

pmtud implementor's report

John Heffner
IETF 62
Minneapolis, MN
March 8, 2005

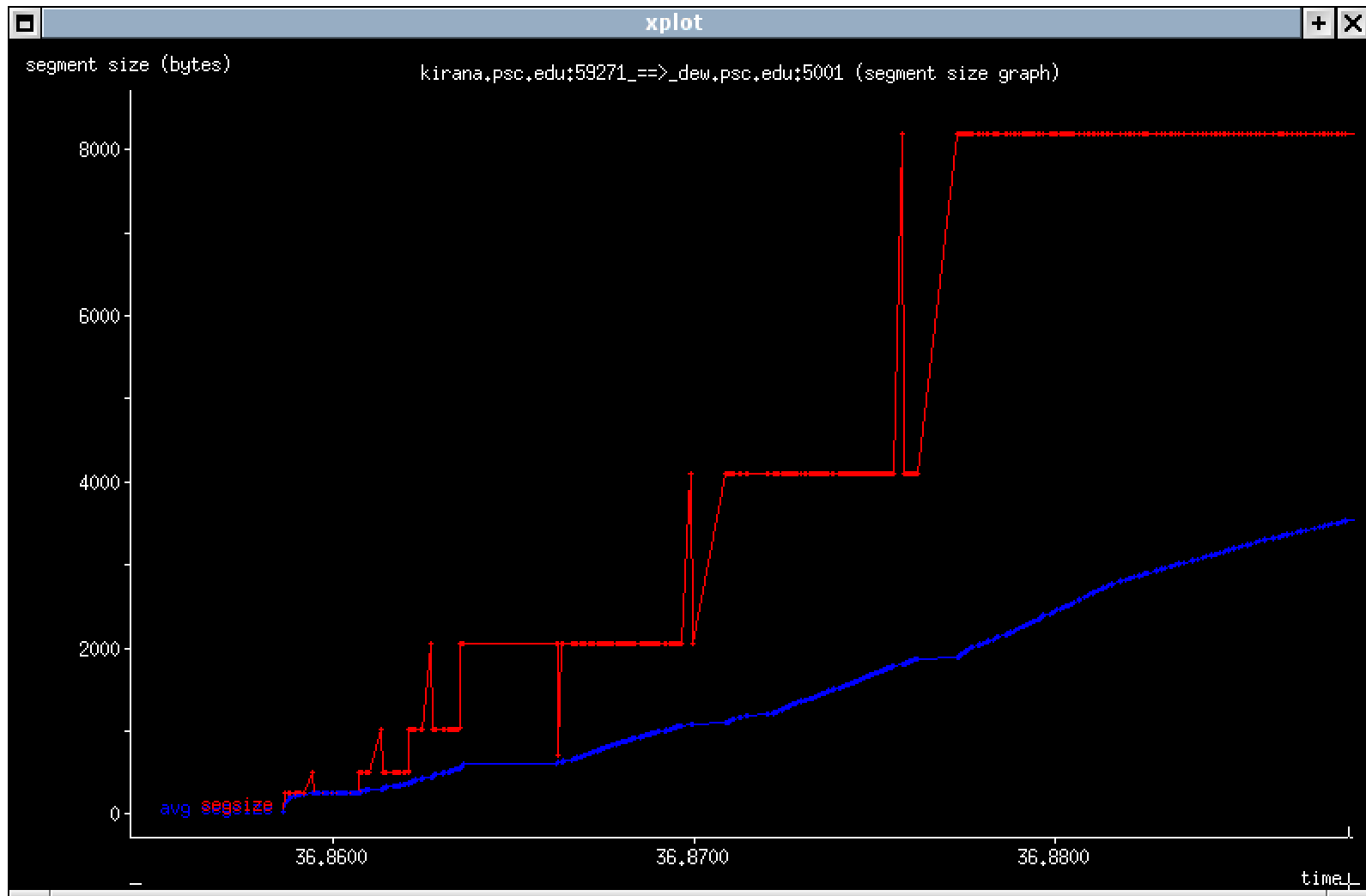
New implementation

- TCP implementation for Linux 2.6 (currently 2.6.11, tracking Linus's bitkeeper tree)
- Just “finished”
 - Still working on it, but the basics are there.
 - Not well tested. **Volunteers (guinea pigs) are welcome.**
- Can be found at
<http://www.psc.edu/~jheffner/patches/mtup-2.6.11.patch>
- Expect new versions.
- I may put up a more complete web page with a change log. I'll post to the list about this.

Eating my own dogfood

- Running on my laptop right now

In action



Implementation overview

- Selection of initial MSS
- Search strategy
- Deciding when to probe
- Verification
- Response to probe/verification results
- Moving to new MSS

Selection of initial MSS

- Currently a sysctl variable
- Idea:
 - Start with maximal mss from current pmtu, and enter verification phase immediately.
 - Failure (timeout) results in backing off to search_low.

