Packetization Layer Path MTU Discovery

IETF BOF

To WG or not to WG?
3/20/03

Matt Mathis <mathis@psc.edu>
John Heffner <jheffner@psc.edu>
Kevin Lahey <kml@patheticgeek.net>
Peter O'Neil <poneil@ucar.edu>
Fred Templin <osprey67@yahoo.com>

http://www.psc.edu/~mathis/MTU/

Packetization Layer Path MTU Discovery

- This new algorithm does not rely on ICMP or other messages from the network (so it is not subject to the problems described in RFC2923). Instead it finds the proper MTU by starting with relatively small packets and searching upwards by probing with test packets.

- This BOF is to clarify the scope of the work and to discuss how it might be completed within IETF.

Agenda

- Background (40 Minutes total)
  - Quick overview of the problem
  - Quick overview of the work-to-date
  - Quick introduction to open issues
  - Inventory of intersecting standards & technologies
    - Harmed by current problem
    - We need to consider interactions
- How can we best complete the work? (20 Minutes)
  - Is this issue currently relevant to the IETF?
  - Is this work tractable within IETF?
  - How do we consider the operational impact?
  - How do we consider deployment strategies?

Current Path MTU Discovery

- Defined in RFC1191 (IPv4) and RFC1981 (IPv6)
  - Send a large packet with the Don't Fragment (DF) bit set
  - Routers forward packets as possible; send an ICMP message when the packet is too big for the outgoing link
  - Host receives ICMP message, starts sending smaller messages
- How small can an MTU Be?
  - IPv4: 68 octets [RFC791]
  - IPv6: 1280 octets [RFC2460]
- With IPv4, we can send larger packets without the DF bit set and they will be fragmented in the network. IPv6 routers will not fragment, but since the minimum link size is larger, this less of an issue.

