ns-3 IETF hackathon plan

Dates: July 20-21, 2019 (remotely attending Montreal meeting)

Times: Saturday (all day), Sunday morning? (Canada Eastern Daylight Time)

Registration link: https://www.ietf.org/how/runningcode/hackathons/105-hackathon/

Zoom room for L4S: https://cablelabs.zoom.us/j/195175655

Slack channel for L4S: https://l4s-workspace.slack.com/

ns-3 chat room: https://ns-3.zulipchat.com

ns-3 project participants

- Tom Henderson (times available TBD, probably mainly Saturday morning/afternoon)
- Mohit Tahiliani
- Shravya KS
- Vivek Jain
- Viyom Mittal
- (others TBD)

Drafts and references:

- **L4S architecture:** https://tools.ietf.org/html/draft-ietf-tsvwg-l4s-arch-04
- **DualQ:** https://www.ietf.org/id/draft-ietf-tsvwg-aqm-dualq-coupled-10.txt
- **DCTCP:** https://www.rfc-editor.org/rfc/rfc8257.txt
- **Paced chirping:** https://datatracker.ietf.org/meeting/102/materials/slides-102-iccrg-flow-start-faster-and-less-overshoot-with-paced-chirping-00
- **AccEcn:** https://tools.ietf.org/html/draft-ietf-tcpm-accurate-ecn-09
- **NetDev 2019 talk:** Implementing the 'TCP Prague' Requirements for L4S: https://www.netdevconf.org/0x13/session.html?talk-tcp-prague-l4s

ns-3 background and goals for this hackathon

Several ns-3 projects over the past few years have been working on L4S/Prague components. Most are not yet in the mainline; some are out of date with respect to the tip of ns-3-dev.

The goal of this hackathon is to further develop the models needed to simulate L4S with ns-3. Primarily, this means:

1) a DCTCP implementation aligned with the latest Linux code (there are several patches around to fix things in Linux DCTCP so the ‘latest’ is unclear and should be clarified this weekend)

2) the start of a TCP Prague congestion control implementation; TCP Prague will differ from DCTCP in at least the slow start behavior (possibly paced chirping) but possibly other ways as well

3) an updated DualQ model current with the latest draft

4) AccECN and ECN++

5) some evaluation scenarios to test behavior and performance

There is no complete TCP Prague Internet-Draft, but the TCP Prague requirements and optimizations are (from Briscoe’s NetDev slides):

Requirements

- L4S-ECN Packet Identification: ECT(1)
- Accurate ECN TCP feedback
- Reno-friendly on loss
- Reno-friendly if Classic ECN bottleneck
- Reduce RTT dependence
- Scale down to fractional window
- Detecting loss in units of time

Optimizations

- ECN-capable TCP control packets
- Faster flow start
- Faster than additive increase

ns-3 code location

Let’s use or create the following branches. Participants should fork ns-3-dev as usual from GitLab, or clone, and add this repository as a remote:

```
$ git remote add tomhenderson https://gitlab.com/tomhenderson/ns-3-dev.git
$ git fetch tomhenderson
```

and then branch from the various branches there:

- tomhenderson/dctcp
- tomhenderson/ecn
- tomhenderson/dualq
- tomhenderson/prague

If instead someone else wants to host the canonical branch for a feature, that would be fine also.

Send merge requests in the usual way to my repo (push your branch with changes to your fork, and then create merge request towards my repo). Here is what needs to be done for each branch.

1) DCTCP
Based on Shravya's code, a few people (Ankit, Aditi, Shikha, and Apoorva) have been working on it.

Some DCTCP things that need to be done.

1a) alignment with Linux due to its bug fixing

DCTCP has some bugs in Linux that are in the process of being patched, and we should get alignment and some review from Koen and Bob (or others on L4S team).

Linux bug for last 4 years had no ssthresh reduction on loss
DCTCP cwnd decrease on drop
TCP over-strict ECN negotiation
DCTCP EWMA gap in precision
DCTCP non-default parameter reveals hard-coded values
DCTCP insufficient EWMA precision
DCTCP window reduction sticks

1b) ECT(0) or ECT(1) requirement (option to use either, but shouldn't use neither)

We should prevent DCTCP from being used without either ECT (0) or (1).

2) AccECN

Need to review and update WenYing's latest GSoC 2018 code. This also includes ECN++.

3) DualQ

Shravya's code from GSOC 2017 must be ported to ns-3-dev, and not updated to the latest draft. The port involves making some lower-level changes to the queue sizing based on API changes to the base queue classes introduced since then.

4) Prague

Joakim Misrund sent a patch for Paced Chirping but I am waiting for his confirmation to open source it.

We can make a TcpPrague class starting from TcpDctcp, and then add chirping once Joakim concurs, and start to add other things.

Fallback behavior when Prague takes an actual loss needs to be defined and tested.

5) Scenario support

Need to write down a methodology for thoroughly testing Prague and L4S, and start to implement the test scenarios. We can try to reproduce the scenarios and methodology that the L4S team has defined for their testbed experiments.