Internet Technology

Experimental Transport Enhancements

Michael Welzl http://www.welzl.at

DPS NSG Team http://dps.uibk.ac.at/nsg
Institute of Computer Science
University of Innsbruck, Austria

Research issues

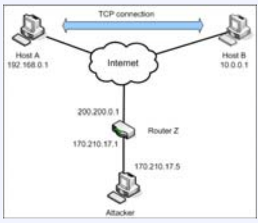
- Main goal: make the Internet faster
- TCP problems are well known + efficient data transfer = important goal
  - Thus, better-than TCP protocols are a popular research topic (mainly congestion control enhancements)
  - ...and so are AQM mechanisms that make TCP work better
- What are the problems?
  - Stability, fairness and security
  - Various performance limitations, e.g. with "long fat pipes", wireless links, mobile environments / highly dynamic routing - multipath transfer,..
  - No multicast support
  - Nevertheless, hard to implement
  - ...just outdated?
    But how do we replace it, given the TCP-friendliness requirement?

Current IETF concern: TCP security

- Historic viewpoint: can an attacker blindly disturb a TCP connection?
  - Hardly: would have to know 4-tuple (src/dst addr, src/dst port and seqno)
  - Thus, no countermeasures in TCP
- Assumption no longer correct!
  [Paul Watson: "Slipping in the Window" (cansecwest/core04 conference)]
  - Window size larger for high speed links (RFC 1323) ⇒ larger number of working seqnos
  - Some applications use long lived connections; e.g. H.323, BGP (major concern!)
    ⇒ longer time available for attacker
  - Also, such long lived connections may have predictable IP addresses / ports
    ⇒ better chances of guessing correct 4-tuple
  - RST attack
    - cause connection to be torn down; works because any RST in current window accepted
    - Mitigation: only accept RST with next expected seqno
  - SYN attack
    - in old spec, SYN with acceptable seqno is answered with RST
    - Mitigation: answer with ACK, which is answered with RST (where new rule applies)
  - DATA attack
    - can lead to "ACK war" (sender / receiver negotiation fails) or corruption
    - Mitigation: always check range of ACK

TCP security /2

- Note: BGP problem long known; awareness issue!
  - RFC 2385 (Proposed Standard, 1998) specifies a MD5 message digest for TCP
  - IPSec authentication can also solve the problem
  - So can authentication based on Timestamps option
- Recent discussion: what about ICMP?
  - Messages can indicate reachability problems, but also source quench and MTU (still beneficial for convergence with new PMTUD, but a security problem)
  - Many pro's and con's to ICMP processing
  - Consider figure: should router Z accept ICMP packets from 170.210.17.1 which tell Host A that Host B is unreachable?



Source: http://www.gont.com.ar/papers/icmp-errors/

Some reasons for TCP CC. 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 !

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
  - Advantage: easy to understand
  - Disadvantage: often requires knowledge of flows in routers (switches) - scalability problem
- Theory: Proportional fairness
  - Network should solve a global optimization problem (maximize log utility function)
  - Advantage: "perfect" fairness notion (max. revenue for provider)
  - Disadvantage: very hard to attain in practice
- Internet:
  - TCP dominant, but does not satisfy max-min-fairness / proportional fairness criteria
  - Therefore, Internet definition of fairness: TCP-friendliness

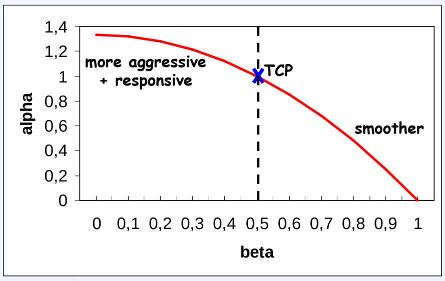
    "A flow is TCP-compatible (TCP-friendly) if, in steady state, it uses no more bandwidth than a conformant TCP running under comparable conditions."

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. α , multiplicative decr. β)
  - $0 < \alpha$, $0 < \beta < 1$ -> stable and fair congestion control
  - $\alpha = 4 \times (1 - \beta\text{^}2) / 3$ -> 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 α and β for TCP-friendliness:



Equation based congestion control

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

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

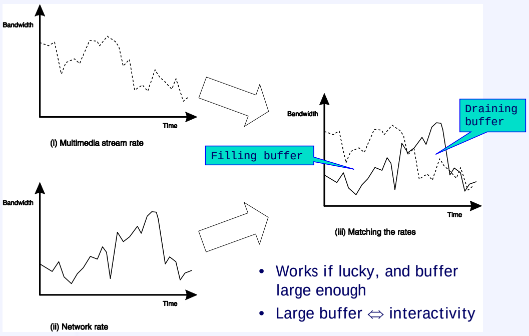
s: packet size
R: rtt
t_{RTO}: TCP retransmit timeout
p: steady-state loss event rate (the difficult part!)