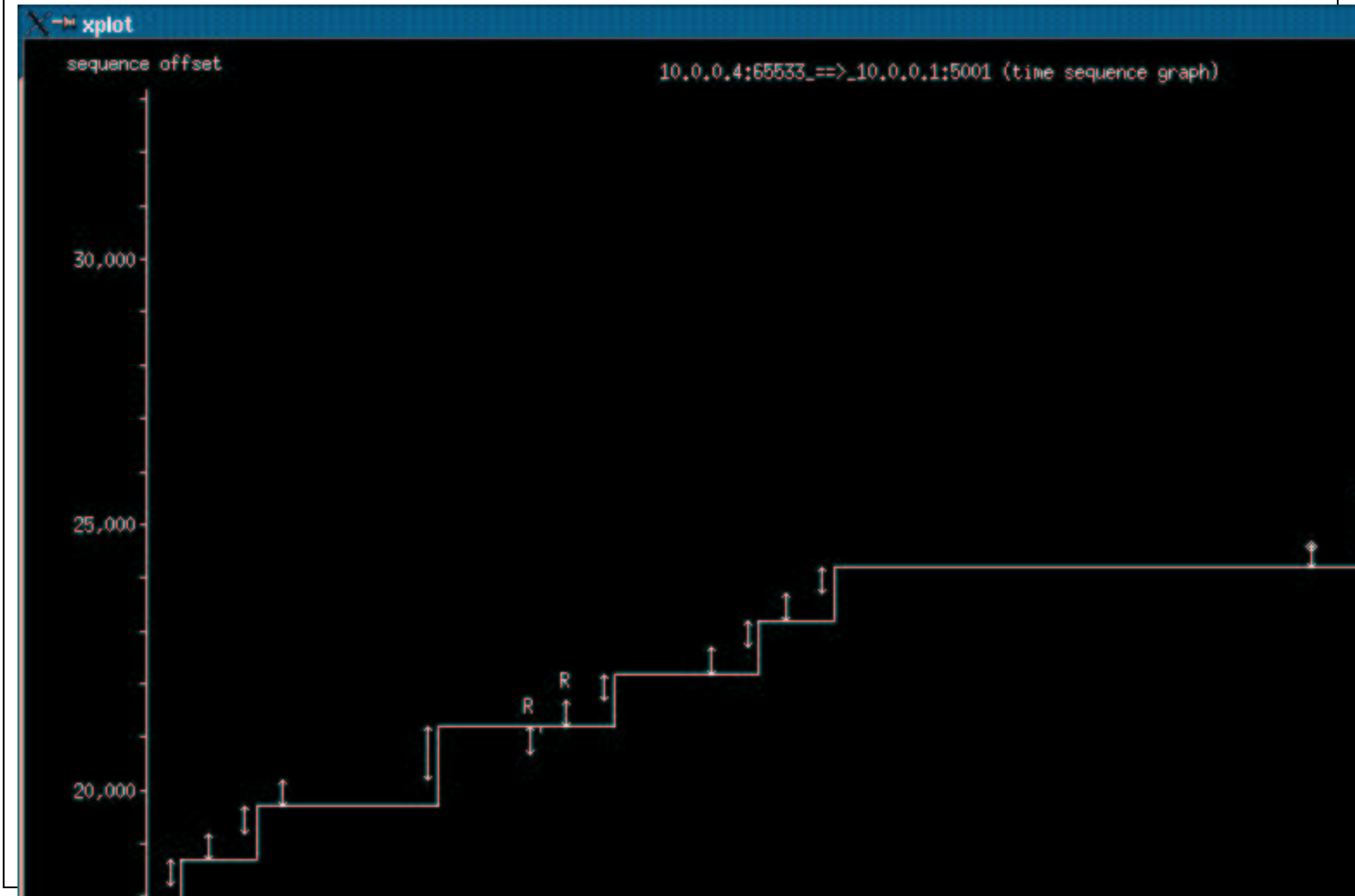
ICMP Issues

- ICMP message are sometimes blocked
- Broken routers don’t bother to send ICMP messages
- Tunnel endpoints fail to send ICMP messages, or do not notice their own ICMP messages.
- Weird layer-2 devices may bridge networks with different MTUs, and silently drop large packets.
- Broken PPPoE implementations are frequently an issue.

History and work-to-date

- Prior PLPMTUD documents
  - Unpublished work following 41st IETF (1998)
  - RFC2923: Lahey, problems with pMTUd (Sept 2000)
  - Non-IETF web draft (7-Nov-2002)
  - draft-mathis-plpmtud-00.txt (Feb 23, 2003)
- Implementations
  - Unpublished running code in 1998
  - Two written to the 7-Nov-2002 draft
    - John Heffner (Linux)
    - Kevin Lahey (NetBSD)
- Current draft does not reflect lessons learned
  - Need input from other areas

Running code



The Proposed New Algorithm

- Start with a "moderate" MTU (1k?)
- Test larger MTUs by probing
  - Raise MTU if successful
  - (Optional) process any RFC1191 ICMP
  - Do not reduce TCP window on lost probes
  - (Lots of details in the spec)
- Most of the algorithm runs in the transport layer
  - TCP, SCTP, or higher layer (e.g. NFS).
- But cache/share state via IP layer

Document Issues

- Minimal tight standards language
  - Lost probes do not trigger TCP congestion control...
- Nearly everything else is heuristic
  - Mostly just recommendations
  - How to avoid over specification?
  - Tweaks are likely to continue forever
- Start as extensions to RFC1191 and RFC1981
  - But plan to subsume 1191 in the longer term(?)

Open MTU caching issues

- How to use the IP layer to share state
  - concurrent and successive connections to the same remote host.
- What is cached?
  - How do multiple connections with different headers interact?
- IP layer has to account for IPsec, tunnels, etc
  - may differ per connection to the same remote host
- What about NAT (1 addr w/ differing MTU)
- Or just don’t bother caching, always start small?

Open Robustness Issues

- Multi-path with differing MTUs
  - Require an additional lossless RTT(?)
- On double(?) timeouts, reset MTU to 512(?)
  - and trigger restarts at other layers?
    - router discovery
- On persistent double timeouts, stop probing(?)
- Heuristics to reduce MTU(?)
  - Window becomes too small
  - Large MTU causes higher loss rate

Intersecting Standards & Technologies

- What protocol (deployments) have been harmed by pMTUd problems?
- 341 RFCs contain the word "tunnel"
- What standards might interact with PLPMTUD
  - e.g. should TCP-IW become 4 * 1k?

Collected input

Lower layers:

- 2516 PPPoE
- 2401 IPSec
- 2893 IPv6 over IPv4
- 3056 6 to 4
- ISATAP Internet-Draft
- Wireless handoff causing mtu changes

Overlay networks: emulab, xbone, mbone, etc

Transport:

- 2414 TCP Initial Window
- 1889 RTP
- multicast
- RTTP/DCCP
- SCTP
- 3489 STUN (NAT traversal)

Event Notification:

- 2338 Virtual Router Redundancy Protocol
- 2281 Cisco Hot Standby Router Protocol

Who ignores the DF bit?

Working Group Issues

- My broader goal (Pushing up the Internet MTU) seems to be out of scope for this process.
- See http://www.psc.edu/~mathis/MTU
- If we charter a new WG, it will have a new mailing list.

Questions to Discuss

- Is this issue currently relevant to the IETF?
- Is this work tractable within IETF?
- How do we consider the operational impact?
- How do we consider deployment strategies?

(or do we want to go for the broader goal?)