Internet congestion control and resource allocation

IERG 5090

Network resource allocation

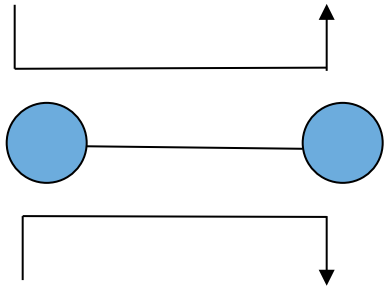
- If network capacity (bandwidth) is larger than all the user load added together, resource allocation is trivial.
- Otherwise, need to decide who to serve first, or how to “allocate” your attention to different users.

- How to model “user load” in order to do some analysis?

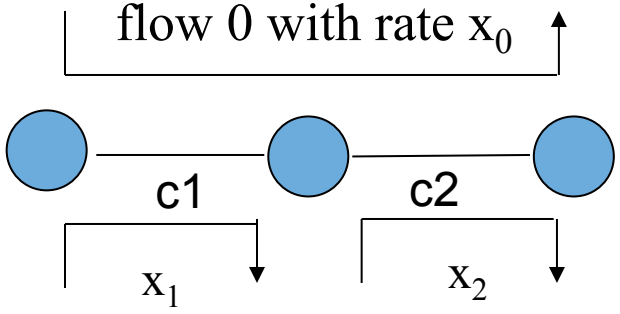
Elastic flows as demand model

- Each user is a flow (from certain source to certain destination), with an infinite supply of packets.
  - No matter what capacity the network has, users can use more
  - The higher the throughput the better, but the desire for higher throughput diminishes as higher throughput is reached
- How would you allocate network resources to serve this demand?

Example 1:



Example 2:



Fairness (equality)

- In various situations of resource allocation, the desire to give fair treatment to users, when other things (e.g. efficiency) being equal

- If there are n users, the allocation is x_1, x_2, …, x_n, we want a function to measure how equal (fair) it is.

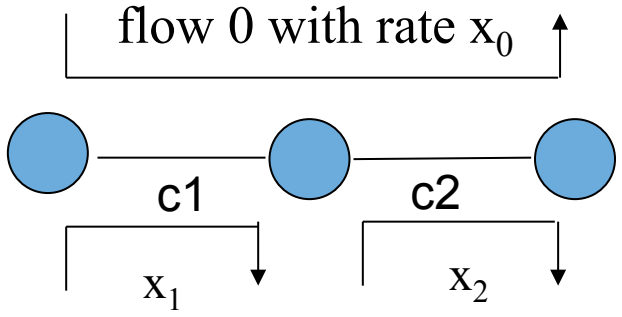
- Gini index: https://en.wikipedia.org/wiki/Gini_coefficient
- Jain-chiu-hawe fairness index: https://scholar.google.com/citations?user=6foKsAIAAAAJ&hl=en&oi=ao

$$\mathcal{J}(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2}$$

But these fairness indices are both for the situation with multiple demands for same resource, i.e. example 1.

Fairness for different demands

- Consider example 2
- It may not be possible (or reasonable) to assign same rate to each flow
  - Consider different examples of c1 and c2



- We can define the problem as a constrained optimization problem:

Max F(x0, x1, x2)

Subject to:

x0 + x1 <= C1

x0 + x2 <= C2

What function F() should we optimize? Are the fairness functions shown before applicable?

Max-min Fairness

- Given a network, and a set of flows each traversing a set of links of the network
- An **allocation** is an assignment of rates to the flows
- An allocation is **feasible**, if the sum of allocation traversing each link is less than that link's capacity
- A link is **saturated** if the sum of allocation traversing that link equals to the **capacity of the link**
- An allocation is **max-min fair**, if increasing the rate of any flow will cause the smallest flow at some link to reduce rate in order to maintain feasibility

- Why use Max-min fairness?
  - It is intuitively reasonable
  - It is roughly implementable, by centralized or distributed algorithms
  - But there is no theoretical justification this is the best objective to aim at

Water-filling algorithm

This algorithm is a constructive way of defining Max-min fairness:

- Start with all flows with rate equal to zero
- Grow all flow rates at the same pace, until some rate(s) become limited by their bottleneck(s)
- Leave these rates alone, and grow the rest of the rates (that have not been limited by their bottleneck), till all rates become limited

It can be readily implemented:

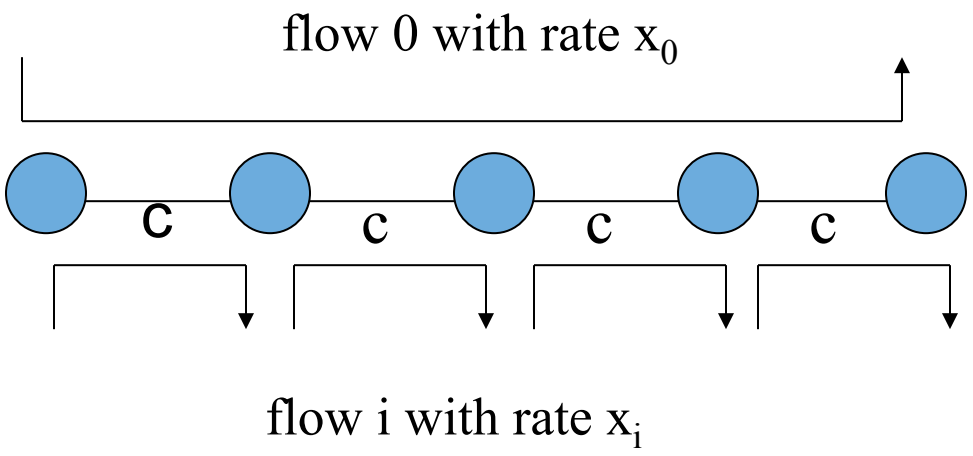
- In a central controller (e.g. in SDN)
- By distributed congestion control (e.g. TCP)