- well known example: **TFRC** - TCP-friendly rate control protocol
  - smooth sending rate
  - IETF status: specified in separate RFC, embedded in DCCP; RTCP-based specification and small-packet variant (for VoIP) in the works

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

Mapping stream and network rates



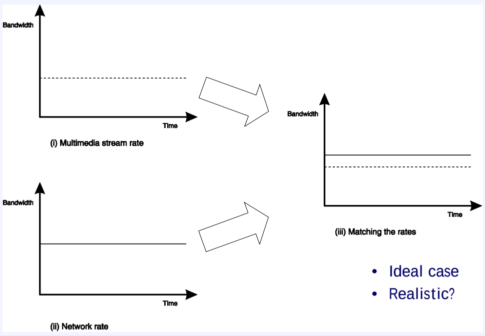
(i) Multimedia stream rate

(ii) Network rate

(iii) Matching the rates

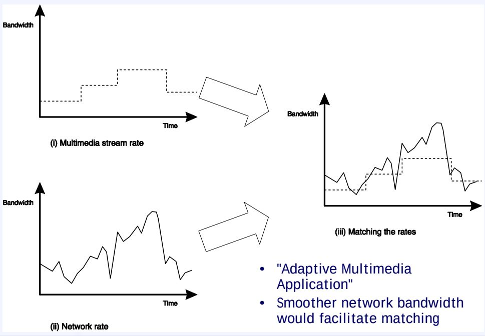
- Works if lucky, and buffer large enough
- Large buffer ⇔ interactivity

Mapping stream and network rates /2



- Ideal case
- Realistic?

Mapping stream and network rates /3



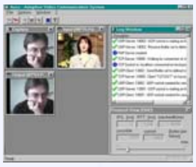
- "Adaptive Multimedia Application"
- Smoother network bandwidth would facilitate matching

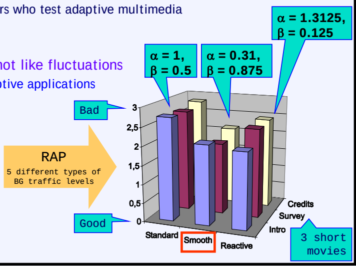
Multimedia adaptation

- Common mistake:
  - adaptation schemes often assume arbitrary data stream scalability
- Problems:
  - Data streams show fluctuations (example: MPEG I-, B-, P-frames)
  - 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
- Common solution:
  - Special CBR design for communication - H.261 designed for ISDN
  - Note: not always feasible
  - In any case, a smooth yet TCP-friendly rate is desirable

Adaptive multimedia: the user experience

- Studied by several research groups
  - Automatically evaluate "user experience" by judging received content based on knowledge about users
  - Study heartbeat etc. of users who test adaptive multimedia
  - Surveys
- Consistent result: users do not like fluctuations
  - Decline of research on adaptive applications





Issues with TCP-friendliness

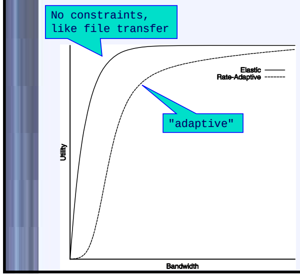
- TCP regularly increases the queue length and causes loss
  ⇒ detect congestion when it is already (ECN: almost) too late!
  - possible to have more throughput with smaller queues and less loss
    ... but: exceed rate of TCP under similar conditions ⇒ not TCP-friendly!
- What if I send more than TCP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

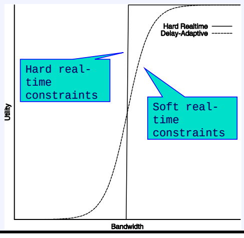
Does TCP-friendliness hinder research?

- 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"?

Characterizing multimedia applications

- Different utility functions
  - not necessarily logarithmic (cf. VoIP) ⇒ proportional fairness not ideal!





Control *what*? Traffic jams, huh?

