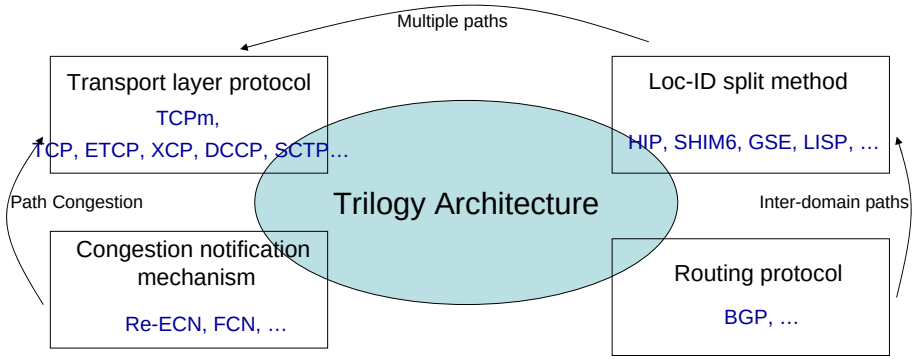
Technologies & Issues for reachability and multipath routing in Trilogy architecture

13/6/2008

Trilogy Core Protocols



TCPm

- Take a server at a multi-homed site and give it two IP addresses
- Now we modify TCP to use both addresses *simultaneously*
  - this isn't the same as other multi-homing supporting transport protocols (like SCTP,…) that switch between the paths.
- A client sets up a connection to one server's address, but in the handshake learns about the other address too.
- Now it runs two congestion control loops, one with each of the server's IP addresses.
- Packets are shared between the two addresses by the two congestion control loops
  - if one congestion-controlled path goes twice as fast as the other, twice as many packets go that way.

Loc-ID split, multipath routing & congestion control

- A method for Loc-ID separation could be the mechanism that present multiple paths to the TCPm, with the transport layer selecting how to use path A,B,C according to its congestion control algorithm.
  - Which Loc-ID split method is appropriate???
- This would be an architecture where
  - the transport layer cares about multipath congestion control, based on abstract paths (i.e. numbers), and
  - the network layer cares about providing the mechanism for actually using those paths (address rewriting/tunneling/...)

Locators and IDs

- There is a general agreement that one of the main problems in current Internet reachability schemes is that IP addresses are used as both Locators and IDs
- An ID is used to address one specific host
  - Long-term
  - Used by transport and application layers
- A Locator specifies a location (which AS, which router, which interface)
  - Short-term
  - Routers care about Locators

Loc/ID and BGP routing

- Existing work in Loc/ID split relies on BGP-based packet routing
- Multiple low-level paths may exist between End-Hosts, but for every pair of locators the path is predetermined by BGP policies.

Proposals for Loc-ID split

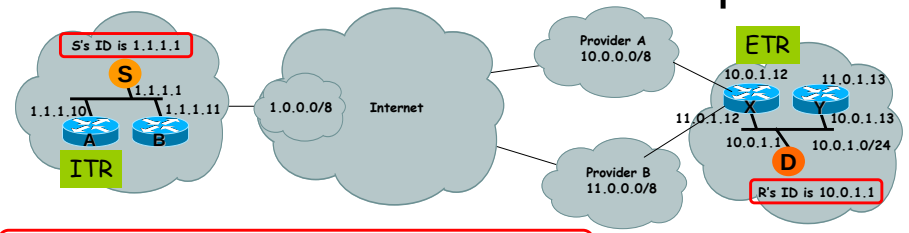
Taxonomy of approaches

- Host vs. Network based approach for mapping destination's IDs to Locators
  - Host based approaches
    - HIP, SHIM6
  - Network based approaches
    - GSE, LISP
- Direction vs. Indirection for data packet forwarding
  - Direction
    - Directly re-write destination IP address as locator
    - HIP, SHIM6, GSE
  - Indirection
    - Such as Map-and-Encapsulation, by using tunneling
    - LISP

Locator ID separation protocol (LISP)

- **Network-based** approach
  - *Ingress Tunnel Router (ITR)*
  - *Egress Tunnel Router (ETR)*
- A **Map-n-Encap** Scheme
  - EIDs are in inner headers (src-dest pair)
  - Route Locators (RLOCs) are in outer headers
- Procedures for obtaining EID-to-Locator mappings

