TCP Fast Open – an Update

draft-cheng-tcpm-fastopen-00.txt

H.K. Jerry Chu - hkchu@google.com
Yuchung Cheng - ycheng@google.com

Quick Recap

- Design principle - KEEP IT SIMPLE
  - Address a performance, not security problem
  - Reasonable security measure
    - High-strength security mechanism an overkill
    - Potential damage limited anyway
  - Server stateless (no per-connection state)
- Requirement
  - Transparent, backward compatible
  - Middlebox friendly to minimize deployment issues

Key Issues

- Consuming data before 3WHS introduces three problems
  - Duplicate/stale SYNs
    - Allowed for apps that are tolerant of stale/dup requests
  - Server Resource Exhaustion attack
    - Bogus requests with spoofed source IP burn CPU cycles
    - Max qlen for pending (SYN-RCVD) requests limits the damage: max qlen = max CPS * average RTT
    - Need to treat RST differently

Amplified Reflection Attack

- The previous two issues are addressed without TFO cookies
- Defense against amplified reflection attack from a large # of servers
  - TFO cookies to prove IP ownership, or
  - Defer the app reply until 3WHS finishes, or
  - Only allow one pkt worth of data to be returned before 3WHS finishes
  - Both may reduce the benefit of TFO

Can Cookie be Made Optional?

- Only if an TFO server ascertains it poses no risk for an amplified reflection attack
  - E.g., the server knows its response size fits in one pkt
- More details need to be worked out
  - Is it worth the trouble?

Sending Data in SYN-RCVD State

- Current implementation responds SYN+data with SYN/ACK acking SYN+data right away
  - Server response data have to go out in separate pkts
  - SYN/ACK could be delayed to catch response data; save one pkt just like delayed ack
  - Don't include data in SYN or SYN/ACK retransmits to avoid problem with middlebox

Client Side - Data After SYN

- To accommodate request size > MSS
- But ACK flag will be off
  - Is this even a legal TCP pkt w/o the SYN bit? (Doesn't seem so according to RFC793, section 3.1)
- Many ISPs drop non-SYN pkts w/o ACK flags
- Current implementation limits data to only 1MSS
  - More data will have to wait after SYN is acked

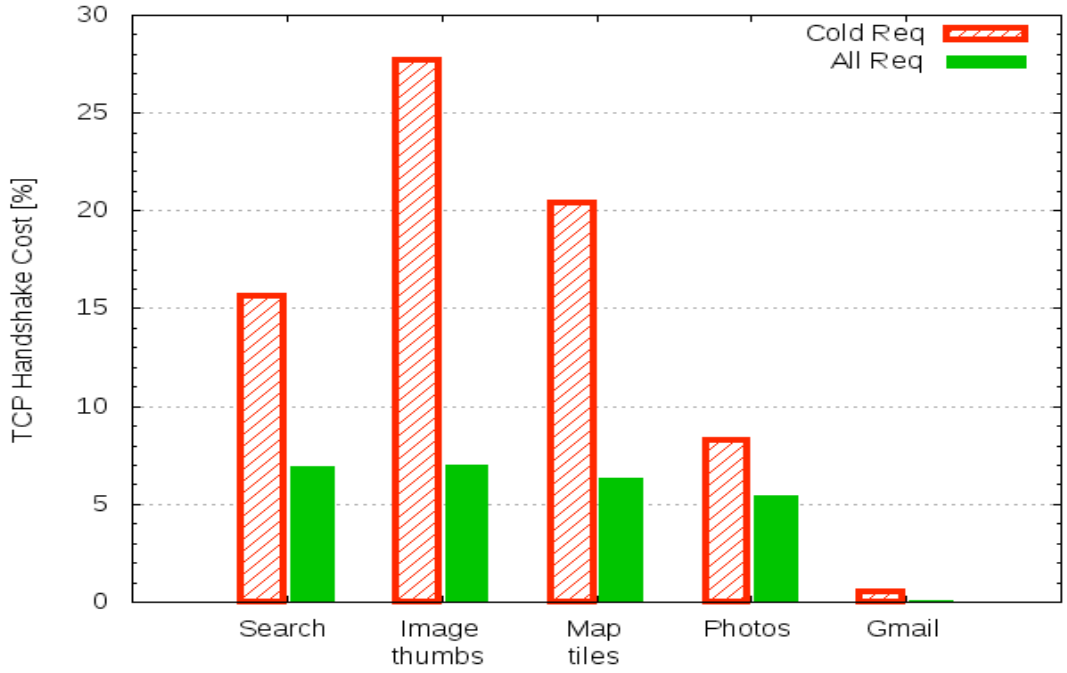
New State Transitions

- What to do if SYN-SENT socket with unsent or unack'ed data is closed or half-closed?
  - Is SYN/Data/FIN (xmas tree) allowed?
- What do do if SYN-RCVD socket with unsent or unack'ed data is closed or half-closed?
  - Is SYN/ACK/Data/FIN allowed?
- Kamikaze pkts may be problematic (RFC1379)
  - Not welcomed by IDS
  - Only reduces pkt count, not round trips

