Path Maximum Transmission Unit Discovery

draft-ietf-pmtud-method-02.txt
3-Aug-2004

Matt Mathis <mathis@psc.edu>
John Heffner <jheffner@psc.edu>
Kevin Lahey <kml@patheticgeek.net>

Slides: http://www.psc.edu/~mathis/papers/pmtud200408

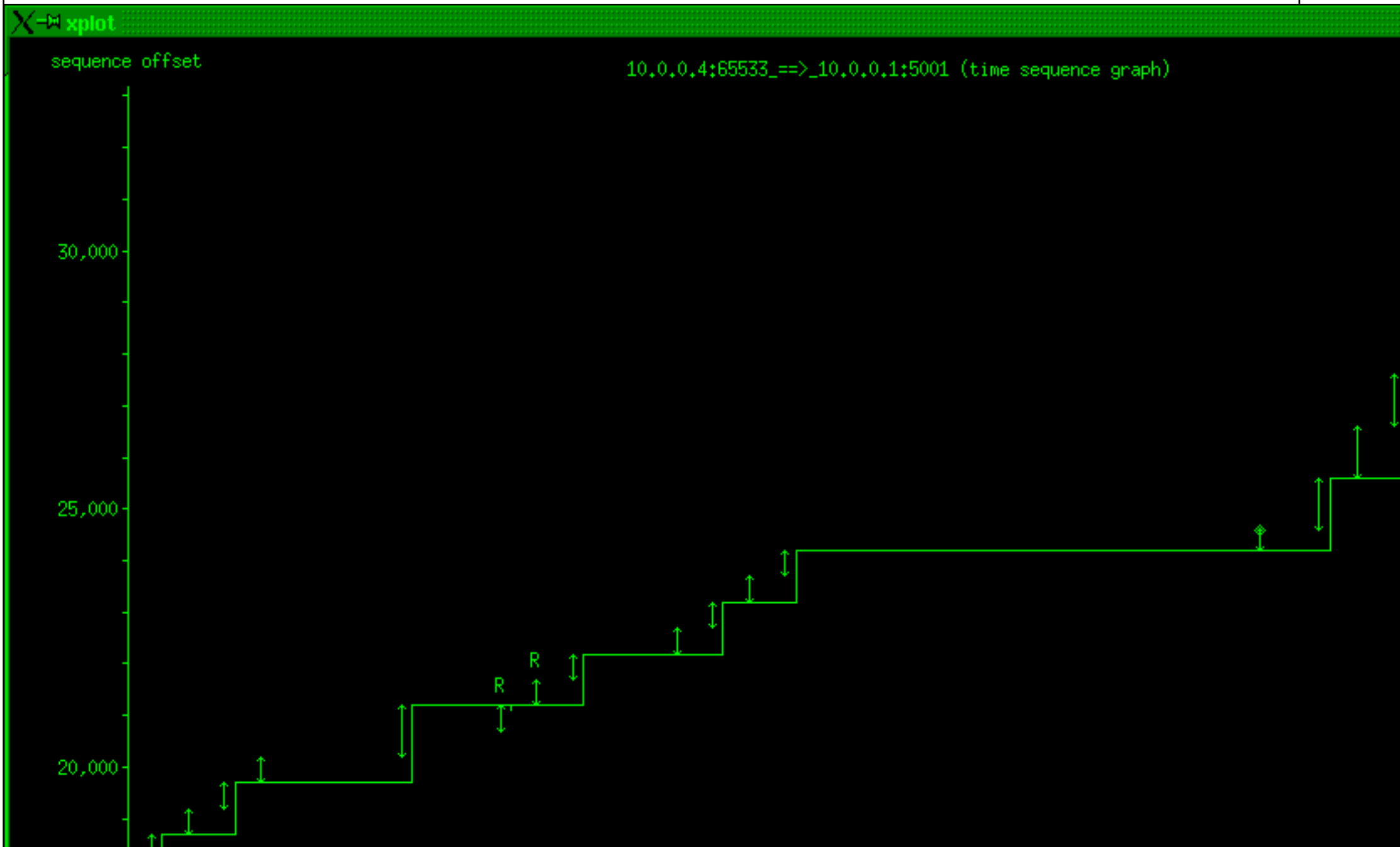
Comments to the list:
pmtud {-request} @ietf.org