DNS-based LISP example



1) S sends packet to D with SA=1.1.1.1, DA=10.0.1.1

2) A does a DNS lookup for '1.1.0.10.in-addr.arpa' (like ENUM…).
The A records in DNS Reply are used as locators (e.g. 10.0.1.12 and 11.0.1.12)

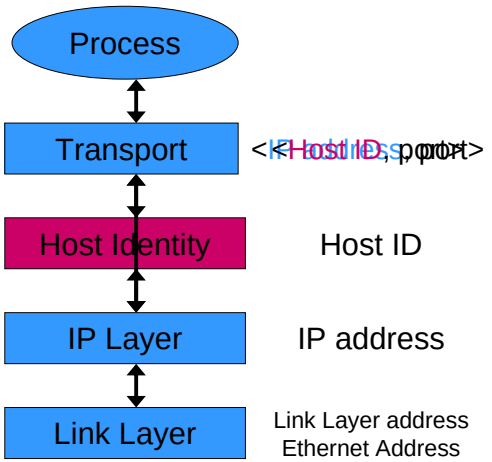
3) Packets are encapsulated with SA=1.1.1.10, DA=10.0.1.12.

4) X decapsulates each packet and delivers to host D.

HIP - Host Identity Protocol

Host Identity Protocol (HIP)

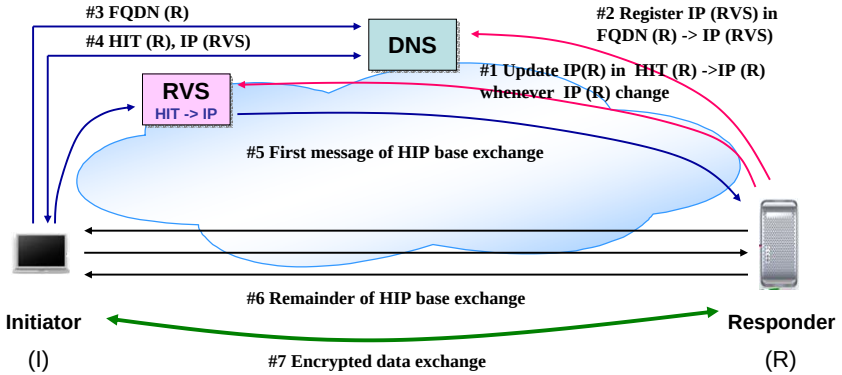
- **Host-based** approach
- Introduce a new layer in stack
  - Higher layers only see identities, not addresses
  - Mapping of identity to current IP address of destination during session setup



Host Identity Protocol (HIP)

- **Host Identity** (HI)
  - A globally unique name, chosen to be the Public Key of a Public/Private Key pair
- HIP layer uses a hashed version of the HI
  - IPv6 applications use the Host Identity Tag (HIT)
  - IPv4 applications use the Local Scope Identity (LSI)
- Translate (rewrite) HIs to current IP address transparently in the kernel
- After the mapping and before packet exchange, destination must prove his identity and agree on encryprion key.

HIP with rendezvous server

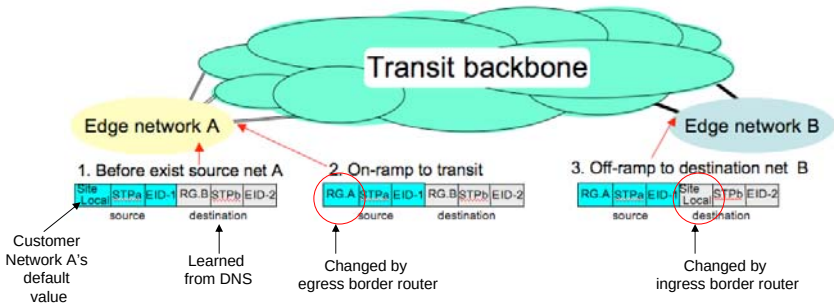


HIP supports secure Mobility and Multihoming

GSE - Global, Site, and End-system address elements

GSE - Global, Site, and End-system address elements

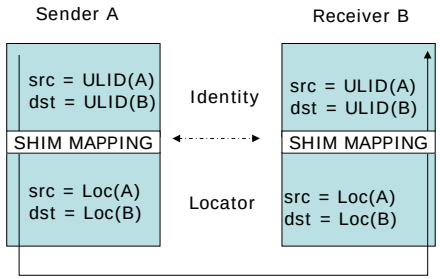
- Divide 16-byte IPv6 address into three parts:
  - lower N (=8) bytes being the End-System ID (EID), which is globally unique
  - the middle M (=2) bytes representing site topology partition (STP) for local routing,
  - and the top (16-M-N) bytes being the Locator or RG, for routing between providers.
- GSE hides a site's RG from its internal hosts and routers, so that they are *insulated* from the external topological connectivity and changes!
  - upper-layer protocols (i.e TCP) must use only the ESD



