SEC-DIR Review & Discussion: “Test Protocol for One-way IP Capacity Measurement”

draft-ietf-ippm-capacity-metric-protocol-02

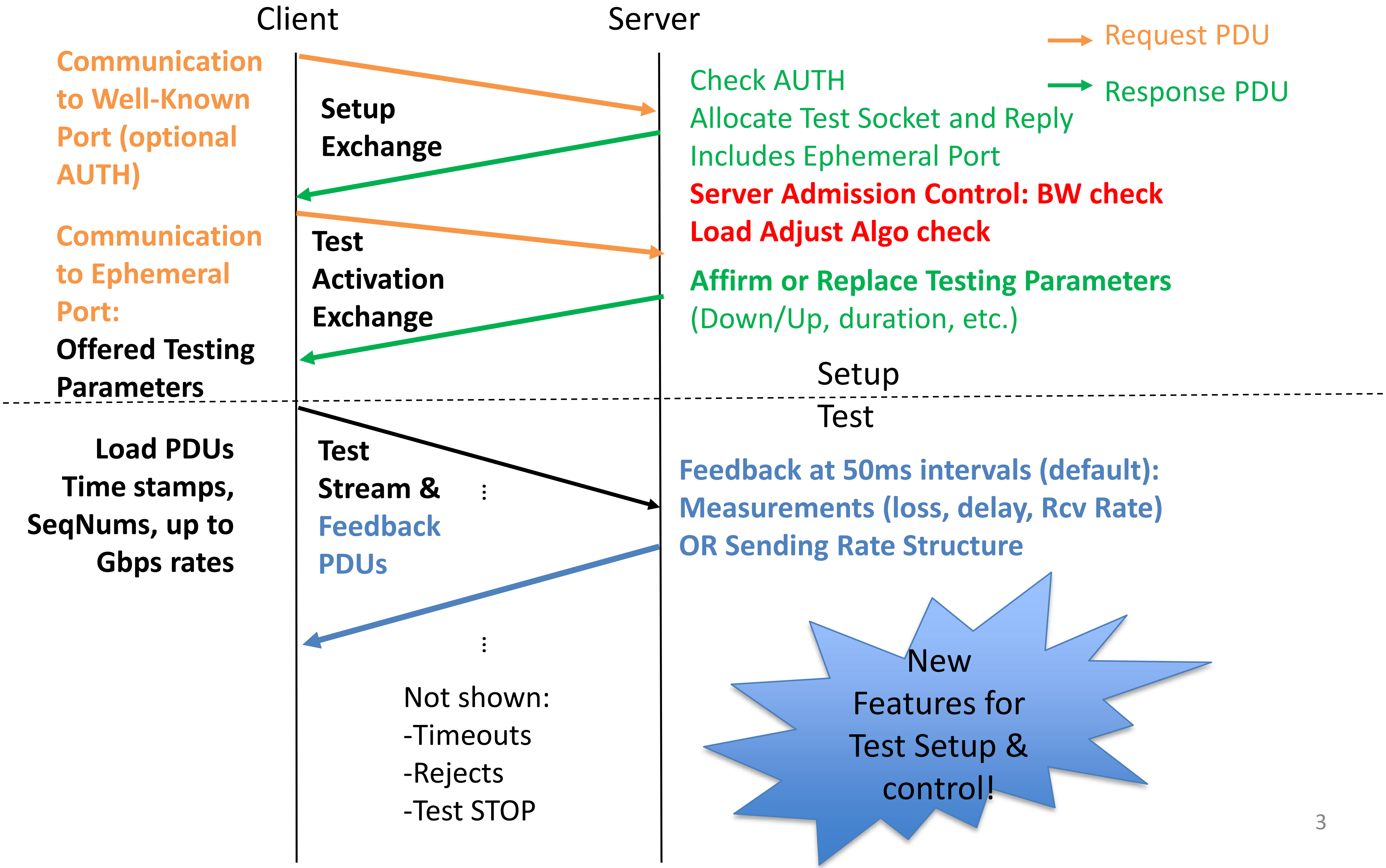
L. Ciavattone, A. Morton

SEC-DIR Review (thanks Brian Weis!) New features & comments, mostly resolved:

- There are two categories of changes needed: text clarifications alone, and text+protocol modifications.
- Implemented the clarifications in version -02
- We use a conventional communications setup, with a well-known port at the server.
- We appreciate your observation that the Authenticated mode can be expanded to use the authDigest field to achieve important message protections and features like bit error checking.
- We have implemented text clarifications in the working text for -02.
- FOUR security modes of operation:
  A. Un-AUTH,
  B. password AUTH(coded)
  C. AUTH “all important messages”
  D. “Encrypt All The Things”
- Need one more recommendation, mode !
- The main comment where we are looking for additional feedback is your comment number 3):
- Is a fully-encrypted mode of operation in IETF Stds track protocols REQUIRED, and this mode must be the default mode operation?
  - We will gladly implement a strong recommendation from you/SEC-DIR in the protocol specification.

