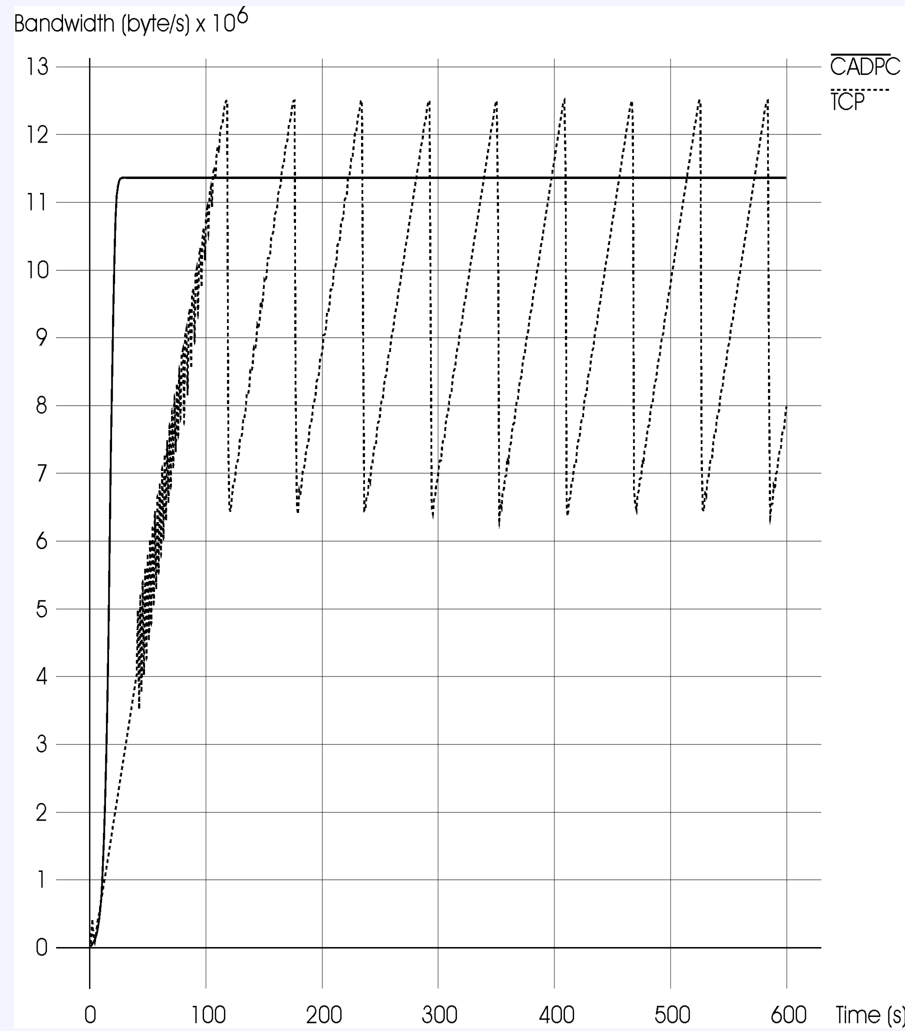
| Abstraction level | No. users | RTTs | No. resources | Traffic model | Method |
|---|---|---|---|---|---|
| 1 | 1 | equal | 1 | fluid | maths |
| 2 | 2 | equal | 1 | fluid | vector diagrams |
| 3 | n | equal | 1 | fluid | maths |
| **4** | **2** | **heterogeneous** | **1** | **fluid** | **CAVT** |
| 5 | n | heterogeneous | 1 | discrete | "normal" simulation |
| 6 | n | heterogeneous | m | discrete | "normal" simulation |
| 7 | n | heterogeneous | m | discrete | real life experiments |

- Roughly follow this table from 1 to 7 ... if something fails, go back!