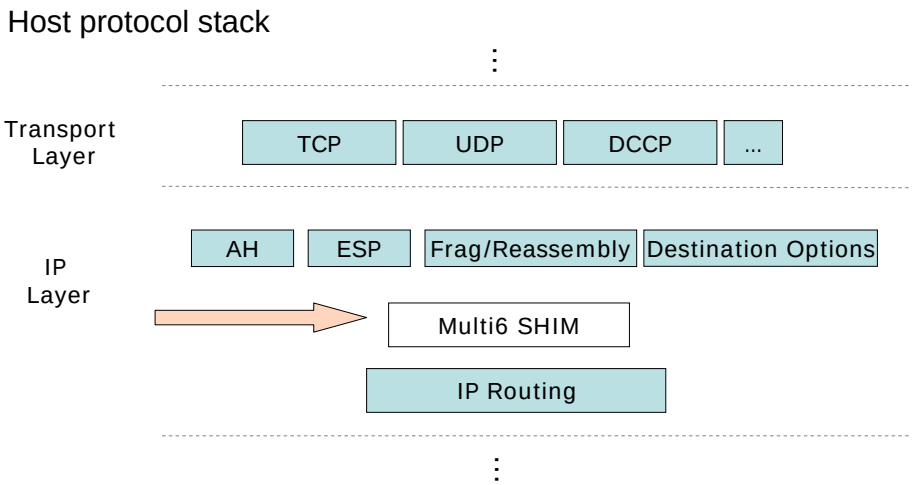
SHIM6 - Basic Approach

SHIM6 - Basic Approach

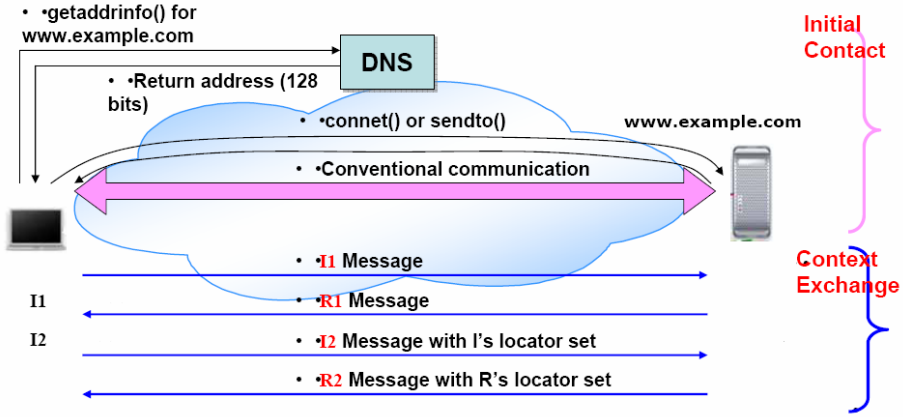
- Host-based approach
- It does not introduce any new namespace
  - IDs are routable
  - backwards compatible
- Uses an external failure detection and recovery mechanism.
- Does it allow multiple paths to be used in parallel?



Where is SHIM6 deployed?



SHIM6 example



Among the set of available addresses contained in R2, one is selected as upper-layer identifier (ULID); remaining addresses are considered as locators

GEPROD - Generic Proxying as a Deployment Tool

- A generic way of using Forwarding Proxies, designed to be used as a **transition mechanism** in implementing various flavors of the so called Identifier / Locator separation, including both "above IP" and "below IP" approaches.
- Host vs. Network based approach for mapping destination's IDs to Locators
  - Host based approaches
  - Network based approaches