Search strategy

- Very simple: double current MSS
- If $\text{target} > \text{search_high}$, we are done
- May implement more complicated heuristics later, but:
 - Maybe this strategy is good enough
 - MSS being an exact multiple or fraction of page size is good for performance

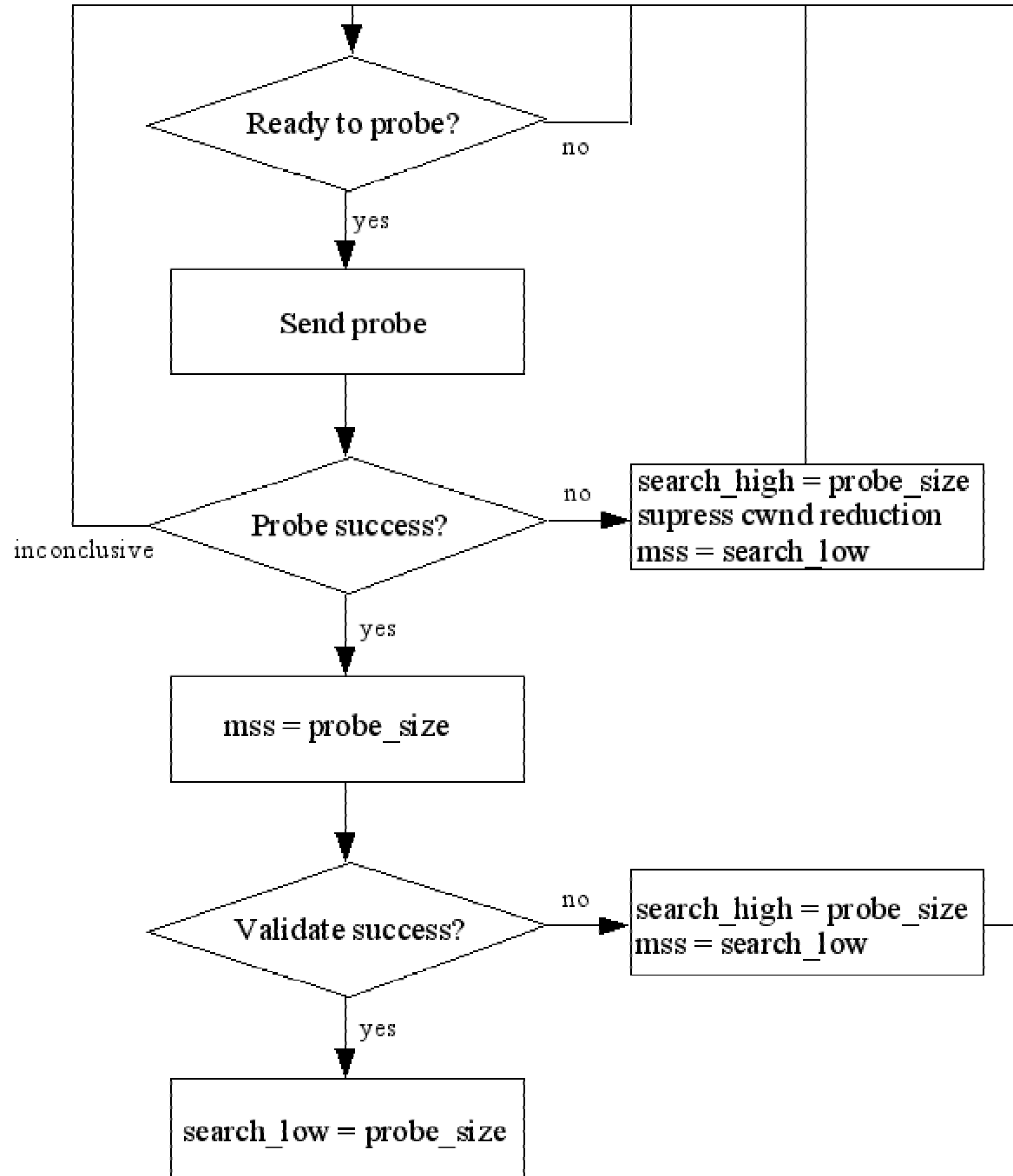
Deciding when to probe

- Three results
 - Probe now
 - Don't probe (continue to send data if appropriate)
 - Wait (don't send probe or any data)
- Tests:
 - Currently probing or verifying Don't probe
 - In recovery Don't probe
 - $cwnd < 11$ Don't probe
 - Less than `probe_size` data in send queue Don't probe
 - $packets_in_flight + 2 > cwnd$ Wait
 - probe not in receive window
 - $rwin < probe_size$ Don't probe
 - else Wait
 - Otherwise Probe now

Verification

- Want to use only full-sizes packets for verification
- If header lengths change, hard to determine exact length of packets when sent (we only know the payload length)
 - Is this “good enough”?
- Chose to use a fixed number of 10 packets for verification, not cwnd as the draft recommends
 - My verification is currently fragile since I don't time out and retry

Response to probe/verification results



Moving to the new MSS

- Linux segments data when copying from user space
- Can have a full send buffer of data already segmented at old MSS
 - Verification could take a while
- Consider probing higher before verification complete?

What's NOT implemented

- Timeout/retry for various events
- Handling of some failure cases (mostly related to above)
- Recommended search strategy (no fine scan)
- ICMP attack protection

Open issues

- Adding learned data to route cache
 - Especially important for short-lived flows
 - What to cache?
 - Is saving search_high too fragile?
 - How often to access? (Locking issues)
- Bogus ICMP handling
 - Shared IP-layer pmtu value causes difficulty
 - Would need partition between “secure” and insecure protocol pmtu's
 - Maybe an issue for tsvwg? (current discussion in tcpm)
 - Not planning on implementing anything here soon
 - Paranoid systems can filter ICMP and rely only on mtu probing