

Landing Page

Welcome to the SIGCOMM Education Workshop!

Quick Links:

- Workshop Webpage: <https://a4ne-workshop.github.io/>
- Schedule at a Glance:
- Papers: <https://a4ne-workshop.github.io/papers.html>
- Workshop Registration: <https://forms.gle/9i7Yx4fdDz2b8ymHA>
- Zoom link:
<https://swarthmore.zoom.us/j/88141139200?pwd=YN2XgkFvY29bfK2Nh8au3j2P0DAg3M.1>

Interested in an Open-Mic Session to share your hot-take? [Please consider signing up here!](#)

Interested in exploring a topic further? [Suggest a BoF for the Break-out Sessions!](#)

[Birds of a Feather \(BoF\) Break-Outs!](#)

Open Mic Session

Open-Mic Slots!

Please stay within a five minute slot so we get to hear a diverse array of perspectives!

Time	Slot	Name	Title	Virtual/ In-Person
1:20 PM	1	Ram Rustagi	Experiential Learning approach: How to scale up the assessment	In Person
1:27 PM	2	Fraida Fund	Doing bigger networking projects with less friction using AI coding agents + research infrastructure	Virtual
1:34 PM	3	Kapil Mehta	Course Innovation :- Collective Communications and New Transport Protocols for "Networking for AI"	In Person
1:41 PM	4	Koojana Kuladinithi	To evaluate students autonomy (ability to reason critically and remain responsible for what they do), are oral examinations the best way to grade? Any other alternatives ?	In Person
1:48 PM	5	Pamela Zave	Teaching abstraction.	In Person
1:55 PM	6	Tess Despres	How to train to become an educator in networks	In Person
2:02 PM	7	Shiv Panwar	Teaching students not necessarily interested in networks as a career	Virtual
2:09 PM	8	Karen Sollins	Outrageous opinion: Eliminate lectures: lessons from law school and intro to programming	Virtual
2:16 PM	9	Aaron Gember-Jacobson	Alternative grading practices	In Person
2:23 PM	10	Thomas Zinner	From VMs to Lab as Code and the Quest for Tooling for Students and Educators	Virtual
2:30 PM	11			
2:37 PM	12			
2:44 PM	13			

Birds of a Feather (BoF) Break-Outs!

BoF Break-outs!

BoF breakouts are designed to explore topics of your interest in Network education more deeply!

Structure:

- During the coffee break beforehand, we'll organize BoFs (birds of a feather) so each topic has a home room by the time breakouts start
- 3:15–3:25: everyone settles into their breakout room, in-person or virtual
- 3:25–4:25: breakout discussion (one hour)
- 4:25–4:45: everyone reconvenes for a five-minute quick share/debrief per room

We're aiming for 4–5 breakout rooms so we can wrap up with a 20-minute debrief at the end. Each topic will run as two parallel rooms, one in-person, one virtual — covering the same subject at the same time, not co-located but simultaneous. We recommend each room (in-person and virtual) have a scribe tracking what was discussed, to share a 5 minute summary at the end.

Breakout Room 1: Title:What should remain at the core of networking education? Which traditional topics should we keep, and which should we rethink?

Topics of interest: What should remain at the core of networking education? Which traditional topics should we keep, and which should we rethink?

...both for a core “foundations” class and for a multi-course curriculum sequence?

Virtual Participants: (Name, Institution, Role)

* Musab Ahmed, Hamburg University of Technology, Research Fellow

In-Person Participants: (Name, Institution, Role)

*

Virtual Participants Scratch Space:

- Abstractions as a main focus
- Don't focus on details and start with practical examples —< even a complete course can be structured around it
- TCP is central, but it shouldn't be the only focus (QUIC is also there) → Reliable Transmission stands behind it (asan example of a central idea)
- Mental Models/the problem solution model/central principles are possible approaches to paedagogically structure a course

- regarding the future evolution (and the corresponding relevant topics) the test of time is a good indicator

In-Person Participants Scratch Space:

Breakout Room 2: Title: Integrating national networking infrastructure (PAWR, NRP testbeds etc.) into networking curriculum for hands-on (AI-assisted) projects

Topics of interest: Integrating national networking infrastructure (PAWR testbeds, NRP etc.) into networking curriculum for hands-on (AI-assisted) projects ...strategies, testbed options, onboarding issues, reuse (as pointed out by one of the panelists), step towards experiential learning

Virtual Participants: (Name, Institution, Role):

- Venkat Sai Karanam, Bowling Green State, Asst. Prof. (CS)
- Zahmeeth Sayed Sakka, WVU - Tech | WVU, Asst. Prof (CS&IS)
- Fraida Fund, NYU
- Ilknur Aydin, Farmingdale State College
- Ranysha Ware, Swarthmore

In-Person Participants: (Name, Institution, Role)

-

Virtual Participants Scratch Space:

- Students encouraged to reserve resources ahead of timeProject-level reservation (COSMOS, chameleon, POWDER)
- **Bridge the gap between theory and practice**
- **Engaging with real infrastructure vs pure coding**
 - “Closer” to what they may work with post-graduation
 - Currently, AI coding agents can do all lab assignments – even those based on infrastructure.
 - In terms of cheating – it is hard to AI-proof even these
 - Open question: even if infrastructure is used we cannot make sure that students may rely excessively on AI assistance; more importantly, potential for students can automate and bypass the learning goal
- **Exchange of teaching delivery materials across institutions**
- Capstone projects enabled by infrastructure – pathways for students outside networking but not majoring in it to get familiar
 - Examples: those overlapping with SE/security
- MERIF: <https://merif.renci.org/merif26-meeting-information>