- Nowadays, networks are often overprovisioned
  ⇒ no traffic jams; no congestion
  - often, but not always (e.g. wireless links)
  - this situation may change (access vs. core bandwidth changes)
- Networks are underutilized...
  exactly, that's the issue!
- Essentially, the problem changed from

  "*how do we get rid of all this congestion*"

  to

  "*how do we efficiently use all this spare bandwidth*"

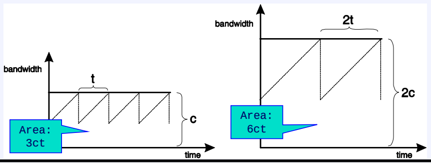
TCP with High Speed links

- TCP over "long fat pipes": large bandwidth*delay product
  - long time to reach equilibrium, MD = problematic!
  - From RFC 3649 (HighSpeed RFC, Experimental):

```
For example, for a Standard TCP connection with 1500-byte packets and a 100 ms
round-trip time, achieving a steady-state throughput of 10 Gbps would require
an average congestion window of 83,333 segments, and a packet drop rate of at
most one congestion event every 5,000,000,000 packets (or equivalently, at most
one congestion event every 1 2/3 hours). This is widely acknowledged as an
unrealistic constraint.
```

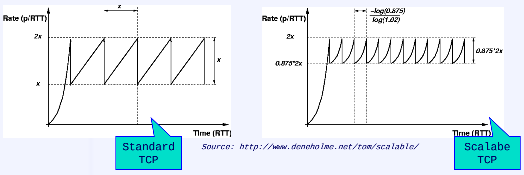
Theoretically, utilization independent of capacity

But: longer convergence time



Proposed solutions

- Standards: larger initial window / window scaling option, TCP SACK
- Scalable TCP: increase/decrease functions changed
  - cwnd := cwnd + 0.01 for each ack received while not in loss recovery
  - cwnd := 0.875 * cwnd on each loss event
    (probing times proportional to rtt but not rate)



Source: http://www.deneholme.net/tom/scalable/

Proposed solutions /2

Rate	Standard TCP recovery time	Scalable TCP recovery time
1Mbps	1.7s	2.7s
10Mbps	17s	2.7s
100Mbps	2mins	2.7s
1Gbps	28mins	2.7s
10Gbps	4hrs 43mins	2.7s

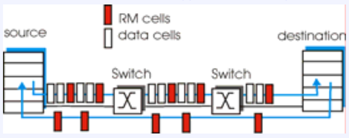
- HighSpeed TCP (RFC 3649 includes Scalable TCP discussion):
  - response function includes a(cwnd) and b(cwnd), which also depend on loss ratio
  - less drastic in high bandwidth environments with little loss *only*
  - **Significant step!**
  - Previously, either TCP-friendly or better-than-TCP; no combinations!
- TCP Westwood+
  - different congestion response function (proportional to rate instead of $\beta = 1/2$)
  - Proven to be stable, tested in real life experiments, available in your Linux

Proposed solutions /3

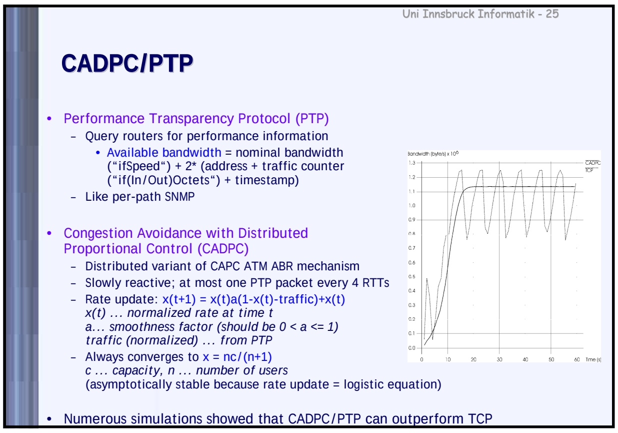
- FAST TCP
  - Variant based on window and delay
  - Delay allows for earlier adaptation (awareness of growing queue)
  - Proven to be stable
  - Commercially announced + patent protected, by Steven Low's CalTech group
  - another delay-based example: TCP Vegas
    - Vegas = impractical because less aggressive than standard TCP
- BIC, CUBIC
  - BIC (Binary InCrease TCP) uses binary search to find the ideal window size:
    - when loss occurs, current window = max, new window = min
    - check midpoint;
    - if no loss ⇒ new min, increase; else new window = new max
  - CUBIC = BIC++ using cubic function; growth does not depend on RTT

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!