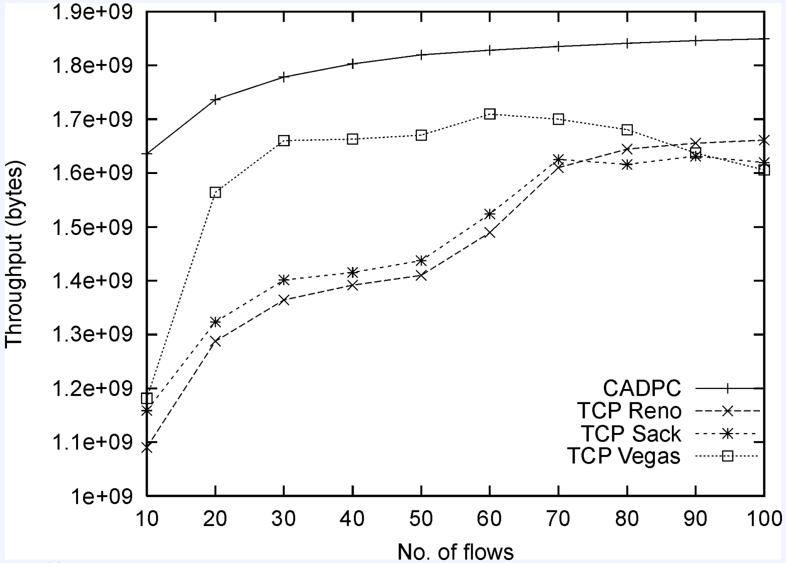
- In my case:
  - Level 2: intuition (fooling around with CAVT)
  - Level 4: the "Heureka" experience - intuition was right!
  - Level 1: sheet of paper
  - Level 3: MS Excel + CAVT refinement (fooling around, part 2)
  - Level 5: the typical ns + dumbbell experience
  - Level 6: additional simulations
  - ... that was good enough. Hoping to reach level 7 soon.