Example:

- There are four links, 1, 2, 3, and 4, each with unit capacity
- There are five flows, a, b, c d and e:
  - a traverses 1, 2
  - b traverses 2, 3, 4
  - c traverses 3, 4
  - d traverses 4
  - e traverses 1
- Max-min fair solution:
  - Flows b, c, d saturate first at 1/3 due to link 4
  - Flow a, e saturate at ½ due to link 1

The tradeoff between efficiency and fairness

- In this example, max-min fair allocation gives each flow c/2
  - Hence total throughput is 5c/2
- We could achieve total throughput 4c, by allocating 0 to flow 0.
  - More throughput, but less fair
  - What to do?
- Assume each user has a utility for given rate
  - Then maximize the total utility



Convex optimization – proportional fairness

- Assume each flow as a utility function
  U(x) = the happiness for getting allocation x
- The "best" bandwidth allocation is obtained by
  Maximize sum($U(x_i)$) for all feasible x
  This is a **constrained optimization problem**

A reasonable utility function is U(x) = log(x)

The resulting allocation is called **proportional fairness**

Work by Frank Kelly and his students in late 1990s:

Rate control in communication networks: shadow prices, proportional fairness and stability
F. P. Kelly, A.K. Maulloo and D.K.H. Tan. Journal of the Operational Research Society 49 (1998), 237-252.

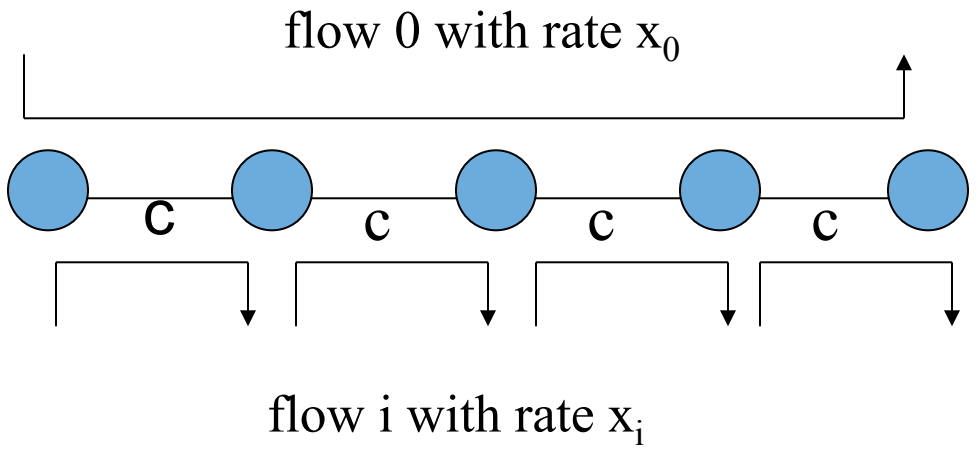
Example for proportional fairness

Proportional fair allocation gives:

$x_0 = c/(n+1)$

$x_i = nc/(n+1)$

Where n = 4 here



Maximize sum of utility means

sum $U(x) = \log(x_0) + \log(x_1) + \log(x_2) + \log(x_3) + \log(x_4)$

Since $x_i = c - x_0$

sum $U(x) = \log(x_0) + 4\log(c-x_0)$

Setting the gradient to 0 gives

$x_0 = c/(n+1)$

Implication for congestion pricing

- Assume each link is a seller, and it asks all flows to bid for allocation of its capacity
- Assume each flow is a buyer, who tries to maximize its utility (for its given allocation), minus what it has to pay
- It can be shown that the **optimal allocation** for the constrained network optimization problem is the allocation reached in **market equilibrium**

- This leads many to study network pricing as a way to do **"optimal" congestion control**

- But should network resources be traded like a commodity? Or should it be like air, water and electricity, made available to all?

Summary of network resource allocation

- A simplified view is that all network user demand are **elastic flows**
- Given a set of elastic flows, network resource allocation to best serve these flows becomes a well-defined optimization problem
- Depending on the chosen objective function, the outcome can be:
  - Max-min fairness, proportional fairness, or other solutions