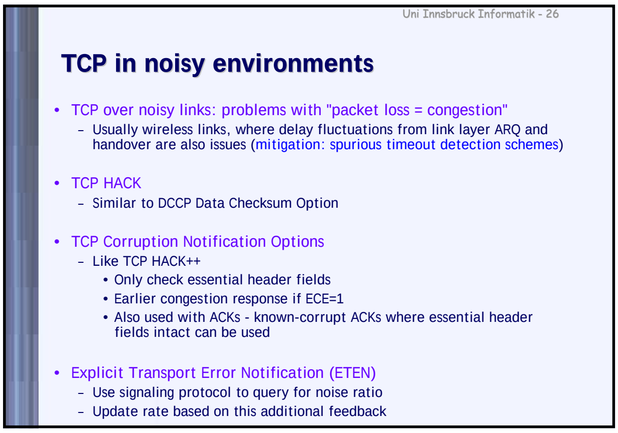
- Experimental Internet approaches:
  - Multilevel ECN (two bits), eXpress Control Protocol (XCP), CADPC/PTP (my own)
  - Quick-Start: query routers for initial sending rate with IP options
    - IETF effort; many discussions: security (nonces again), IP option handling
    - Routers often drop or delay packets with options; thus, suggested for controlled environments only



CADPC/PTP

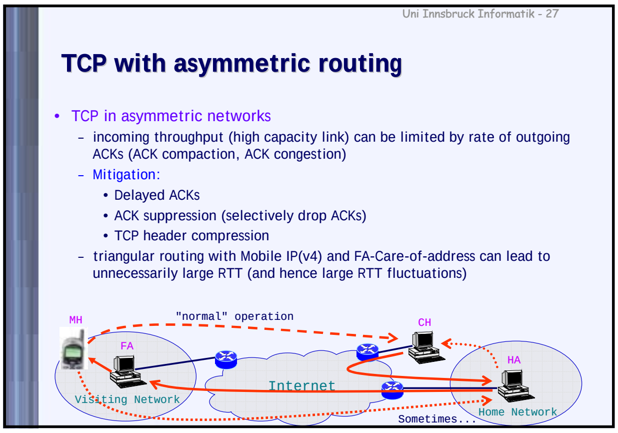
- Performance Transparency Protocol (PTP)
  - Query routers for performance information
    - Available bandwidth = nominal bandwidth (“ifSpeed“) + 2* (address + traffic counter (“if(In/Out)Octets“) + timestamp)
  - Like per-path SNMP
- Congestion Avoidance with Distributed Proportional Control (CADPC)
  - Distributed variant of CAPC ATM ABR mechanism
  - Slowly reactive; at most one PTP packet every 4 RTTs
  - Rate update: x(t+1) = x(t)a(1-x(t)-traffic)+x(t)
    x(t) … normalized rate at time t
    a… smoothness factor (should be 0 < a <= 1)
    traffic (normalized) … from PTP
  - Always converges to x = nc/(n+1)
    c … capacity, n … number of users
    (asymptotically stable because rate update = logistic equation)
- Numerous simulations showed that CADPC/PTP can outperform 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



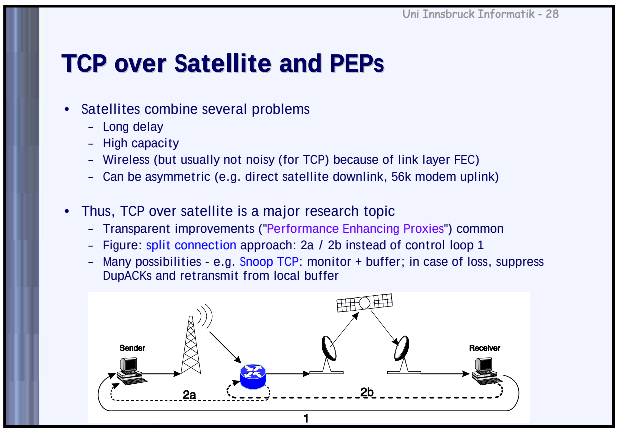
TCP in noisy environments

- TCP over noisy links: problems with "packet loss = congestion"
  - Usually wireless links, where delay fluctuations from link layer ARQ and handover are also issues (mitigation: spurious timeout detection schemes)
- TCP HACK
  - Similar to DCCP Data Checksum Option
- TCP Corruption Notification Options
  - Like TCP HACK++
    - Only check essential header fields
    - Earlier congestion response if ECE=1
    - Also used with ACKs - known-corrupt ACKs where essential header fields intact can be used
- Explicit Transport Error Notification (ETEN)
  - Use signaling protocol to query for noise ratio
  - Update rate based on this additional feedback