Protocol: Setup and Test Phases (v9)

draft-ietf-ippm-capacity-metric-protocol-01



Communication to Well-Known Port (optional AUTH)

Check AUTH
Allocate Test Socket and Reply
Includes Ephemeral Port
Server Admission Control: BW check
Load Adjust Algo check

Communication to Ephemeral Port:
Offered Testing Parameters

Affirm or Replace Testing Parameters
(Down/Up, duration, etc.)

Load PDUs
Time stamps,
SeqNums, up to
Gbps rates

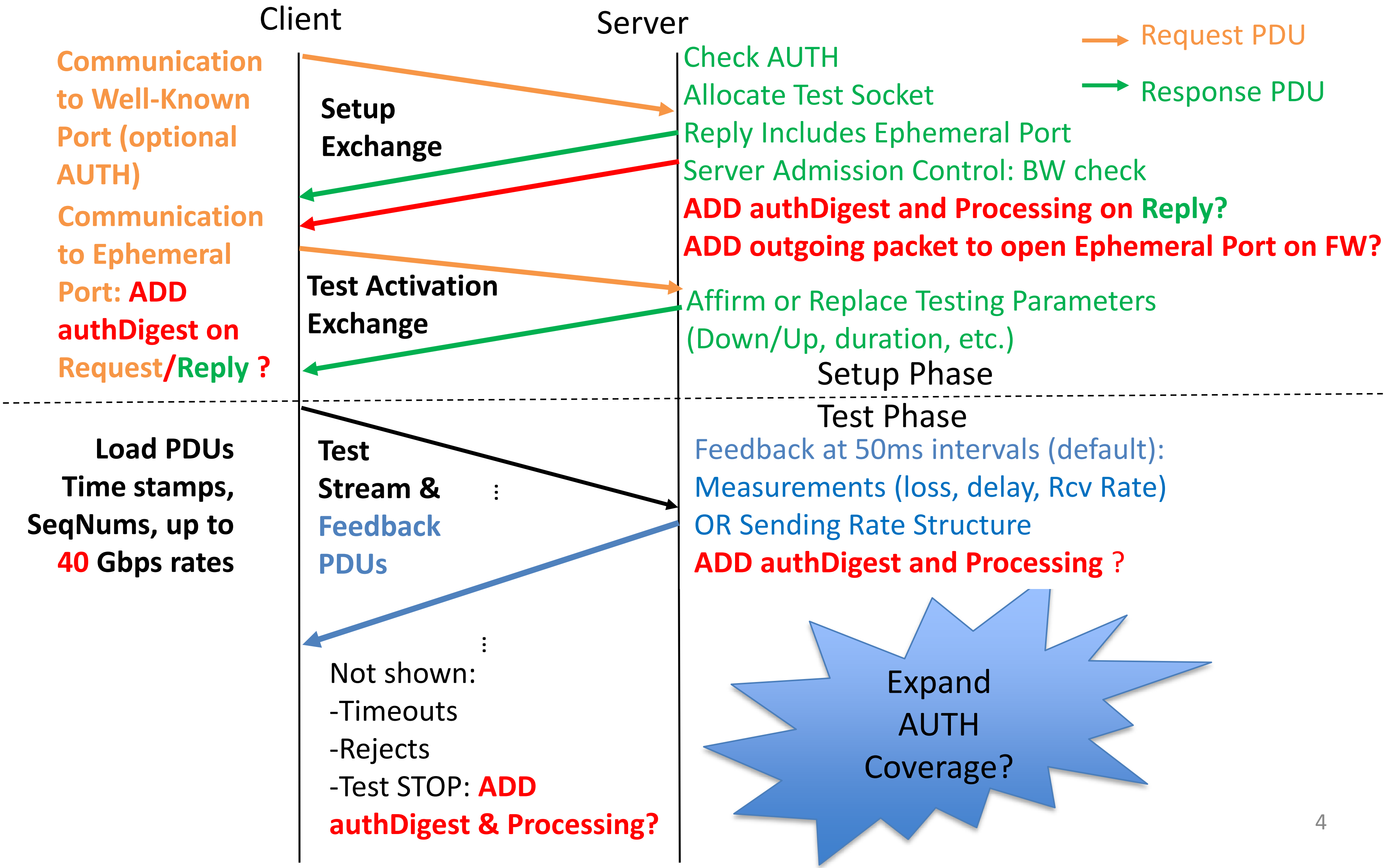
Feedback at 50ms intervals (default):
Measurements (loss, delay, Rcv Rate)
OR Sending Rate Structure

Not shown:
-Timeouts
-Rejects
-Test STOP

New Features for Test Setup & control!