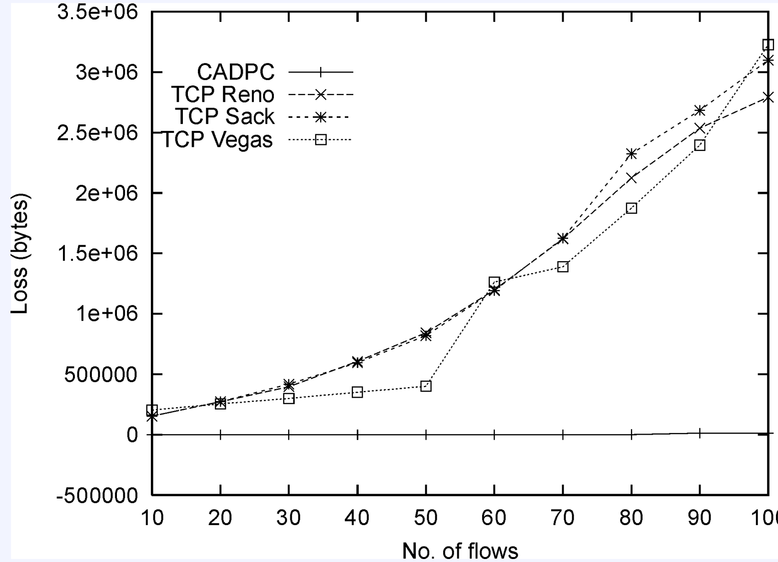
# Eventually: CADPC vs. TCP

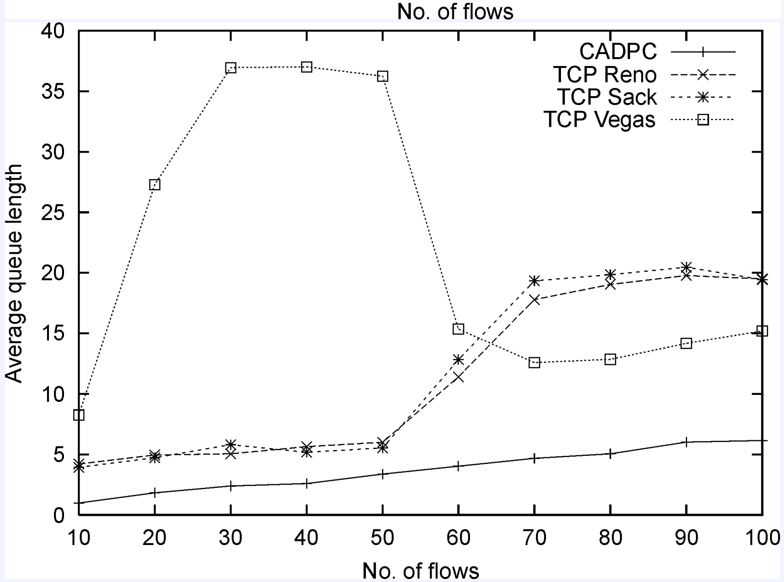


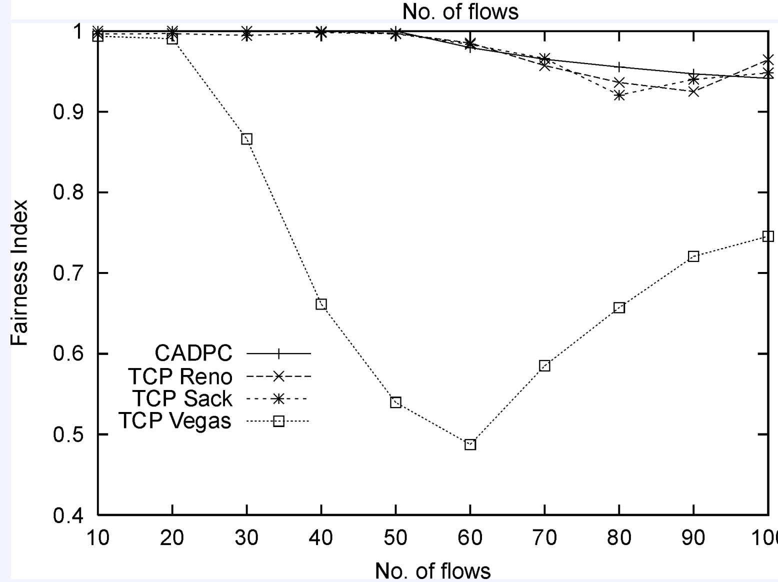



# CADPC vs. 3 TCP(+ECN) flavors



# References

## 1. Congestion Control: a quick introduction


- Raj Jain and K. K. Ramakrishnan, "Congestion Avoidance in Computer Networks with a Connectionless Network Layer: Concepts, Goals and Methodology", Proceedings of Computer Networking Symposium, Washington, D. C., April 11-13 1988, pp. 134-143.
- Van Jacobson, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM 1988, pp. 314-329.
- D. Chiu and R. Jain, "Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks", Journal of Computer Networks and ISDN, Vol. 17, No. 1, June 1989, pp. 1-14.
- Sally Floyd and Van Jacobson, "On Traffic Phase Effects in Packet-Switched Gateways", Internetworking: Research and Experience, V.3 N.3, September 1992, p.115-156. Earlier version: Computer Communication Review, V.21 N.2, April 1991.
- Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993.
- K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

