Agenda

- Introductions and agenda bashing
- Status on strongly-typed units
- Policies on use of coding agents
- How to promote smaller MRs for easier reviews?
- Plan/organization for triaging our issues and MRs
- General roundtable on roadmap/priorities (what does each maintainer want to prioritize for ns-3.48 and beyond?)
- (insert any other topics proposed, time permitting)

Status on strongly-typed quantities and units

- We have been discussing since 2023, and we now have three viable library candidates:
  - 'si-units' custom library from Jiwoong and Sebastien
  - 'nholthaus/units' port
  - a new 'mp-units' port
- Only recently have coding agents become good enough to help with the drudgery of this porting task
  - Claude Code did a pretty good job of porting nholthaus/units ns-3 support to mp-units
- I now favor using mp-units and favor trying to port ns-3 wifi and possibly more libraries over the next couple of weeks
  - Needs review and concurrence (or counter-proposals) from Peter (core), Gabriel (build), Wi-Fi maintainers, and others (LENA project?) with a stake in this decision

Policies on use of AI and coding agents

- I am a proponent of using these tools and think that they will accelerate ns-3 development and utility
  - However, I understand and respect other opinions in open source communities about whether and how to use these tools
- The landscape is changing rapidly; we need to signal to agents, and to contributors using agents, how we want them to participate with these tools
- For now, I have posted a proposed AI policy, and proposed initial AGENTS.md
  - Ready to merge?

- Do not review for more than 60 minutes at a time - your efficiency starts to drop off heavily

How to encourage (require?) smaller MRs

- Large MRs languish because they are costly to review
  - This may only get worse over time with coding agents
- Smart Bear study suggested
  - 400 lines of code or less per pull request
  - require no more than 60 minutes of time (before attention drops off)
- We tried GitLab.com merge trains and MR dependencies but did not like the experience
- I half-jokingly proposed in a recent discussion about adding a CI check to fail if a MR exceeded a certain size
  - this failure could be overridden at merge, but might disincentivize large MRs
- For large proposals, we could ask for contributors to generate small MRs but link to a larger complete branch that has the big picture

Plan/organization for triaging our issues and MRs

- There have been proposals in the past to close stale issues/MRs, but could not gain consensus
- Most recent agreement (from April 2023)
  - Mention persistent open issues and feature requests in Sphinx documentation (need to avoid the case in which documentation falls out of line with tracker)
  - Only started to do so last month (wifi module)
- Many issues are poorly specified with requirements, or not agreed, leading to unfortunate outcome that a new contributor brings a MR that is rejected
- We could use coding agents to work on this backlog, if maintainers have enough attention span to review their output
- Thinking about a semi-regular triage meeting (virtual) to work on our backlog
- Also thinking about new tags: 'dormant', 'agent-only', 'human-only'

My ns-3.48 priorities

- GSoC merges (6LoWPAN ND, satellite orbital mobility)
- Units conversion if we come to agreement
- Wi-Fi powersave review/merge
- Review DCE patches and try to revive that with a release if ready
- Otherwise, work on existing backlog
- ns-3-dev libraries coming and going (ns-3.48 or ns-3.49 timetable)
  - move some additional modules out of ns-3-dev to a nsnam-hosted repository, and include in the app store and ns-allinone releases
    - candidates: brite, click, openflow, topology-read, uan
  - could also, in the future, migrate some stale models within existing modules (e.g., a 'tcp-extra' or 'wifi-rate-controls' repository for models that are not as relevant anymore)
  - could move some from outside of ns-3 into ns-3 (sip?, lorawan?)