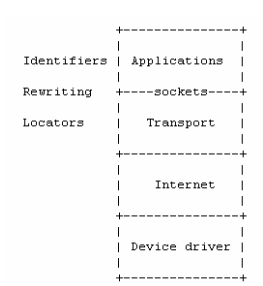
'Above Transport'
'At Transport'
'Below Transport'

draft-nikander-ram-generix-proxying-00.txt

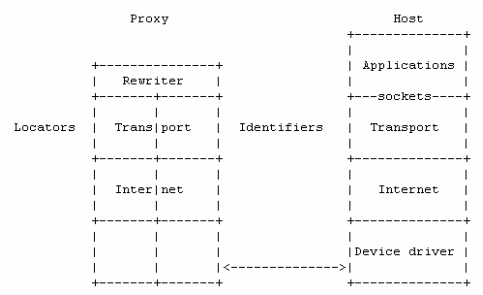
Mapping performed above Transport layer

One fundamental problem is the need for TCP to resynchronise in the case of locator change.
If the old TCP connection was not properly shut down, the hosts cannot be sure how much of sent data the other host received

Host



Network-based

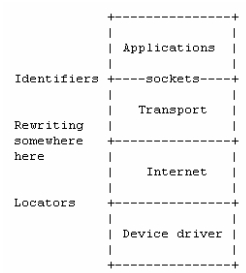


Mapping performed at Transport layer

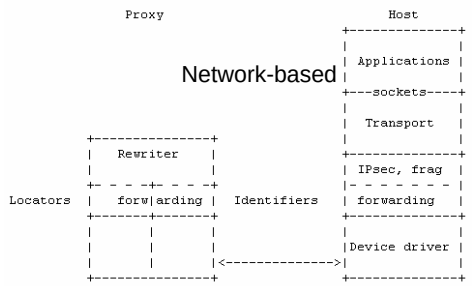
this approach is followed by HIP

Rewriter replaces locators with IDs on inbound traffic and picks suitable locators for outbound packets.

Host



Network-based

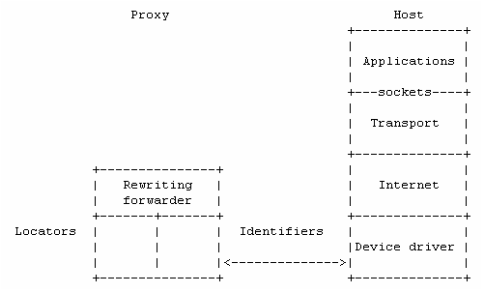


'old' host sends IP packets that have destination's identifier in their IP header (learned from DNS…)

Mapping performed below Transport layer

the main property of this category is that the rewriting is done at the 'routing' side of the IP stack, i.e., after IPsec, fragmentation, and reassembly, rather than 'above' those functions.

SHIM6 belongs to this category



Multi-path (congestion control)

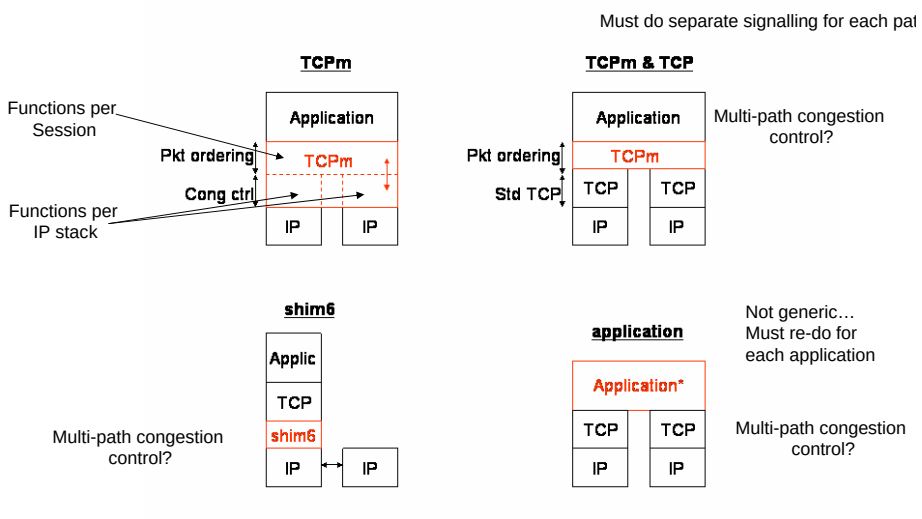
TCPm

- A transport layer protocol that will be able to utilize the existence of multiple paths towards a destination
- Its congestion control algorithm should send traffic across multiple paths, in response to congestion on each of those paths.
  - The key insight is that load-balancing and routing and congestion control are all aspects of a single algorithm.
  - Most other approaches supporting multiple paths are essentially 'multiple unipath' solutions…
    - only one path is active each time
- Some functions should be:
  - Per IP stack, i.e. congestion control
  - Per Session (packet ordering, **which of the possible IP addresses to use, how many of them, when to swap to a new IP address**, …)

