Flipped Team-Based Learning and Labs for a Course on Computer Networking*

Thomas Zinner[1], Katrien De Moor[1], Stanislav Lange[1], David Palma[2], and Trond Vatten[1]

[1] Norwegian University of Science and Technology (NTNU), Trondheim, Norway
{thomas.zinner,katrien.demoor,stanislav.lange,trond.vatten}@ntnu.no
[2] Takinobori, Coimbra, Portugal
david.palma@takinobori.com

Abstract. Introductory computer networking courses must help students move beyond memorizing protocols toward reasoning about packet-level behavior, configuration, and debugging. This experience report presents a flipped Team-Based Learning design that integrates weekly preparation, readiness assurance tests, team activities, and reproducible hands-on labs in a 13-week undergraduate networking course. We describe the course workflow, assessment model, and support structure, including an oral exam for individual accountability. Drawing on student feedback and teaching observations, we identify perceived strengths, operational trade-offs, and lessons for scaling active, lab-centered networking education in mid-sized cohorts.

Keywords: Networking education · Team-based learning · Flipped classroom · Open educational resources

1 Introduction

Computer networking is a foundational topic in many computing curricula, and it remains hard to teach well. Students must connect layered abstractions to a complex, operational reality, practice tool-based skills, and develop debugging intuition. These challenges have motivated work on tailored pedagogical approaches and infrastructures that increase engagement and support learning, including flipped classroom course designs, hands-on labs and testbeds, assessment innovations, and open, reusable educational resources.

In this context, the latter is a major strength of the field. Kurose and Ross provide extensive instructor-facing slide decks to accompany their foundational text book [9] and released these for broad reuse and adaptation [10]. Bonaventure et al. have developed an open-source eBook and a set of open educational resources for introductory and advanced networking courses, including an automated grading platform that helps scale hands-on assignments to large cohorts

* An earlier version of this work was presented at the ACM SIGCOMM Education Workshop, which has non-archival proceedings.

with limited staff [4]. A range of freely available lab platforms enable experiential learning, from lightweight emulation such as Mininet [12] and per-team isolated environments such as iLab@Home [14] to larger course-wide operational environments such as ETH Zurich's mini-Internet project [7].

These resources reduce important barriers for educators, but they do not make course design "plug-and-play". Much of the difficult work lies in the integration between them. An instructor still has to decide what students should prepare before class, how to detect and respond to misconceptions, how class time should connect conceptual discussions to later lab work, how practical environments can be kept reliable, and how to give meaningful feedback without overburdening the teaching staff. Without such an integration, students are likely to experience preparatory readings, class activities, and labs as disconnected tasks, and educators face high coordination and support costs.

The goal of this paper is to make this integration problem concrete and to share transferable experiences. We report on the design and operation of an introductory networking course in the five-year integrated master's program *Cyber Security and Data Communication*, cf. Section 2.1. The design combines three elements into one recurring weekly cycle. We use Team-Based Learning (TBL) [15] to structure preparation and in-class activity, with individual readiness checks preceding team discussion. Class time is organized around team discussion, application-oriented activities and targeted clarification rather than repeated presentation of the preparation material. Compulsory labs, formative checkpoints and periodic deliverables connect concepts to configuring, observing and debugging running networks.

We evaluate the course design as an experience report, drawing on course surveys, reference-group dialogue reports and observations by the educators and teaching assistants. We use these sources to identify student perceptions, trade-offs and lessons for educators who want to combine active learning with practical networking labs under realistic staffing constraints. We further contribute with an integrated 13-week workflow that aligns preparation, in-class readiness assurance and labs into one reusable cycle; networking-specific readiness-assurance questions and team activities; a lab and feedback model based on per-team emulation environments, notebook checkpoints and integrative deliverables; and the supporting materials themselves, released for adoption and adaptation [18].

2 Course Context and Learning Goals

2.1 Course context and constraints

The course is an undergraduate "Computer Networks" course, offered in the second semester of the five-year integrated master program on *Cyber Security and Data Communication*. As the first networking principles-course in the program, it lays the groundwork for the subsequent curriculum in networking and distributed systems. It is a 7.5 European Credit Transfer and Accumulation System (ECTS) course taught in English, which runs over a 13-week semester with a typical cohort of 50–70 students [13].

The edition reported here is Spring 2026, and the design has been adjusted over several iterations. Team-Based Learning, weekly readiness assurance and an individual oral examination have been part of the course since Spring 2020, while the course components around them evolved significantly. The RATs were first paired with a semester project, from 2022 with practical team tasks and now with the lab deliverables of Section 3.4. The weight of the oral examination varied between 30 and 60 percent of the final grade over that period. Written reflections were added as a separate graded component and dropped again for 2026, cf. Section 4.