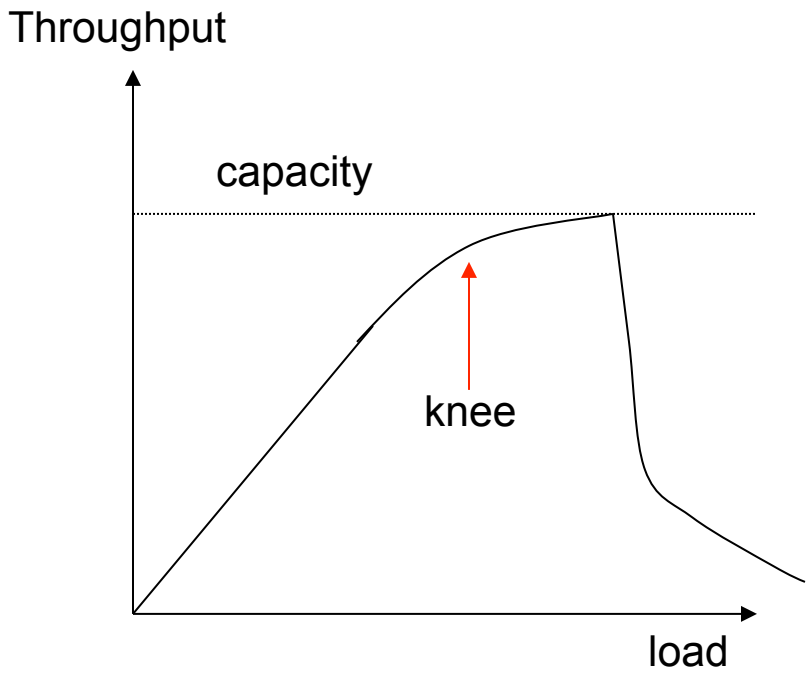
- Given this understanding, how do we realize the chosen resource allocation policy?
- In reality, network is complicated with different stakeholders, such as different ISPs, application service providers, users, how does this affect policies?
- How should resource allocation policy be agreed by different stakeholders? Does government need to regulate?

Network congestion

- Network congestion can be avoided:
  - If network is a telephone-network style circuit switching network, each connection comes with fixed bandwidth
  - Admission control was used to control load
- By design, Internet tries to provide “packet-switching”, best effort service, and congestion inevitably arise
- In original Internet, congestion control was based on load-dependent routing; which was abandoned due to oscillations
- Congestion control was added to TCP protocol in 1980s
  - Original TCP only deals with flow control, error control and connection state control

Congestion collapse

- As best effort service reaches service capacity, service level slows, due to queueing
- But at some point, too much load can bring service level down sharply
- Possible reasons:
  - More congestion -> more packet losses, more wasted usage -> less and less real work done
  - Similar to gridlock in road traffic

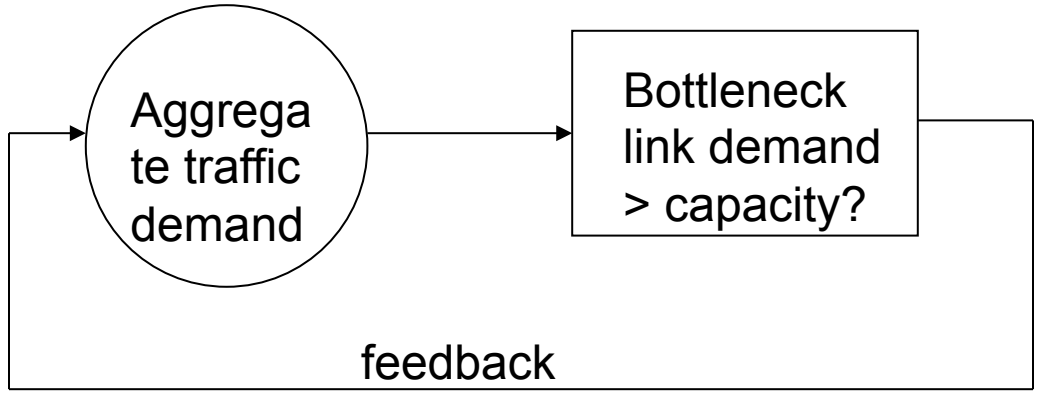


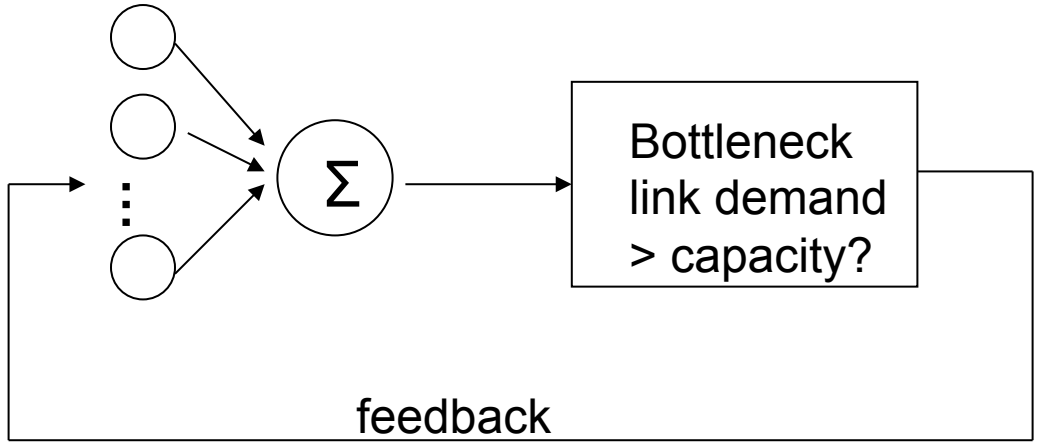
Congestion control

For a control problem, relevant issues are:

- What is the form of feedback
- Is it stable – or how close can you get demand = capacity

Given a multi-user control problem, what is the resource allocation result? Is it fair? Does it reach the desired allocation whatever the initial condition is?





