



xWAMP extensions

Access

Yaakov (J) Stein
RAD Data Communications, Ltd.



data communications

xWAMP extensions

I would like to discuss 5 extensions to OWAMP/TWAMP :

Control protocol extensions

- Multiplier
- Packet size specification

Test protocol extensions

- TCP test sessions
- Handling multipath
- Throughput measurement

Multiplier

Presently one can specify only 1 test packet per slot

This is inefficient for complex specifications

Many scenarios would be efficiently encoded

by adding a multiplier to specify the number of packets

Thus the schedule slot information would include:

- Slot type – fixed / exponential
- Multiplier – number of packets
- Time parameter

Thus to encode bursts of 50 packets separated by 1 second

one would encode

- FIXED 0 1 second
- FIXED 50 0 seconds

Packet size

For the throughput estimation discussed later,
it is useful to specify the packet size (in bytes)
The combination of packet size and interpacket delay
determines the consumed bandwidth

Thus the schedule slot information would include:

- Slot type – fixed / exponential
- Multiplier – number of packets
- Time parameter
- Size parameter

Thus to encode bursts of 50 packets of increasing sizes
(128, 256, 512, 1024 bytes)
separated by 10 seconds, one would encode

- FIXED 0 1 second 0B
- FIXED 50 0 seconds 128B
- FIXED 0 1 second 0B
- FIXED 50 0 seconds 256B
- ...

TCP test sessions

Many networks treat TCP and UDP packets differently
and so measuring performance using UDP
does not reliably predict performance of a TCP stream

It would be useful to allow TCP test sessions

This would also enable measuring layer-4 parameters
presented in draft-ietf-ippm-tcp-throughput-tm

Handling multipath

SP networks frequently exploit multipath

- LAG
- ECMP

which can be assumed to choose paths based on header hashing

They need a mechanism to probe each path separately

This can be accomplished by using multiple source ports

Although we can certainly use separate test sessions for this
it might be better to coordinate these

Open questions :

- how do we know how to set ports to ensure different paths ?
- should we interleave packets in time ?
- should the results be combined, and if so – how ?

Throughput measurement

After packet loss, delay, and PDV

the next desired performance characteristic is throughput
(IP layer network capacity)

xWAMP can be used for RFC 2544-style throughput measurement
but on live networks (in-service) this is very disruptive
(starts blasting at maximum rate and reduces until no packet loss)

Numerous other methods have been proposed

pathChirp, pathLoad, pipechar, nettimer, Spruce, Yaz, AGI, PTR,
ASSOLO, TOPP, MoSeab, ...

some of which minimize disturbance

The state-of-the-art mechanisms

attempt to find the weakest link (queue closest to overflow)
by measuring delay increase when approaching capacity
(something that xWAMP does well!)

Throughput measurement (cont.)

In order to minimize intrusiveness
burst are separated by sufficient silence

What extensions are needed ?

- need to send bursts of at given bandwidth
this requires control over packet size and interval
- multiplier and packet-size extensions are useful
(although separate sessions per burst may be better)
- it requires knowing how much traffic is already flowing
and generating the correct difference bandwidth
- it requires fetching the delays *and bandwidth* received
- for TCP throughput
the TCP test session extension is also needed