Key requirements for TCPm

- TCPm must have access to info about the existence and state of different paths
  - if the network layer hides the different paths perfectly (as HIP, SHIM6 and LISP do), then the congestion state of each path is hidden
    - Crowcroft's idea …?
    - Do we have to incorporate a Loc-ID split method inside TCPm??
  - when TCPm receives a congestion indication, it needs to be able to work out which path is congested
- Some functions of TCPm must be able to operate independently of changes at the lower layers, i.e packet ordering, connection setup, ...

TCPm & alternatives



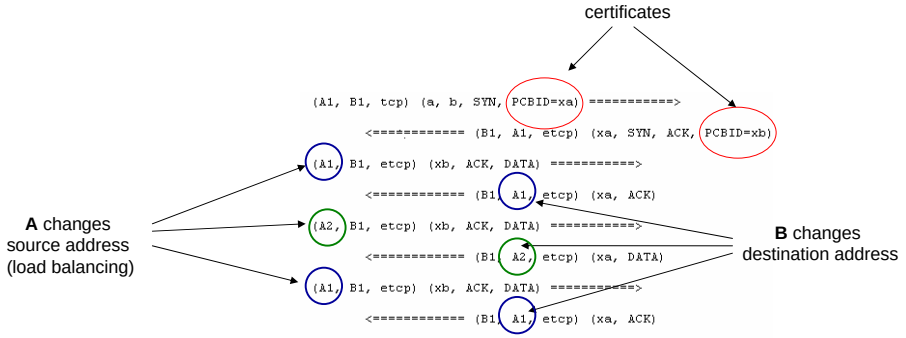
ETCP - Extended TCP

ETCP - Extended TCP

C. Huitema draft-huitema-multi-homed-01

- Extends TCP so that multi-homed hosts can change dynamically transport address(es)
- During TCP connection initialization (SYN), each host can announce willingness to accept data from alternate source addresses by including a ‘certificate’
- A compatible peer can send "Extended TCP" packets from a dynamically selected address, including in each packet the ‘certificate’ (for allowing sender identification)
  - The receiver *should* reply in that address
  - If a peer sends traffic in a Round-Robin fashion then it’s expected to receive packets in a similar way
    - For each address seen in a incoming packet the receiver keeps the ‘first seen’ and ‘last seen’ date
- How is congestion control algorithm affected??
  - Do we have more frequent time-outs?

ETCP: multi-homing example



Addresses A1, A2 will in turn be the "most recently seen" address

SCTP – Stream Control Transmission Protocol

SCTP – Stream Control Transmission Protocol

- Message-based multi-streaming
  - “Message-based” refers to preservation of data message boundaries (like UDP)
  - "multi-streaming" refers to the capability of SCTP to transmit several independent streams of messages in parallel, but over a single path each time
- Benefits of SCTP include:
  - Multihoming support
  - Delivery of data in chunks within independent streams - this eliminates unnecessary head-of-line blocking, as opposed to TCP byte-stream delivery.
  - Path Selection and Monitoring - Selects a "primary" data transmission path and tests the connectivity of the transmission path.
  - Validation and Acknowledgment mechanisms - Protects against flooding attacks and provides notification of duplicated or missing data chunks.

Multi-homing & SCTP associations

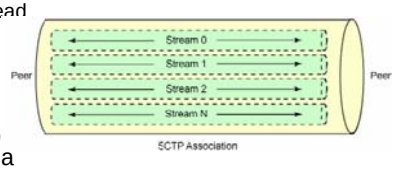
- In TCP, a *connection* refers to a channel between two endpoints (in this case, a socket between the interfaces of two hosts).
- SCTP introduces the concept of an *association* that exists between two hosts but can potentially collaborate with multiple interfaces at each host.
  - only for transparent fail-over between redundant network paths (upon detecting a path failure, the protocol sends traffic over the alternate path)

TCP connection

SCTP association

Multi-streaming & SCTP associations

- In some ways, an SCTP association is like a TCP connection except that SCTP supports multiple streams within an association. All the streams within an association are independent but related to the association
  - You might think of multi-streaming as bundling several TCP-connections in one SCTP-association operating with messages instead of bytes.