The 7.5 ECTS correspond to a nominal workload of about 200 hours. Scheduled time is two two-hour sessions and two three-hour lab slots per week, within which groups normally finish the lab work, and the remainder is preparation. We did not record actual time use. In the course survey, 21 of 32 respondents disagreed that the workload had been too much [16].

Four further conditions shape the course design. Firstly, this is a first-year course in a five-year integrated program, and the intake is heterogeneous in terms of students' programming and operating-systems background. Besides leveling that out, the course carries a responsibility for the cohort becoming a group, which we address through fixed teams and near-peer mentors. Secondly, the lab work is a major graded component and carries half of the portfolio grade, cf. Section 4. That weight motivates students to work on the labs continuously, and to use the formative feedback channels as they go, the notebook checkpoints and the weekly check-ins with their teaching assistant. Thirdly, the course is staffed by four student teaching assistants (TA) and a PhD teaching assistant. Each lab group maintains the same student TA mentor throughout the semester rather than pooling assistants across sessions, so that every group has a stable and safe environment in which to work and ask questions. The way this staffing budget is allocated is described in Section 3.5. Fourthly, in-class sessions take place in an active-learning room with group tables rather than a tiered lecture hall, which makes the weekly team discussions frictionless.

Throughout the semester, the course is accompanied by a *reference group*. A small group of students drawn from the cohort meets the teaching staff at least three times during the term and summarizes the cohort's feedback in a written report. This gives students and instructors a continuous, formal feedback channel during the semester rather than only an end-of-term evaluation.

2.2 Learning goals and topic coverage

The course aims to provide students with (i) foundational knowledge of layered architectures, services and protocols used in computer networks, (ii) practical skills in configuring and operating networks and common Internet services, and (iii) competence to collaborate responsibly and effectively in teams [13]. The formal learning outcomes include core reference models and protocols across the stack, from physical/link to application, practical deployment and use of widely used Internet services, and fundamentals of network and communication security. The topic arc follows the standard networking progression of Kurose

and Ross [9]. It runs from an introductory "big picture" through application-layer protocols, transport-layer principles, network-layer data and control plane concepts, and link-layer/LAN technologies to network security. In our course, security is a dedicated thread of two weeks late in the semester. It covers basic cryptographic concepts followed by applied mechanisms and configuration tasks such as TLS, VPNs and firewalling, in the context of the earlier layers.

2.3 Generative AI policy

Our institution provides a general policy on the use of generative AI. It is too generic to answer the practical question students face here. What may AI be used for, and what must remain their own work? The question is acute because the learning outcomes go beyond producing correct written answers. Students prepare individually, reason with peers, configure and debug running network services, and must explain their own operational choices.

We therefore introduced a course-level "AI contract" with one core principle. *AI may be used as a coach to support learning, but not as a shortcut to produce work the student cannot explain and defend* [18]. The enforcement rests on the course design and not on the detection tools or disclosure rules. The goal is to make the *assessment* independent of whether and how students use generative AI, without restricting its role in *learning*.

The contract plays out consistently across the weekly activities. Students may use AI to prepare, for explanations, self-quizzing and condensing the week's material onto the permitted one-page cheat sheet. The individual and team readiness assurance tests, iRAT and tRAT, cf. Section 3.2, are written on paper in class without online aids. The in-class application phase is carried out within the team, so disagreements are contested rather than outsourced. And lab checkpoints verify behavior in the team's actual environment, so a plausible-sounding suggestion passes only once it has been configured and verified. For lab deliverables, AI may assist with clarity and structure, but students must cite primary technical sources rather than the chatbot and explain their weekly task work to an assistant as they go, cf. Section 3.4. The oral examination requires each student to defend their understanding without AI access.

3 Teaching Methods

3.1 Weekly workflow

The weekly design follows Biggs' notion of *constructive alignment*. Learning outcomes, activities and assessment are aligned, so that what students *do* to succeed is what the course intends them to learn [2]. Networking behavior is context-dependent, and plausible-sounding answers may be wrong without the specific environment and traces to verify them against. The weekly readiness assurance, team discussion and labs therefore reward evidence-based reasoning and debugging practice, not recall alone.