New API for TFO

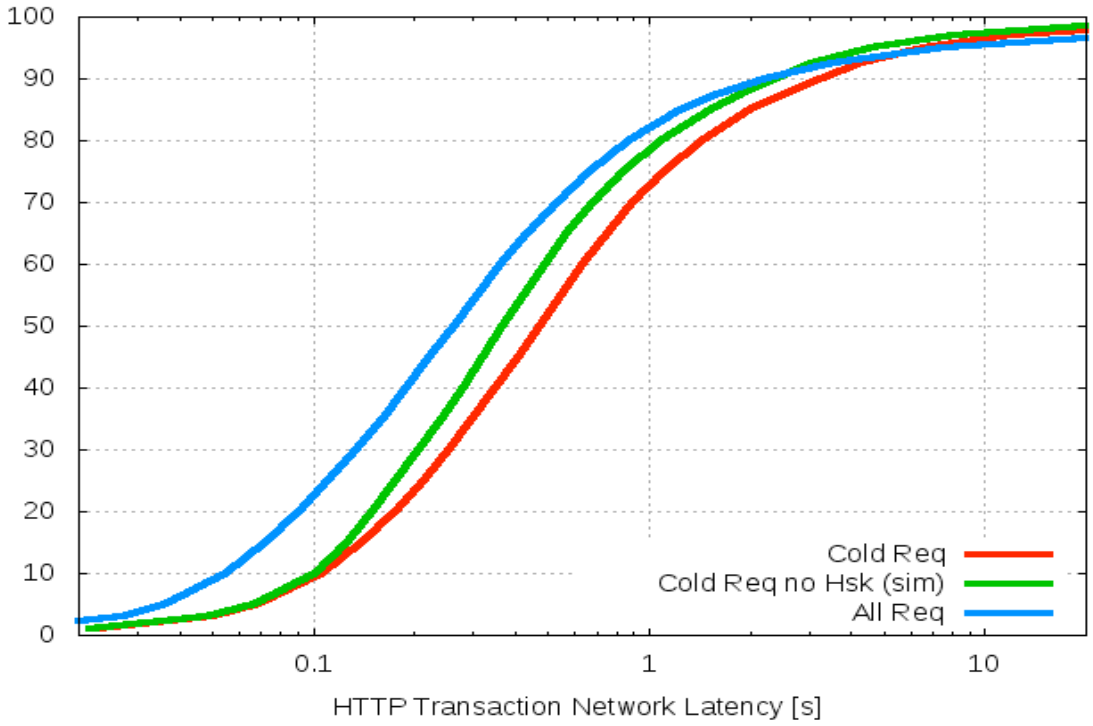
- Client side
  - Employs sendto()/sendmsg(), which already allows dest IP/port as an argument
- Server side
  - New “TCP_TFO” socket option to enable TFO on a per listen port basis
- TFO cookie is handled completely within the stack, transparent to the apps

Handshake Overhead (seen by server)



TCP handshake accounts for 8% to 28% latency for major Google services except Gmail

Handshake overhead (seen by browse)



TCP handshake costs 25% latency of cold HTTP requests

Stats from Chrome users who opted-in for stats in June 2011

Whole Page Download Performance

Page	RTT(ms)	PLT : non-TFO (s)	PLT : TFO (s)	Improv.
amazon.com	20	1.54	1.48	4%
	100	2.60	2.34	10%
	200	4.10	3.66	11%
nytimes.com	20	3.70	3.56	4%
	100	4.59	4.30	6%
	200	6.73	5.55	18%
wsj.com	20	5.74	5.48	5%
	100	7.08	6.60	7%
	200	9.46	8.47	11%
TCP wikipedia page	20	2.10	1.95	7%
	100	3.49	2.92	16%
	200	5.15	3.03	41%

TFO can reduce the overall page load time (PLT) up to 41%, especially in high RTT networks, e.g., mobile

Related Proposal

- TCPCT's Accelerated Open, Rapid Restart
- Design for SYN flood defense, saving server state, not for carrying data in SYN
  - For DNSSEC
  - AO & RR were added later
- Different focus (security vs performance)
- Substantial complexity (large cookie size requires header extension)

Related Proposal (cont')

- Vastly different cookie protocol and semantics
  - TCPCT's 2-way cookies serve to prove connection authenticity to the server (the final ack does indeed come from the connection making the original request), hence involves a lot more complexity
  - TFO's server-only cookies only need to prove source IP ownership (i.e., the source IP likely not spoofed)

Related Proposal (cont')

- With server stateless, AO requires
  - App to consumes and produce response data to be carried in SYN/ACK
  - Response limited to a single SYN/ACK pkt
  - Relies on client to retransmit
- RR seems to claim multi-pkt support (??)
  - TCB retention defeats the original design goal (no longer stateless, also why close the connection then?)

Implementation Status

- Linux 2.6.34 based prototype completed and tested on the Internet (through Comcast, AT&T)
  - ~3000 lines of code changes
  - Chrome browser was enhanced to use sendto() for testing
- Plan for production release soon
- Need a new TCP option number from IANA for TFO Cookie

Question?