Protocol: Setup and Test Phases

draft-ietf-ippm-capacity-metric-protocol-next



Client

Server

Request PDU

Response PDU

Communication to Well-Known Port (optional AUTH)

Communication to Ephemeral Port: ADD authDigest on Request/Reply ?

Setup Exchange

Check AUTH

Allocate Test Socket

Reply Includes Ephemeral Port

Server Admission Control: BW check

ADD authDigest and Processing on Reply?

ADD outgoing packet to open Ephemeral Port on FW?

Test Activation Exchange

Affirm or Replace Testing Parameters (Down/Up, duration, etc.)

Setup Phase

Test Phase

Load PDUs Time stamps, SeqNums, up to 40 Gbps rates

Test Stream & Feedback PDUs

Feedback at 50ms intervals (default): Measurements (loss, delay, Rcv Rate) OR Sending Rate Structure

ADD authDigest and Processing ?

Not shown:
-Timeouts
-Rejects
-Test STOP: **ADD authDigest & Processing?**

Expand AUTH Coverage?

Additional Comment Exchanges:

- Firewall operation (both ends)
  - At Client: Control and Data exchanges originate at Client, OK!
  - At Server: Could avoid opening an Ephemeral port range, extra message needed on Ephemeral port – seems likely to work

- Need to look at re-organizing the *Modes of operation*
- One possible set:
  - REQUIRED AUTH for Control msgs:
    - Test Setup exchange
    - Test Activation exchange
  - OPTIONAL AUTH for Data msgs
    - maybe only the Status Feedback messages
  - OPTIONAL Encryption of Setup messages,
    - maybe Activation messages too,
    - maybe use DTLS and
    - maybe re-use keying for AUTH aspects
  - OPTIONAL Un-AUTH Mode

Additional Comment Exchanges:

- Add and process authDigest on <u>all messages:</u>
  - Possible/useful for most ...
- but prefer Not adding to Load PDU:
  - Only "info" is the STOP1 and STOP2 bits
  - If Attacker clears the STOP bits, Tests stop anyway after specified duration (3 sec time-out)...
  - If Attacker adds STOP, premature end of test but no threat to Internet.
  - Adding SHA-256 significantly increases the minimum packet size.

- Add and process authDigest on <u>all messages:</u>
  - Possible/useful for most ...
- Status Feedback PDU: the info is
  - either the measurements collected during the previous interval, or
  - the new Sending Rate for the client based on measurements at the server.
  - also used for sampled RTT measurements, so this is both a control and a data plane message
- So, it seems valuable to protect the control info, but need to keep the RTT measurements in mind.

Comments:

- Key Management
  - Manually configured Keys (now), one key per server instance
  - Ref to RFC 7210 (DB of long-lived keys)
  - Add Key Identifier?
    - Need to know both Key and ID
    - We don’t have a config file.
  - Or, Add a section on orderly Key Rollover.
    - Add text to describe this

- Suggestion: Use DTLS for Confidentiality in the Setup Phase
  - Adds retransmission and ordered delivery
  - Still need to activate and open FWs the ports needed for testing
    - Dummy from the Server, might not make it to Client
    - Dummy packet from Client before the Load PDUs
    - Need Dummy packet exchange?
    - Need time to wait for Load PDU start?
  - ? Derive keys from DTLS session to use with SHA-256 HMAC during the Test Phase ?
    - Feedback messages ONLY
    - ? Support in OpenSSL ?

Comments:

Need a different logic tree for Silent Rejection during Setup Phase:

- Unauthenticated Mode:
  - Silent Rejection
- Authenticated Mode with Successful Validation:
  - Return Rejection Msg with error code
- Authenticated Mode with Failed Validation:
  - Silent Rejection
- Compile-time Server-option: Non-Silent Rejection (always) for troubleshooting

- Client does not currently validate the Server's Setup Response in v9 of the running code.
  - Expanding use of authDigest to the entire Setup Phase would fix this.
  - Need to be sure that authDigest checking lists are correct/fixed when using a more complete AUTH mode

- authUnixTime is not a complete protection against replay attacks:
  - ? Add record of previously received messages within that window ?
  - Add ID ? Can't be replayed with same HMAC
  - (but we're Not NORAD)

Mode D: Encrypt “all the things”, Brian’s safe advice

(1) Define a method for strong authentication of all messages. SHA-256 HMAC is a good choice.