Congestion feedback

- Original design was explicit feedback
  - Using “Source Quench” message from router to source
- Other ways of in-band signaling:
  - DECbit: (from DECnet); later became ECN (Explicit Congestion Notification)
  - Delay as feedback
  - Packet drop: detected by:
    - multiple “duplicate ACKs”
    - No ACK after several round trip times

Packet drop approach adopted by TCP

Considerations:

- Overhead – too much overhead contribute to congestion
- Delay – more delay leads to more oscillation and instability
- Accuracy of information – e.g. packet drop can be caused by other factors than congestion, for example due to error
- Sensitivity, robustness, etc

Decentralized algorithm - AIMD

- Given some simplifying assumptions:
  - Binary feedback
  - Single bottleneck
  - All senders get feedback synchronously
- Is there a decentralized congestion control that gives fair bandwidth allocation?
  - Additive Increase Multiplicative Decrease (AIMD)

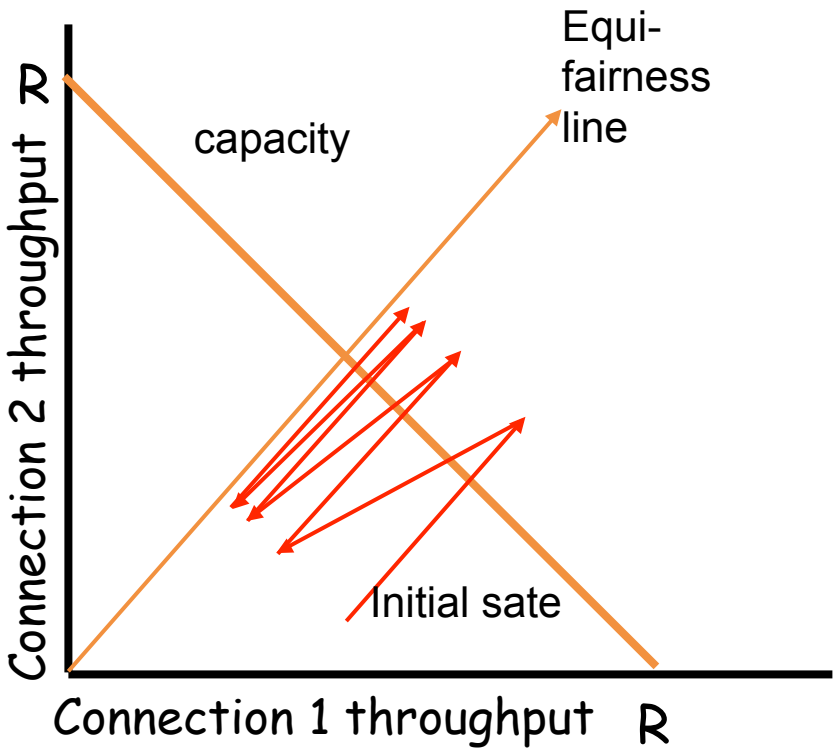
The AIMD paper was published in a lessor known journal:

DM Chiu and R Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", Computer Networks and ISDN Systems Vol 17, pp 1-14, 1989

It is still receiving a lot of attention today:

https://scholar.google.com.hk/citations?user=6foKsAIAAAAJ&hl=zh-TW&oi=ao

Two user example to illustrate AIMD



- In order to converge to efficiency and fairness, conclusions are:
- Decrease should be multiplicative
- Increase should be additive (possibly with multiplicative component)
- These are sufficient conditions, not necessary

TCP Congestion Control

- Detects packet loss
  - based on (a) duplicate ACKs of same packet; (b) no ACK after Time-Out period
  - Packet loss = congestion feedback
- Number of outstanding packets is controlled by a Window
  - Initially Window = 1
  - Additively increase w/o congestion
  - Upon congestion, cut to half
  - i.e. implementing AIMD
- The congestion control standard passed IETF
  - V Jacobson and S Floyd did implementation and lots of simulation experiments, using Network Simulator (NS)
  - The theoretical foundation comes from AIMD and Fair Resource Allocation model
- TCP congestion control deployed in 1980s and 1990s, used till now
- There have been lots of follow-up work

Well-known problems with TCP congestion control

1. Like in any control mechanism, if the feedback delay is large, the result is less stable – meaning there will be oscillations
2. In wireless networks, packet drop may be due to bit errors, rather than congestion – need better congestion indication
3. Most multimedia applications do not need TCP-kind of reliability, and needs steady rate – elastic flow assumption does not hold
4. In today's networks, increasingly some traffic is more urgent, or is considered "worth more than others" – elastic flows assumption no longer hold
   - In data centers
   - Different services in the Internet

High Bandwidth Delay Product networks

- High bandwidth
  - Optical links (gigabits)
  - Even wireless can be high bandwidth
- Large delay
  - Satellite links
  - Wide area network
  - Wireless
- When the product of these two is large, TCP becomes oscillatory:
  - Reacts too much to congestion (drop window by half)
  - Takes many round trips to build back

- Fairness also becomes a problem:
  - Short flows can react faster than flows with large RTTs (e.g. satellite), the latter suffers more

This is an active research problem in 2000s:

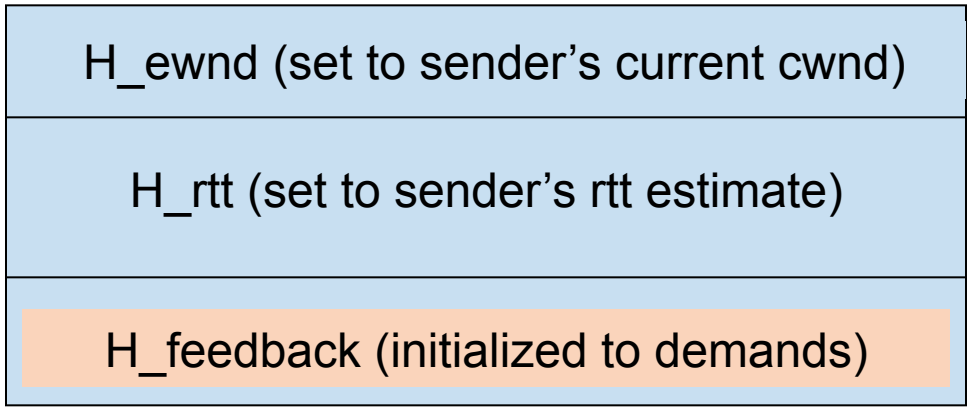
- One of the more elegant solutions (2002): XCP (explicit control protocol)

