

Host/Application Based Multi-Homing

Aleksi Suhonen

Tampere University of Technology

Department of Communications Engineering

8.4.2009 / ICT SHOK FI WP1

Problem Statement

- Internet **Adoption** leads to **Reliance**
- **Reliance** has led to need for **Resilience**
- One method to increase **Resilience** is BGP based **Multi-Homing**
- **Multi-Homing** has led to DFZ **table fragmentation**
- This has led to calls to separate **locator** and **ID**

Old Solution

- Locator and ID were separated long ago
- IP is the locator
- FQDN/DNS is the ID
- I'm advocating host/application based multi-homing instead of network/address based multi-homing
- Presented ideas do not necessarily fix all problems, but will have noticeable impact
- They will also work well in conjunction with other solutions

Problem w/Old Solution

- Real world usage does not make good use of the possibilities provided by the technology
 - Name servers are seldom well distributed wrt network topology
 - There are usually only two or three name servers
 - There's usually only one IP address for servers
 - Applications often only try one address at a time, even if many are provided, causing suboptimal setup times
 - Applications seldom make use of DNS hints or multipathing protocols
 - Multipathing congestion control and load balancing methods aren't implemented yet

New Solution

- Write up new best current practises that would make better use of existing protocols
- Infiltrate some relevant open source projects and improve support from within
 - E.g. Apache, Mozilla, Squid, gnu libc, ...
 - Commercial products tend to follow when established BCPs and prototype implementations have proven themselves, otherwise they might lose market share
- Research and develop interaction between the network and multipathing protocols, such as SCTP

Multipathing Protocols

- Initially I thought I'd focus on SCTP, but many of the ideas or concepts can also be applied to:
 - HIP
 - iSCSI
 - Multipath TCP
 - ...
- Keep this in mind whenever multipathing is mentioned

DNS Hints

- Name service should give hints about the locators:
 - Multi-Homed
 - Anycast
 - Reverse Proxy
 - ...
- Use some old DNS RRs or standardize new ones?
 - Maybe IN SRV could be used for these hints?
 - Maybe SPF and others could adopt whatever hint mechanism we end up with?
- Resolvers and applications may function better with hints

DNS Hint: Multi-Homed

- FQDN has several A and/or AAAA records
- They are all different addresses of the **same** host
- Multipath protocols can assume that they represent the same host and can pre-optimize themselves
 - E.g. pre-populate a peer address table
 - Attempt simultaneous connection initiation with multiple given addresses using the same connection ID

DNS Hint: Anycast

- A single address is shared by several hosts
- Routing changes may lead to packets flowing to a different host
- Connection oriented protocols (e.g. TCP, NTP) may want to keep sessions short or to prefer unicast addresses over anycast addresses, if both are available
- Multipathing protocols may wish to migrate to a unicast address after session initiation

DNS Hint: Reverse Proxy

- A reverse multiplexer is used to load balance incoming connections to one address between several hosts
- Much like anycast, except:
 - An established connection will flow to the same host from beginning to end
 - Subsequent connections are likely to go to different hosts
 - Connection oriented protocols do not need to be afraid of being rerouted unexpectedly

Other DNS Hints

- Research may come up with even better ideas
- Using IN LOC to as a coarse grained optimization input?
- What about mixed cases? (e.g. two multi-homed hosts under one FQDN)
- IN SRV could be used to tell if a service is available over TCP, UDP, SCTP and ... (IPv6?)

Improving Application Performance

- Applications should try to initiate connections to more than one address semi-simultaneously
 - Not too many of course, within reason
 - This improvement would be especially beneficial to IPv6 migration
 - Non-multipathing protocols may gain some of the advantages enjoyed by multipathing protocols
- Add SCTP support to most popular TCP based applications

Improving Resolver Performance

- Resolvers should send simultaneous queries a bit more aggressively as well
- Resolver API should be able to pass hints to applications
- Or perhaps the resolver functionality should be incorporated to the TCP/IP stack
 - Instead of opening a TCP socket to an IP address
 - Apps would be opening reliable data streams to FQDNs
 - (Vogt has written a very promising I-D regarding this idea)

Improving Address Selection

- The current RFC for address selection is under review
- Some of the considerations presented here should also be considered in that process

Improving Multipath Performance

- Find out how different queueing and congestion control strategies affect multipathing protocols
- Find out how network signaling (e.g. ICMP) affects multipathing protocols
- Find out how current implementations do best path selection and load balancing
- Find out how lossy (wireless) links affect multipathing protocols
- Improve on the results of all of the above