(2) Make that method required to be implemented for all messages, although it could be optional for a site to deploy it on all messages.

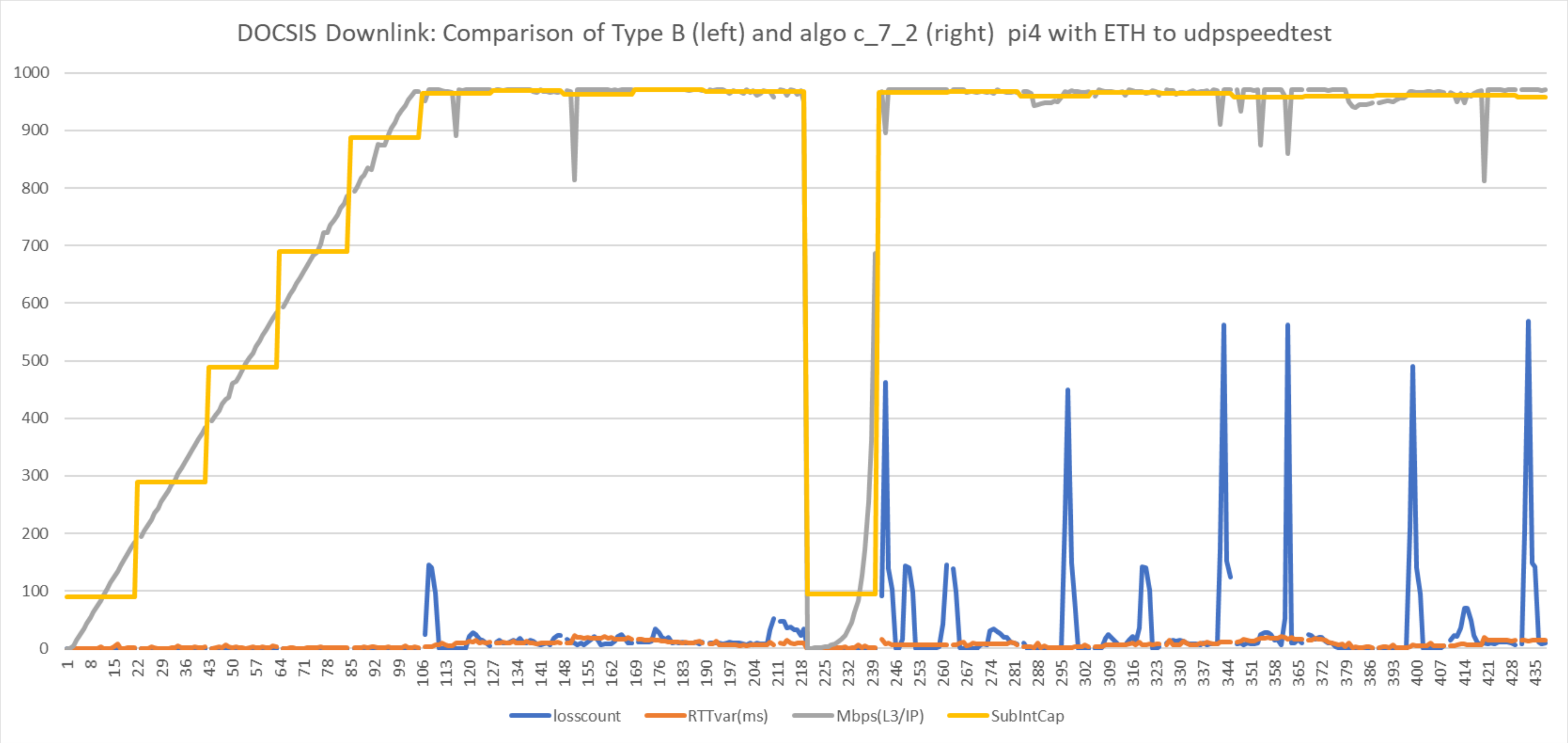
(3) Require authentication of test setup messages (i.e., Setup Request/Response, Test Activation Request/Response), except possibly for diagnostic purposes. This seems defensible to me if they are “control plane” protocols, for which the added processing for authentication is feasible.

(4) Make authentication of “data plane” messages optional. This seems defensible to me, when accompanied with an explanation that the process of adding authentication to test messages can impact the test result accuracy.

