Providing Packet Obituaries

Katerina Argyraki (Stanford, ICSI)
Petros Maniatis (Intel Research)
David R. Cheriton (Stanford)
Scott Shenker (UCB, ICSI)
Best-efforts delivery

- Network forwards packets best-efforts
  - simplicity
  - scalable, robust algorithms
- No guarantee about what *will* happen to each packet
  - a feature, not a bug
Opaque failure

• No account of what *did* happen to each packet

• Your packets are dropped somewhere
  – local administrator blames ISP
  – ISP blames “the Internet”
  – no one can prove anything

Internet provides no accountability
Network espionage

- Ping, traceroute, etc.
- Sneak inside ISP internals
  - ISPs have incentive to disable
- Probes are post-hoc
  - doesn't say what happened to real traffic
- Not an architectural solution!

Probing may soon be over
Solution: accountability framework

- ISPs provide feedback on packet fates
- End-systems can use to detect problems quickly and accurately

Each ISP reports on forwarded traffic
Basic information reported

• Where packets get dropped
  – verify SLA
  – route around failing/congested areas

• AS-level granularity
  – which specific link/router failed useful only to local AS
  – SLA between ASes
  – source routing at AS-level
Infrastructure

- A-boxes on inter-AS links
  - can be integrated in border routers
  - can be shared between ASes
Hop-by-hop feedback

- Keep per-packet short-term state
  - digest, lastAS, prevAbox, timeout
Hop-by-hop feedback

- Keep per-packet short-term state
  - digest, lastAS, prevAbox, timeout
Hop-by-hop feedback

- Keep per-packet short-term state
  - digest, lastAS, prevAbox, timeout
Hop-by-hop feedback

- Keep per-packet short-term state
- Periodically send to previous A-box
Hop-by-hop feedback

- Keep per-packet short-term state
- Periodically send to previous A-box
- Previous A-box merges feedback with local state
Hop-by-hop feedback

- Keep per-packet short-term state
- Periodically send to previous A-box
- Previous A-box merges feedback with local state
Hop-by-hop feedback (cont'd)

- Short-term state removed when
  - feedback received or
  - timeout expired
- Moved to long-term state
  - for post-facto investigation
Hop-by-hop feedback (cont'd)

Each AS that forwards a packet learns in which AS that packet died.
Resources: memory

- Short-term state, transmission buffer, long-term state
- Assume OC192 link
  - 400 bytes avg packet size
- To support 3 Mpps
  - 44 MB of CAM, 80 MB of SRAM
  - 220 GB of long-term storage

Reasonable requirements for WAN link
Resources: bandwidth + processing

• Bandwidth depends on feedback entry size + avg packet size
• 4.6% bandwidth overhead
  – 96-bit entries
  – 400 bytes avg packet size
• 2 CAM lookups + updates per packet

Reasonable resource requirements
Challenges

- Ingress point disambiguation
  - encapsulation
- Determine entry timeout
  - destination prefix to AS-path length map
Alternative design

- Keep per-packet digest
Alternative design

- Keep per-packet digest
Alternative design

- Keep per-packet digest
- Periodically compress and send to source AS (AS-A)
Alternative design

- Keep per-packet digest
- Periodically compress and send to source AS (AS-A)
• Keep per-packet digest
• Periodically compress and send to source AS (AS-A)
• Separate feedback from each AS
Alternative design (cont'd)

- No encapsulation
- No timing dependencies
- More bandwidth overhead?

Source AS learns where its packets died
Further variations

- Per-flow feedback
  - not per-packet
- Add delay info
- AS-path info
Security issues

I delivered!

I never got it!
Security issues

- Source narrows it down to two ASes
  - seeks alternative route
- All involved ASes notified
  - falsely accused AS knows who lied
Approach 1: Digital signatures

- Source sends post-facto queries to involved ASes
- Sign queries + responses
- Requires public keys

C dropped it!
Approach 2: Disjoint paths

- Send feedback through disjoint paths
- No public keys
- Requires route control
Summary

• Probing not a long-term solution
  – ISPs likely to disable
  – it's not enough anyway

• Accountability framework instead
  – ISPs report back to source/other ASes

• Reasonable resource requirements

• Secure…
Impact on the Internet

• Expose ISP performance
• Help make routing decisions
  – much said about end-controlled routing
  – less on how end-system collects info
• Faithful to end-to-end principle
  – don't expect network to adapt
  – end-systems collect info, adapt to network