- Multi-streaming is important because a blocked stream (for example, one awaitin re-transmission resulting from the loss of a packet) does not affect the other streams in an association. This problem is commonly referred to as *head-of-line blocking*.

SCTP congestion control

- is always applied to the entire association, and not to individual streams
- AIMD
- Due to multi-homing support, the sender may need a separate set of congestion control parameters for each of the destination addresses it can send to
  - Receiver advertised window size (rwnd, in bytes)
  - Congestion control window (cwnd, in bytes)
  - Slow-start threshold (ssthresh, in bytes)

kept on the entire association

maintained on a per (destination, address) pair

Congestion Notification

FCN – Fast Congestion Notification (or GCN – Granular Cong. Notif.)

- The problem with TCP is that is inefficient, especially as the product Bandwidth x Delay for a network increases
  - Also because of the conservativeness of its Additive Increase and slow start algorithms
- The aim of FCN is that end hosts get *precise* information about the current congestion along the path for each packet
- The source can therefore learn (via the receiver) almost immediately (within a RTT) exactly what the total congestion is on the path, from which it can calculate the appropriate equilibrium sending rate
  - ECN has only a single codepoint (CE) and so it effectively takes end hosts many packets to learn about the congestion level

Theoretical work on algorithms for stable multipath congestion control

- There are two sorts of multipath congestion control:
  - primal (end-systems control their sending rates or window sizes, like TCP) [Voice and Kelly]
    - Run separate copies of TCP on each of the paths, but coordinate their window adaptation.
    - Specifically, each of the TCPs should increase its rate whenever it receives an ACK on its path;
    - and each of the TCPs should cut its rate whenever it detects a drop on its path; and the drop should be proportional to the user's *total* transmission rate.
  - and dual (routers send a congestion measure to traffic sources, like XCP).

XCP: An eXplicit Control Protocol

Proposed Solution:

Decouple Congestion Control from Fairness

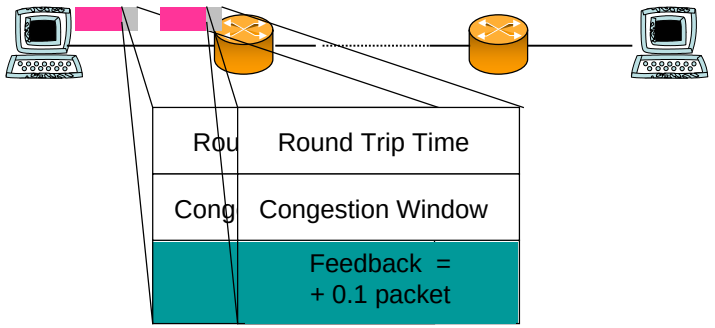
Coupled because a *single* mechanism controls both

Example: In TCP, Additive-Increase Multiplicative-Decrease (AIMD) controls both

How does decoupling solve the problem?

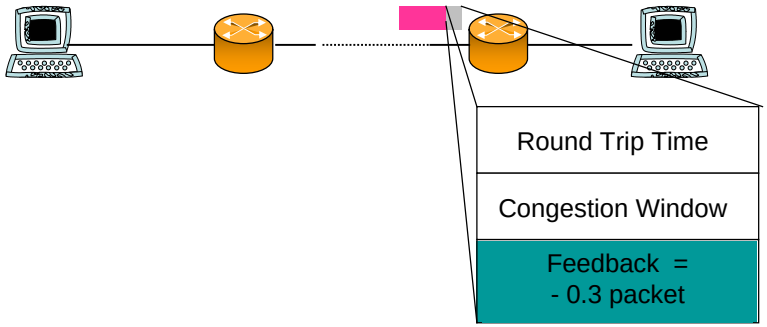
1. To control congestion: use MIMD which shows fast response
2. To control fairness: use AIMD which converges to fairness

How does XCP Work?

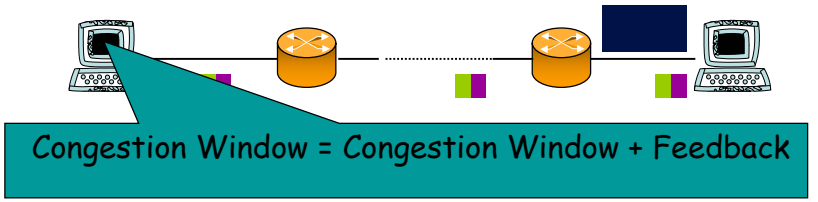


Congestion Header

How does XCP Work?