Ideas behind XCP

- Efficiency of a link involves only the aggregate traffic's behavior
- Fairness is the relative allocation to flows sharing a link
- Okay to have efficiency but not fairness, or different fairness; but dealt with together by AIMD

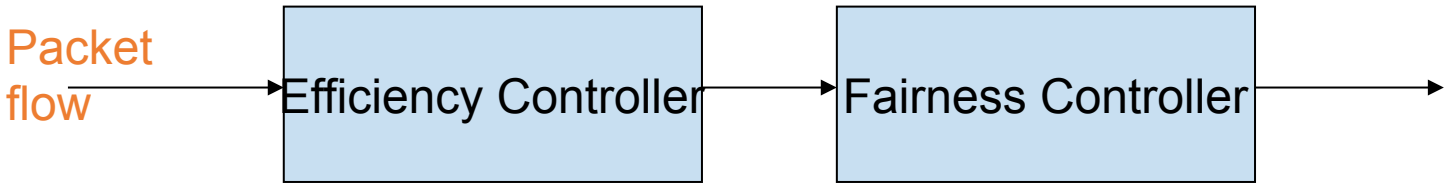
Hence:

- Separate efficiency and fairness controls
- Have more explicit feedback (extending ECN's idea)



- H_cwnd – sender's current cong. Window
- H_rtt – sender's current RTT estimate
- H_feedback – initialized by sender; modified by routers along path to **directly control the congestion windows**

Where are XCP implemented

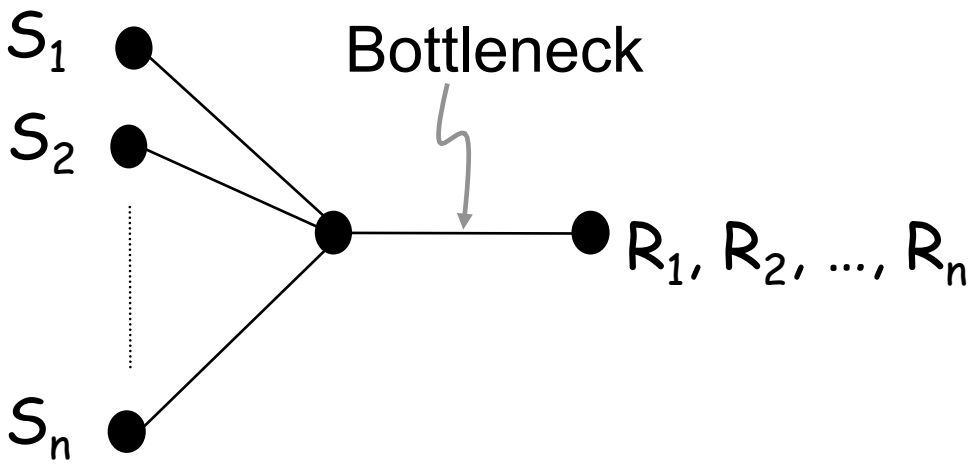


XCP Router:

- Both together compute the explicit feedback information
- Efficiency feedback takes into account of stability
- Fairness controller is based on AIMD

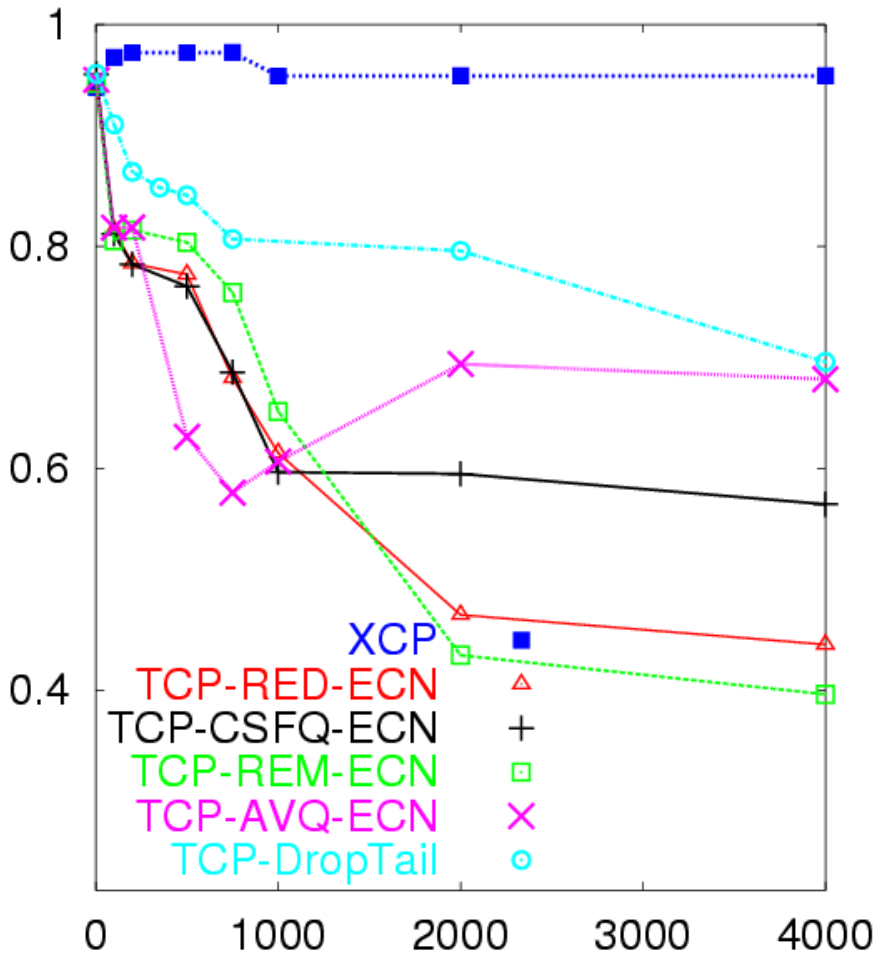
- Sender computes rate based on feedback info
- Receiver needs only to copy header info into ACK messages