Algorithm Review

- Start with 1k MTU
  - RFC 2414 allows 4*1k Initial window
- Test larger MTUs by probing with larger packets
  - Provisionally raise MTU if successful
  - (Optional) process any RFC1191/1981 ICMP
  - Do not reduce TCP window on lost (unsuccessful) probes
- Verify provisional MTU for 1 RTT
  - Additional losses imply MTU limits
  - (Total time is 3 RTT per MTU step)
- Most of the algorithm runs in the transport layer
  - TCP, SCTP, or higher layer (e.g. NFS)
- Keep cached/shared state in the IP layer
  - IP Maximum Payload Size (MPS)

Read the Internet-Draft!

Running Code



Key Properties

- We are not defining a protocol
  - A method of using existing protocols
- Careful thought to maximize robustness
  - Try to compensate for bugs elsewhere
- Implementation differences do not affect interoperability
  - Start now
    - The hard part is getting clean layering

Status Update

New unofficial status page

- http://www.psc.edu/~mathis/MTU/pmtud/index.html
- Live draft: -02bis is already open
  - including a version with change bars!
- Updates in "near real time"

Overview of document changes

- No significant changes in the algorithms
  - Focus on improving the clarity and generality
- Restructured for cleaner layer separation
  - Still not as clean as it should be
- Much less TCP centric
  - TCP turns out to be the hard case
  - Added first draft SCTP as a Packetization Layer
    - Have since discovered a better method
  - IP Fragmentation as a Packetization Layer
    - Needs an adjunct protocol to do the probing
  - Still more protocols needed

Robustness issues

- Added discussion of "full stop timeouts"
  - Potentially interacts with other parts of the stack
    - e.g. NIC restarts, first hop router discovery
  - More research (and references) needed
    - Is there a volunteer?
- Removed state machine to detect pMTU discovery induced failures
  - Pathological cases where raising MTU causes failures
    - e.g. Router/NIC restarts
  - Deemed not worth the complexity for automatic detection
- Devices that ignore DF remain the big worry

New Topics

- Tunnel discussion
  - Includes a sermon on not ignoring DF
    - See: draft-mathis-frag-harmful-00.txt
  - Fold in ideas from Michael Richardson
    - draft-richardson-ipsec-fragment-01.txt
- Subnets with non-uniform MTU
  - Solve major operational problem today
    - Any node that can't raise MTU vetos upgrading a subnet
    - Does anybody know of a reference?
  - Becomes a non-problem with PLPMTUD
- Recommendation that IPv4 fragmentation emulate IPv6
  - Only use host fragmentation
  - Always set DF on Fragments
  - Must treat IP fragmentation as a PL

Additional Packetization Layers

- In principle we need a section for "every" Packetization Layer
  - Including all future protocols
- Would be better to generalize based on common properties
  - How to generate probes
    - Use live data (TCP only?)
    - Pad with out-of-band data (SCTP, RTP?)
    - Use adjunct protocol for probes (UDP, DCCP)
  - How is the effort balanced between the application and OS?
- Approach
  - Design methods for several specific protocols/applications
    - e.g SCTP, generic UDP, NFS, IP fragmentation
    - Input would be a huge help
  - Identify and generalize common properties

Design questions per Packetization Layer

- How should probes be generated?
- How does the sender know for sure that probes are (not) delivered?
- What are the costs of successful and unsuccessful probes?
- How should the verification phase be implemented?
- Are there special restrictions on changing packet sizes?
  - e.g. does it require cooperation by the application
- Are there any other restrictions on MTU?
  - e.g. SCTP multi-path requires that all messages be acceptable to all paths

The end

pmtud {-request} @ietf.org