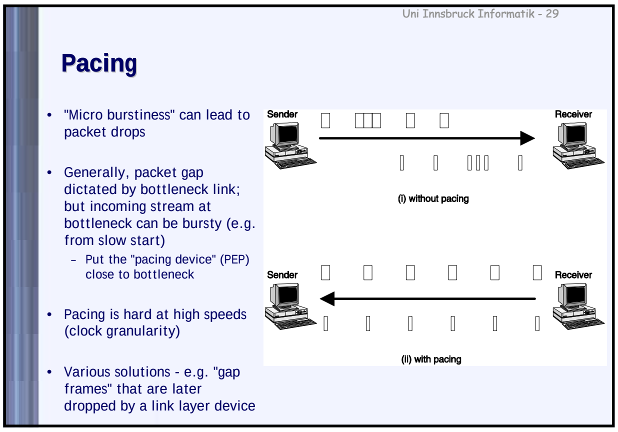
TCP with asymmetric routing

- TCP in asymmetric networks
  - incoming throughput (high capacity link) can be limited by rate of outgoing ACKs (ACK compaction, ACK congestion)
  - Mitigation:
    - Delayed ACKs
    - ACK suppression (selectively drop ACKs)
    - TCP header compression
  - triangular routing with Mobile IP(v4) and FA-Care-of-address can lead to unnecessarily large RTT (and hence large RTT fluctuations)



TCP over Satellite and PEPs

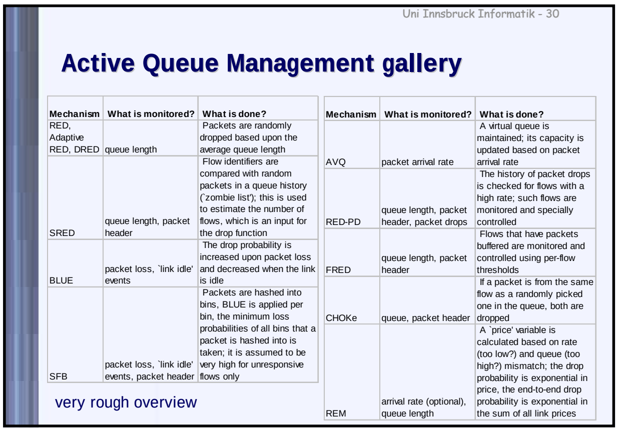
- Satellites combine several problems
  - Long delay
  - High capacity
  - Wireless (but usually not noisy (for TCP) because of link layer FEC)
  - Can be asymmetric (e.g. direct satellite downlink, 56k modem uplink)
- Thus, TCP over satellite is a major research topic
  - Transparent improvements ("Performance Enhancing Proxies") common
  - Figure: split connection approach: 2a / 2b instead of control loop 1
  - Many possibilities - e.g. Snoop TCP: monitor + buffer; in case of loss, suppress DupACKs and retransmit from local buffer



Pacing

- "Micro burstiness" can lead to packet drops
- Generally, packet gap dictated by bottleneck link; but incoming stream at bottleneck can be bursty (e.g. from slow start)
  - Put the "pacing device" (PEP) close to bottleneck
- Pacing is hard at high speeds (clock granularity)
- Various solutions - e.g. "gap frames" that are later dropped by a link layer device

Sender
Receiver
(i) without pacing
Sender
Receiver
(ii) with pacing



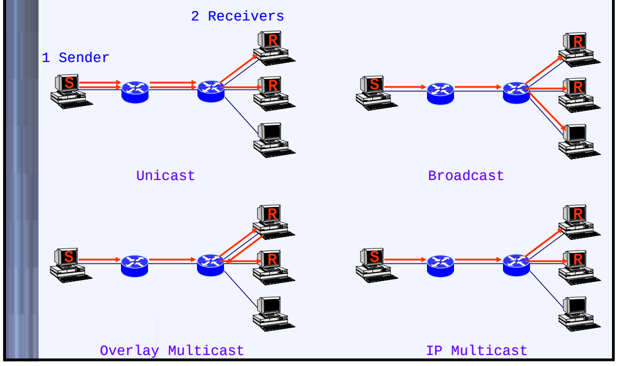
Active Queue Management gallery

Mechanism	What is monitored?	What is done?
RED, Adaptive RED, DRED	queue length	Packets are randomly dropped based upon the average queue length
SRED	queue length, packet header	Flow identifiers are compared with random packets in a queue history (`zombie list'); this is used to estimate the number of flows, which is an input for the drop function
BLUE	packet loss, `link idle' events	The drop probability is increased upon packet loss and decreased when the link is idle
SFB	packet loss, `link idle' events, packet header	Packets are hashed into bins, BLUE is applied per bin, the minimum loss probabilities of all bins that a packet is hashed into is taken; it is assumed to be very high for unresponsive flows only