Simulation study



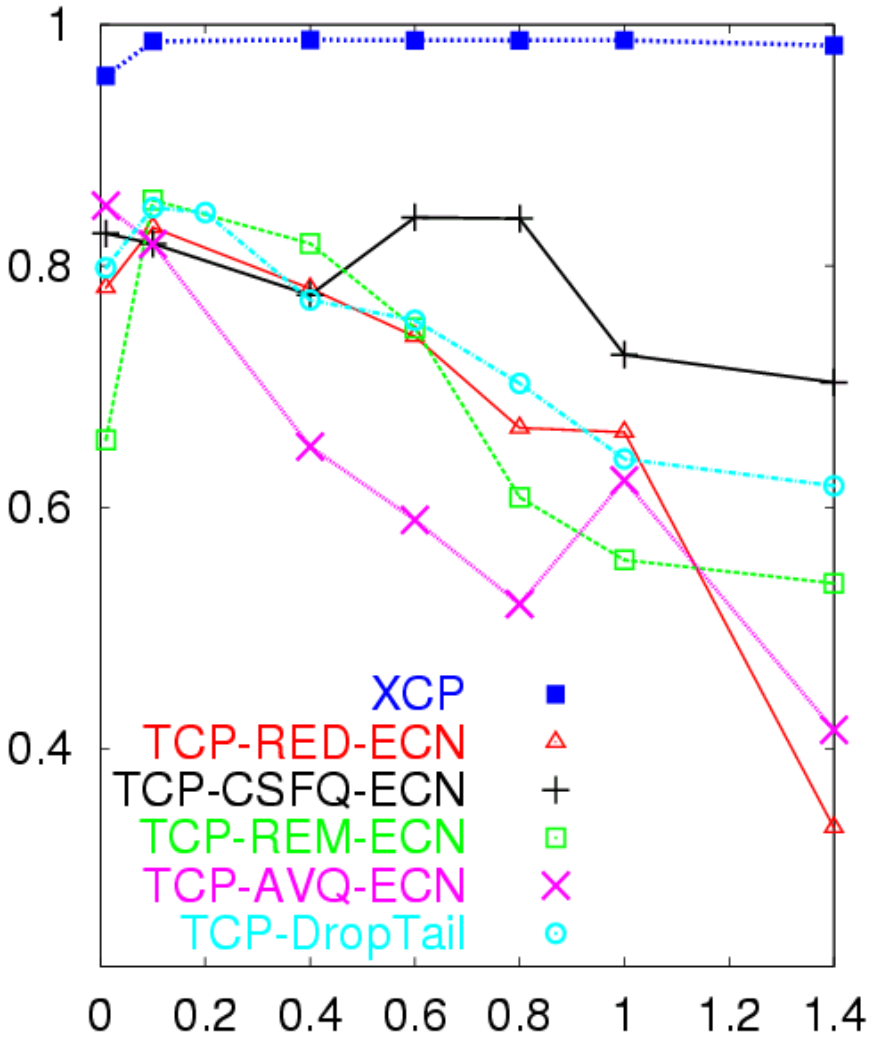
Utilization versus Bandwidth

- 50 long-lived TCP flows
- 80ms Prop. Delay
- 50 flows in reverse direction to create 2-way traffic
- *XCP is near optimal!*