- **Connecting AI tools or agents directly to the testbed/infrastructure can help them do a lot of cool things**
- Question: Biggest barrier to testbed adoption
 - Ugrad (for capstone) or non-networking discipline are highly unfamiliar – so may struggle initially getting their hands dirty
- Recommend students to subscribe to newsletter – events, trainings, workshops
 - Tailored towards researchers, which may be more than what an undergrads need to be effective for their goals
- **Two use cases: (1) to learn the infrastructure (2) to use the infrastructure to achieve a separate learning goal**
- Grading weightage – can treat means to an end (if materials are provided)
- *If something breaks or has the potential to break – do not include it*
- Vibe coding within the infrastructure – agent runs within the testbed
- Financial constraints – may(?) run through tokens quickly, since compute is not (normally) a limit
- Equity in resource availability
- **Open models need to be encouraged**
 - Cost constraints – no free meal
 - May even train on student data
 - Students morals may not align with commercial LLM agents
 - What is the option there?
 - SIGCSE survey reports on LLM use for courses – available
 - [Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools | 2024 Working Group Reports on Innovation and Technology in Computer Science Education](#)
 - Rest of the Robots
 - We Need More Reproducibility

Resources to share (from Fraida):

- Experiments for networks/wireless: <https://witestlab.poly.edu/blog/>
- Artifact that launches a compute environment with OpenCode AI coding agent on Chameleon Cloud
<https://trovi.chameleoncloud.org/dashboard/artifacts/d0cb594d-dd07-436f-85bd-50b09d9b7230>
- Artifact that launches a compute environment with OpenCode AI coding agent on CloudLab (Requires CloudLab account)
<https://www.cloudlab.us/p/nyunetworks/jupyter-opencode>

In-Person Participants Scratch Space:

Breakout Room 3: Title: Rethinking Assessment

Topics of interest: oral assessment, in-class vs out-of-class assessment, scaling to “large” classes, implementation/measurement assignments vs conceptual exams, team based assessments

Virtual Participants: (Name, Institution, Role)

- Mary Hogan, Oberlin College, Assistant Professor of CS

In-Person Participants: (Name, Institution, Role)

- Aaron Gember-Jacobson, Colgate University, Associate Professor of CS
- Koojana Kuladinithi, Hamburg University of Technology, Senior Researcher
- Thea Rossman, Stanford University, PhD Student
- Kapil Mehta, Cisco

Group scratch-space:

Oral assessments, interviews

- Question raised: unequal preparation for public speaking / oral exams among students.
 - Scaffold throughout the course continuously, not an "all of a sudden oral exam" at the end
 - Similar to "quizzes throughout the course before (or instead of) a final exam" model, but with additional goal of scaffolding public speaking
- "Interview-style", tried to make it fun
 - Need ~half an hour to get to some level of depth, which isn't scalable
- Mix of modalities:
 - Students can optionally "make up" lost points on an in-person exam by coming to an in-person interview

Exams with online components

- Tried proctored online + in-person exam to scale – rampant use of AI tools
- Alternative: allow students to use LLM tools; design the questions so that they can't be trivially answered by an AI tool (or can't answer within reasonable time bounds)
 - Problem: do we then start asking questions that are too complex for a true novice?

AI-assisted assessments as an in-between, a way to scale, or a learning tool

- Feedback from AI system as you go; grading on the quality of the questions you ask it
 - [Grounded in industry perspective] "satisfaction score" as the framework for whether you are asking the right question

- Learning in a sandbox knowing that an AI system is the one that's assessing them, it could make them more open to making mistakes and exploring
- The AI system is not performing assessments, but it can be as a learning tool
- The questions a student asks to the AI system could be a point for assessment
- Industry perspective: as a tool for training / education in industry

Programming assignments

- We broadly feel attached to programming assignments, but we feel like students are getting less out of them. What do we do?
- For assessments: in order to get a certain grade in the class, you have to get that grade on the programming components and the in-class setting
- Measurement assignments can be a bit more "AI-proof" (at least the act of gathering the data) – maybe
- Grading: programming assignments graded on completion and then quizzes on implementation
 - Take what a student submitted, give it back to them in a controlled setting, and ask them questions about it (e.g., document what would happen with X in your own code)
 - Would this scale?
- Should students be expected to code?
 - They should definitely understand what the code is doing
 - Concrete proposal: in-person interaction to assess understanding and facilitate engagement around the code
- Give students broken code generated from AI, ask them to fix it, then ask them to do a writeup of how they got to a working version and reflect on the process
 - Note: harder to do as AI coding gets better
- Ideas brought up in workshop papers (e.g., Despres/Krentsel paper on protocol class at Berkeley) around how to scaffold programming assignments
 - Focus on scientific analysis
- Trying to get students to understand the process they went through (work in the education community about this)
- High alignment that syntax isn't the goal

"Configuration" assignments: set students up with a (virtual) infrastructure, which they have to configure

- Track the revision and iteration process to the configuration

Developing good rapport with students, being someone they want to listen to, is one way to

- If you tell students "you're not going to learn this if you do it using these tools", they might listen. (Maybe – then there's the time crunch.)

Measurement-based assignments

- Explicitly set up the mindset that this is deliberately underspecified and open-ended
- Start with something at a smaller scale with some ground truth as an initial scaffold

- Let them make revisions to get some credit back if they need more scaffolding
- [Not many thoughts on this!]

Some of us have student populations that are very AI-averse (e.g., for ethical and sociopolitical reasons) and really not wanting to use it