# References /2

## 2. Problems


- Scott Shenker, "Fundamental Design Issues for the Future Internet", IEEE Journal on Selected Areas in Communications, 13, pp. 1141-1149, 1995.
- Frank Kelly, "Charging and rate control for elastic traffic", European Transactions on Telecommunications, 8. pp. 33-37. An updated version is available at http://www.statslab.cam.ac.uk/frank/elastic.html
- Ramesh Johari, "Mathematical Modeling and Control of Internet Congestion", SIAM News, Vol. 33, No. 2.
- L. Massoulié and J. Roberts, "Bandwidth sharing: objectives and algorithms", Proceedings of IEEE Infocom 1999, New York City, New York, 21.-25. 3. 1999.
- Ramesh Johari and David Tan, "End-to-End Congestion Control for the Internet: Delays and Stability", IEEE/ACM Transactions on Networking 9 (2001) 818-832.
- Ashraf Matrawy and Ioannis Labadaris, "A Survey of Congesiton Control Schemes for Multicast Video Applications", IEEE Communications Survey & Tutorials, Fourth Quarter 2003, Vol. 5, No. 2

# References /3

## 3. Some proposed enhancements


- Mark E. Crovella and Azer Bestavros, "Self-similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Transactions on Networking, Vol. 5, No. 6, December 1997.
- Andras Veres, Zsolt Kenesi, Sandor Molnar, Gabor Vattay, "On the Propagation of Long-range Dependency in the Internet", Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, August 28 - September 1 2000.
- Guanghui He, Yuan Gao, Jennifer C. Hou, Kihong Park, "A Case for Exploiting Self-Similarity of Network Traffic in TCP", 10th IEEE International Conference on Network Protocols (ICNP'02), Paris, France, Nov. 12-15, 2002.
- Jörg Widmer, Robert Denda, and Martin Mauve, "A Survey on TCP-Friendly Congestion Control", IEEE Network Magazine, Special Issue "Control of Best Effort Traffic" Vol. 15, No. 3, May 2001.
- Philippe Oechslin and Jon Crowcroft, "Differentiated End-to-End Internet Services using a Weighted Proportional Fair Sharing TCP", ACM Computer Communication Review (CCR), 1998.
- Hari Balakrishnan, Hariharan Rahul, and Srinivasan Seshan, "An Integrated Congestion Management Architecture for Internet Hosts", Proceedings of ACM SIGCOMM 1999, Cambridge, MA., September 1999.

# References /4


- H. Balakrishnan and S. Seshan, "The Congestion Manager", RFC 3124, June 2001.
- Eddie Kohler, Mark Handley, Sally Floyd, and Jitendra Padhye, "Datagram Congestion Control Protocol (DCCP)", Internet-draft (work in progress) draft-ietf-dccp-spec-05.txt, October 2003.
- Metz, C., "TCP Over Satellite... The Final Frontier.", IEEE Internet Computing, 1999.
- Sally Floyd, "HighSpeed TCP for Large Congestion Windows", RFC 3649, Experimental, December 2003.
- Balakrishnan, H., Padmanabhan, V. N., Seshan, S. and Katz, R. H., "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", Proceedings of ACM SIGCOMM 1996, Stanford, CA.
- Ramakrishnan, K., Floyd, S. and Black, D., "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168.
- Dina Katabi, Mark Handley, and Charlie Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks", Proceedings of ACM SIGCOMM 2002, Pittsburgh, PA, 19-23 August 2002.
- Cheng Jin, David X. Wei and Steven H. Low, "FAST TCP: motivation, architecture, algorithms, performance", IEEE Infocom, March 2004.

# References /5

## 4. How to design your own mechanism


- Michael Welzl, "Vector Representations for the Analysis and Design of Distributed Controls", Proceedings of MIC 2002 (IASTED Modelling, Identification and Control Conference), Innsbruck, Austria, 18-22 February 2002.
- Michael Welzl, Max Mühlhäuser: "CAVT - A Congestion Avoidance Visualization Tool", ACM Computer Communication Review, Volume 33, Issue 3, July 2003.
- Michael Welzl: "Scalable Performance Signalling and Congestion Avoidance", Kluwer Academic Publishers, August 2003. ISBN 1-4020-7570-7. Forewords by Jon Crowcroft and Max Mühlhäuser.


**CAVT is available from: http://www.welzl.at/tools/cavt/**

# Thank you!