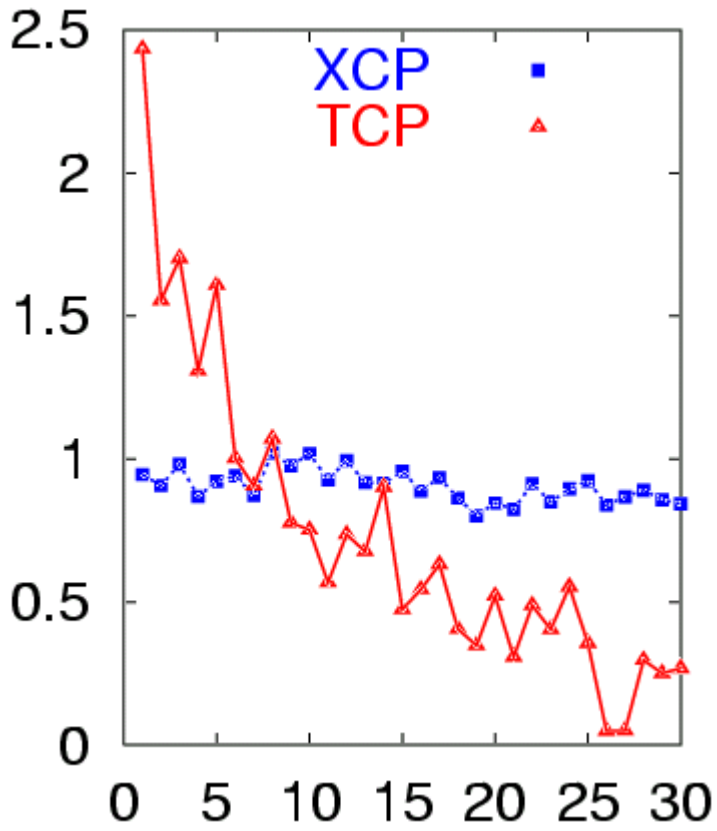
Utilization vs Delay

- 50 long-lived TCP flows
- 150 Mb/s Capacity
- 50 flows in reverse direction to create 2-way traffic
- XCP wins again by adjusting it's aggressiveness to round trip delay



Fairness

- 30 long-lived FTP flows
- Single 30 Mb/s bottleneck
- Flows are increasing in RTT from 40-330 ms
- To the left is Throughput vs. flow.



Summary of XCP

- Understand the problem with TCP for high BDP networks
- XCP is a nice design
  - Generalization of ECN
  - Decouple efficiency and fairness
  - But it requires changes to sender, receiver and routers!
  - It is not deployed in the Internet

D Katabi, M Handley and C Rohrs, “Congestion Control for High Bandwidth-Delay Product Networks”, ACM Sigcomm 2002.

TCP in wireless networks

- A lot of research went into this topic too, not much went into action:
  - Use a proxy server to handle connection over wireless
  - Try to distinguish congestion packet loss vs error loss; we did some work, called TCP Veno
  - Multi-path TCP, a variant of TCP used in wireless especially

Multi-path congestion control

- Multi-path congestion control is another nice extension to TCP congestion control
  - Requires network layer to provide multiple paths to support single flow
  - Can achieve similar effect as load dependent routing – shift traffic from congested path to less congested paths
  - Challenge is to keep it fair to single-path flows

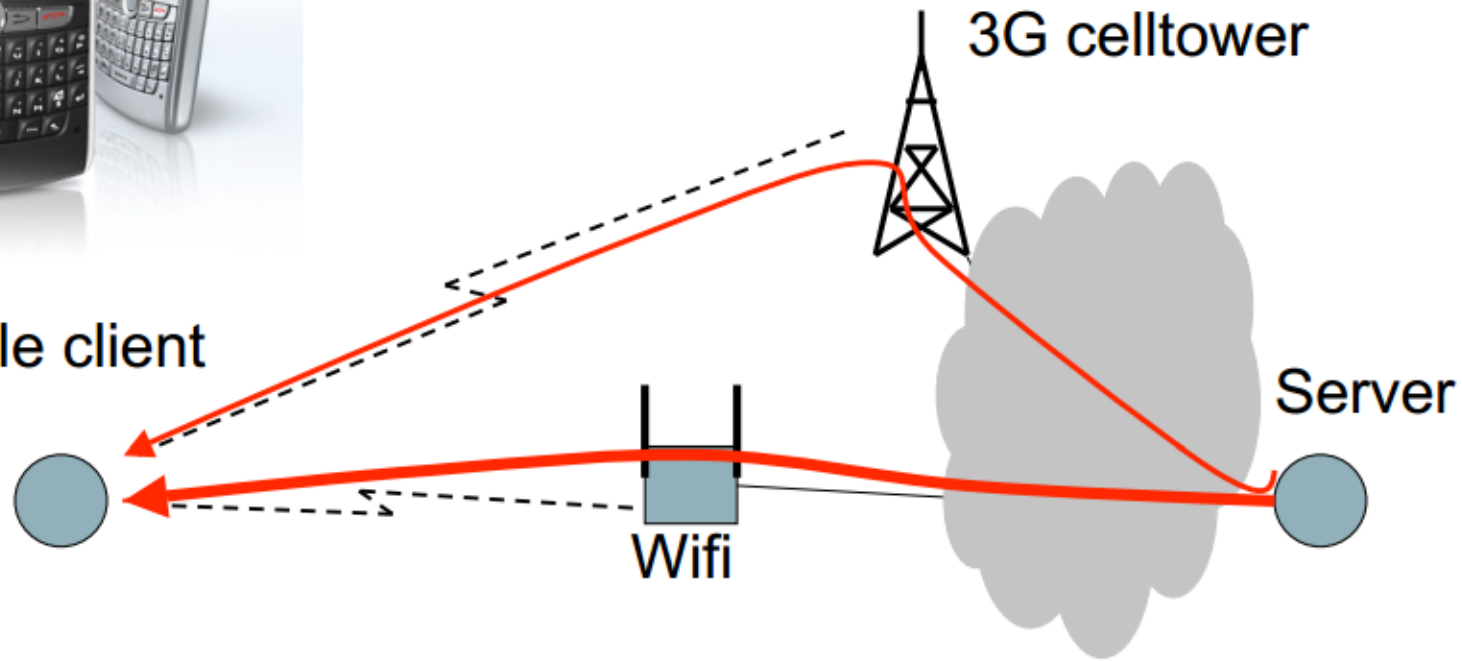
There are some implementations for multi-path TCP

In July 2013, the Multipath TCP working group reported four independent implementations of Multipath TCP:

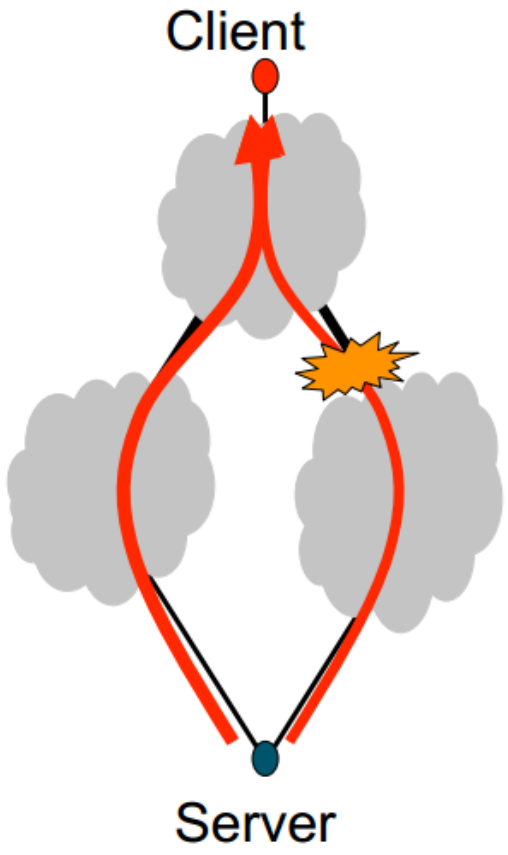
- Linux Kernel (reference implementation) from Université Catholique de Louvain.
- FreeBSD (IPv4 only) from Swinburne University of Technology.
- Citrix Netscaler.
- Apple iOS 7, released on September 18, 2013 is the first large scale commercial deployment of Multipath TCP.

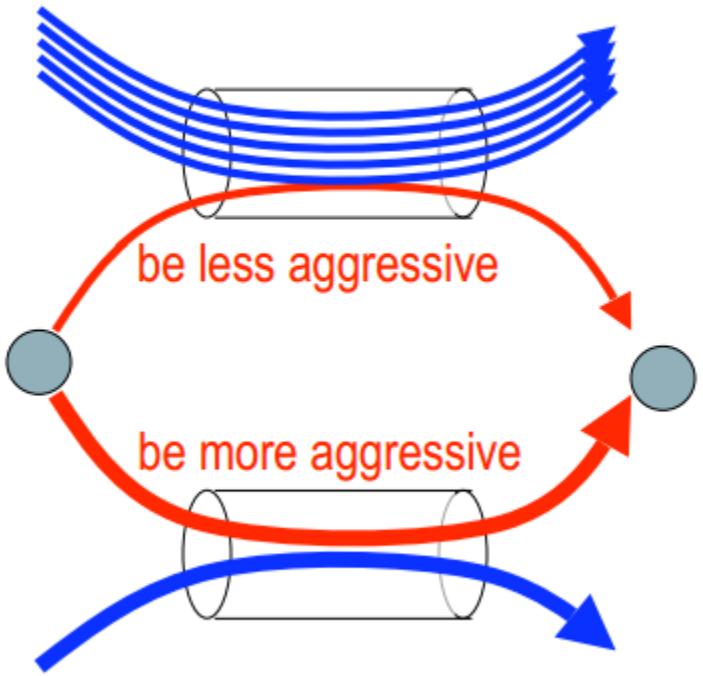
Scenario of using MP TCP

- Smartphone



Benefit of MP TCP





Sending simultaneously across more than one path can provide:

- robustness,
- balance load
- pool resources.

Summary of MP TCP

- It is not hard to make multi-path TCP friendly with regular TCP
  - Each path increase its rate proportionally with its fraction of total rate of flow
- But oscillation between different paths noticed
  - This problem is nicely addressed by the papers in the references
- Multi-path TCP is used first in wireless because wireless access devices already need to deal with multi-paths (e.g. WiFi and Cellular link)

References:

- Practical Congestion Control for Multipath Transport Protocols, Costin Raiciu, Damon Wischik, Mark Handley, UCL Technical Report
- Wikipedia: Multipath TCP http://en.wikipedia.org/wiki/Multipath_TCP
- Multipath TCP Resources http://nrg.cs.ucl.ac.uk/mptcp/
- http://perso.uclouvain.be/olivier.bonaventure/blog/html/2013/09/18/mptcp.html

Support for Multimedia applications

- Implement multimedia applications over UDP instead of TCP
  - Avoid TCP's reliability and CC
  - But UDP traffic often blocked by firewalls, since UDP can be used for DoS attacks
- TCP-friendly congestion control
  - Once a popular topic of CC research
  - Does not help much
- For Video streaming
  - Use HTTP bitrate adaptation protocols (e.g. DASH)

Differentiated service

- Should internet give different QoS (quality of service) to different applications?
  - This was an active topic of research, and standardization in 1990s
  - The ATM effort
  - The DiffServe effort in IETF
  - Most of these failed for a variety of reasons; solution was “over-provisioning”.
- But in today’s world, we have renewed interest in differentiated services:
  - Data center, and cloud services (SDN may become a solution)
  - Certain content providers want their traffic to have higher performance, and are willing to pay for it!

Summary

- We start with considering internet's service as a problem of resource allocation
- We review different policies of resource allocation;
- And then explain TCP's congestion control as a form a decentralized implementation of a certain resource allocation policy
- We discuss TCP's problems, and various extensions to TCP's congestion control mechanism
- The role that mathematical analysis of congestion control and resources allocation is more descriptive, rather than prescriptive