How does XCP Work?



How Does an XCP Router Compute the Feedback?

Congestion Controller

Goal: Matches input traffic to link capacity & drains the queue

Looks at aggregate traffic & queue

MIMD

Algorithm:

Aggregate traffic changes by Δ

Δ ~ Spare Bandwidth

Δ ~ - Queue Size

So, $\Delta = \alpha\, d_{avg}\, Spare - \beta\, Queue$

Fairness Controller

Goal: Divides Δ between flows to converge to fairness

Looks at a flow's state in Congestion He

AIMD

Algorithm:

If $\Delta > 0 \Rightarrow$ Divide Δ equally between flows

If $\Delta < 0 \Rightarrow$ Divide Δ between flows proportionally to their current rates

A (possibly incomplete) list of issues…

Architecture's impact on multi-path congestion control

- TCPm must have access to info about the existence and state of different paths
  - if the network layer hides the different paths perfectly (as HIP, SHIM6 and LISP do), then the congestion state of each path is hidden
    - Crowcroft's idea …?
    - Do we have to incorporate a Loc-ID split method inside TCPm??

Multi-path & Congestion control

- How congestion control algorithms are affected by the ability of per packet path selection?
  - Link utilization?
  - Stability?
  - Fairness?
  - …
- How path setup should be done?
  - how many/which of the available IP addresses to use?
    - Should path diversity play any role?
  - when to swap to a new IP address?
- Does multi-path congestion control can do better traffic engineering than network providers can achieve today?
  - Is it because of fast timescales?

Better throughput for the application

- What is the benefit for the end user?
  - Using multiple paths at once, with the rate on each adapting according to their congestion, should mean that the user's throughput tends to be maintained at a higher average rate.
  - If instead you used one path at a time & switched when the first got too bad, you'd have the switching cost (got to realise first path is bad, got TCP slow start on 2nd path, etc)
- The details of multipath congestion algorithm are very important…
  - what the multipath algorithm is expected to do in the case of link failure and re-routing on one of its paths (which would show up as a loss or delay spike) - does it have to slow start as normal, could it slow start less aggressively, ...?

TCPm & edge-2-edge traffic

- end-2-end traffic can sometimes be changed into a sequence of edge-2-edge traffic
  - Like paths and links in overlay networks…
  - Examples of edge-2-edge traffic: traffic between middleboxes (SIP B2BUA, NAT, …)
- the ISP itself could maybe run TCPm (ie run TCPm edge-to-edge rather than end-to-end), in which case policy could directly influence rate it sends on each link
  - When would this be attractive?
  - How to incentivise inter-domain multi-path?

Increased competition between network providers

- suppose that users are multihomed, able to move their traffic on a very fast timescale, and that their usage is capped in some way
- If end users can do multipath congestion control, there will be some economic impact on network providers.
  - Multihoming should increase competition between ISPs, and multipath in general should increase competition between transit network providers.
  - What sort of economic models are there for this?
- A user's traffic rate on a path signals his preferences to the network providers
  - But each provider has information about paths he is participating only …
  - What sort of economic models are there for this?
  - How this information can be used for deciding how to invest in capacity?

Traffic management games

- What sorts of traffic management games the providers actually like to play (i.e. beyond the 'hot potato routing')?
  - Will that sort of game tend to act to recombine routes which start out diverse (which would be a bad thing)?
    - Do they have the info to do it in purpose?
  - Are providers ever motivated to try to carry more traffic (for example, if peering was not settlement free, presumably providers with efficient infrastructure would try to do this)?
    - What if ReECN is turned on?
  - Assuming TCPm, would an ISP slow down the forwarding rate, which would influence the TCPm to increase rate to other ISP?

End…

TCP Congestion Control

- When **`CongWin`** is below **`Threshold`**, sender in slow-start phase, window grows exponentially (from low starting point).
- When **`CongWin`** is above **`Threshold`**, sender is in congestion-avoidance phase, window grows linearly.
- When a triple duplicate ACK occurs, **`Threshold`** set to **`CongWin/2`** and **`CongWin`** set to **`Threshold`**.
- When timeout occurs, **`Threshold`** set to **`CongWin/2`** and **`CongWin`** is set to 1 MSS.