very rough overview

Mechanism	What is monitored?	What is done?
AVQ	packet arrival rate	A virtual queue is maintained; its capacity is updated based on packet arrival rate
RED-PD	queue length, packet header, packet drops	The history of packet drops is checked for flows with a high rate; such flows are monitored and specially controlled
FRED	queue length, packet header	Flows that have packets buffered are monitored and controlled using per-flow thresholds
CHOKe	queue, packet header	If a packet is from the same flow as a randomly picked one in the queue, both are dropped
REM	arrival rate (optional), queue length	A `price' variable is calculated based on rate (too low?) and queue (too high?) mismatch; the drop probability is exponential in price, the end-to-end drop probability is exponential in the sum of all link prices

Unicast / Broadcast / (overlay) Multicast



Multicast issues

- Required for applications with multiple receivers only
  - video conferences, real-time data stream transmission, ..
    ⇒ 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!

Multicast congestion control proposals

- TCP-friendly Multicast Congestion Control (TFMCC)
  - Rate-based single-rate scheme; multicast variant of TFRC
  - Only the Current Limiting Receiver (CLR) is allowed to send feedback
  - Choice is made automatically by moving rate calculation to the receiver and only allowing feedback if *calculated rate < sender rate*
- Pragmatic General Multicast Congestion Control (pgmcc)
  - Window-based single-rate scheme
  - Co-designed with PGM protocol, which uses NACKS and has features such as FEC, aggregation of NACKs in PGM-capable routers, ..
  - Representative receiver ("ACKer") is chosen; sends ACK in addition to NACKs
  - Emulates TCP behavior
- Receiver-driven Layered Multicast (RLM)
  - Rate-based layered scheme
  - Sender transmits each layer in separate multicast group
  - Receivers periodically probe bandwidth by joining groups ("join-experiment")
- Many other proposals: RLC, MLDA, PLM, FLID-DL, WEBRC,
  ... but Internet deployment questionable

Reality check: high speed TCPs

- After major press release (Slashdot: "BIC-TCP 6000 times quicker than DSL"), BIC became default TCP CC. in Linux in mid-2004
  - Now replaced with CUBIC
- Compound-TCP (CTCP) = default TCP CC. in Windows Vista Beta
  - For testing purposes; disabled by default in standard release
- How will these protocols interact?
  - Standards desirable
- Process devised: proposals will be pre-evaluated by
  IRTF Internet Congestion Control Research Group (ICCRG)
  - Evaluation guidelines: RFC 5033, Transport Models Research Group (TMRG)
  - CTCP and CUBIC proposals currently on the table (October 2007)
  - See: http://www.irtf.org/charter?gtype=rg&group=iccrg for more details

References

- Michael Welzl, "Network Congestion Control: Managing Internet Traffic", John Wiley & Sons, Ltd., August 2005, ISBN: 047002528X
- M. Hassan and R. Jain, "High Performance TCP/IP Networking: Concepts, Issues, and Solutions", Prentice-Hall, 2003, ISBN: 0130646342
- M. Duke, R. Braden, W. Eddy, E. Blanton: "A Roadmap for TCP Specification Documents", Internet-draft draft-ietf-tcpm-tcp-roadmap-06.txt, http://www.ietf.org/internet-drafts/draft-ietf-tcpm-tcp-roadmap-06.txt (in RFC Editor Queue)
- Eric He (editor), Pascale Vicat-Blanc Primet (editor), Michael Welzl (editor), Mathieu Goutelle, Yunhong Gu, Sanjay Hegde, Rajikumar Kettimuthu, Jason Leigh, Chaoyue Xiong, Muhammad Murtaza Yousaf, "A Survey of Transport Protocols other than Standard TCP", Global Grid Forum Document GFD.55, Data Transport Research Group, 23 November 2005.
- IETF TCPM WG: http://www.ietf.org/html.charters/tcpm-charter.html