(5) Since the devices performing measurement are network devices with constrained processing and operations, the required method will likely use manually configured keys. Provide for an orderly key rollover by including key ids in the PDUs for the authentication keys, This will be helpful for both security and operations of the protocol.

(6) Consider whether DTLS can be used as an option for the Setup Request/Response exchange, and possibly add extract keying material from that exchange using RFC 5705 for use of SHA-256 HMAC for the other exchanges.

Comparison of Current (Type B) + New Type C algos


DOCSIS Downlink: Comparison of Type B (left) and algo c_7_2 (right) pi4 with ETH to udpspeedtest

1Gbps Downlink Service
Two 10 second tests in series
UDPST 7.5.0 in Debug mode

50ms status feedback meas:
Packet Loss counts
IP-Layer bit rate
RTT Range

1 second Sub-Interval meas:
IP-Layer Capacity
set for finding Max Capacity

Next Steps:

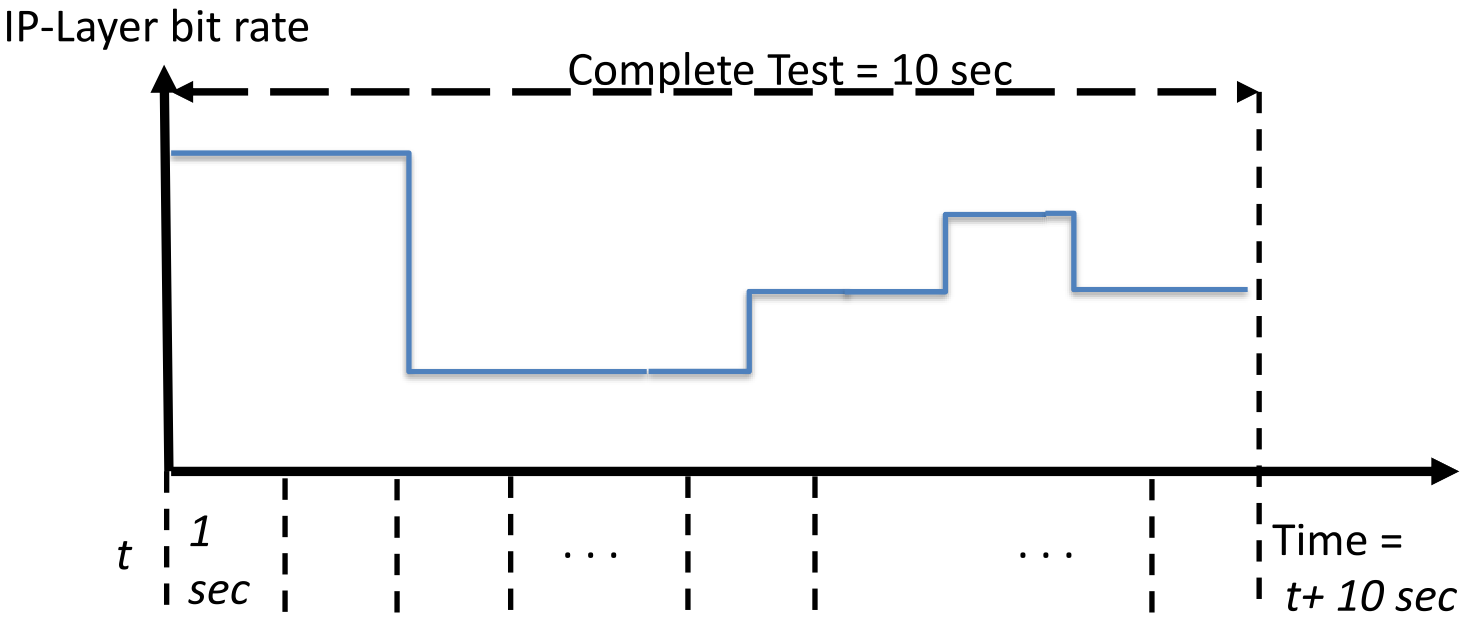
- Reviews, test experiences, proposals, comments, etc. welcome
- Implementations are happening!

Reminders/Review Areas:

- Protocol ver 9 allows for New Load Adjustment Algorithm(s)
  - Describe in Draft?
  - Designate New Default Algo?
- Send Rate Table expanded to 40 Gbps in UDPST 7.5.0
- This protocol permits Latency measurement, reordering, etc.
- could do more than measure Capacity – take a look at IETF-133 slides! (one in backup)

BACKUP

Alternate Forms of Rate Programming


IP-Layer bit rate
Complete Test = 10 sec
t
1
sec
Time =
t+ 10 sec