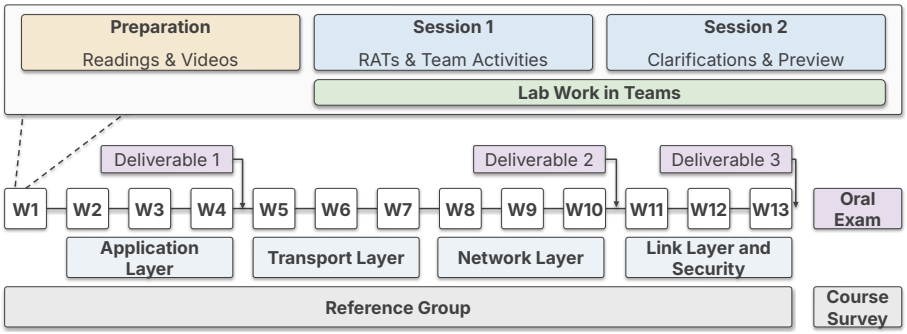


Fig. 1: Semester structure with weekly workflow in the course: flipped preparation, iRAT/tRAT, in-class team activities, clarification, and labs.

The same alignment argument applies to the choice not to lecture. The examination is oral, so students must explain their understanding and defend it under questioning, cf. Section 4. Team-based readiness assurance gives weekly practice in this, since teams have to justify their answers to each other. It also ensures that students have prepared the topic before the corresponding lab. Contact time is therefore used for discussion and clarification, while the textbook covers the material itself. The design was chosen when the course was built and revised since, and we did not benchmark it against alternatives.

Concretely, the course follows a flipped structure. Students prepare before class using curated materials, and class time is used for readiness assurance, structured discussion and targeted clarification. We implement this structure [3] by replacing lectures with the TBL cycle [11]. The in-class sequence runs (1) preparation, (2) individual readiness assurance (iRAT), (3) team readiness assurance (tRAT) with immediate feedback, and (4) application-oriented activities [15].

Figure 1 shows how these elements recur across the 13-week term. Each week pairs preparation and in-class readiness assurance with hands-on lab work on the same topic, the topic progression follows the top-down arc introduced in Section 2.2, and the individual oral examination at the end of the term assesses the same reasoning that the weekly activities practice. The weekly workflow runs as follows. Before the first lecture, students complete assigned reading and preparatory prompts aligned with the week's topics. In the first session they sit an iRAT individually and hand it in, then a tRAT in small discussion groups with immediate feedback (scratch-card style), about 20 minutes each, followed by an appeal opportunity and structured team application activities, cf. Section 3.3. These weekly readiness checks are intended to level out differences due to students' heterogeneous prior knowledge and backgrounds. The second session revisits difficult concepts revealed by the readiness assurance process, previews the next week's material, and runs further team activities. In the labs, which follow the lectures, students work in small teams in reproducible emulation en-

vironments with formative checkpoints, and lab work spans multiple weeks and culminates in three lab deliverables. The split of in-class time between clarification, preview and team activities is adjusted weekly, depending on the topic and on what the readiness assurance reveals. Student discussion groups are formed from a brief self-assessment, so that teams are complementary in skills and prior knowledge.

3.2 Readiness Assurance Tests (RATs)

The weekly RATs are the backbone of the TBL workflow, and both the iRAT and the tRAT contribute to the portfolio assessment, cf. Section 4.

iRAT and tRAT roles The iRAT serves as an individual readiness check that incentivizes pre-class preparation and reveals misconceptions early, while the subsequent tRAT externalizes reasoning through peer discussion, requiring teams to reach consensus and immediately confront incorrect reasoning through the feedback mechanism [15]. We differentiate the two using the revised Bloom taxonomy [1]. iRAT items target the lower cognitive levels of remembering and understanding, which later work needs, whereas the tRAT exchanges some questions to trigger explanation and analysis in context and so improve the quality of team discussion.

Why we add a small number of new tRAT questions. A potential weakness of using identical questions in the iRAT and tRAT (as in the original TBL-approach) is that the team phase can drift towards answer recall ("what did I pick earlier?") rather than explanation and concept application. Farmer et al. [6] evaluated a modified readiness assurance process in which the in-class tRAT used different, but conceptually related questions compared to the iRAT. The approach did not significantly improve unit exam performance. A majority of students (69%) nonetheless preferred it, and reported better content interaction and more higher-order thinking. Motivated by this evidence and our own classroom observations, we include a small number of new, contextualized questions in each tRAT to reliably trigger discussion beyond simple answer recall.

Tooling: Teampy for scalable paper-based RATs. To keep weekly paper-based RATs manageable in a mid-sized cohort, we use *Teampy*, an open-source toolkit for managing readiness assurance tests. Teampy supports writing RATs in a simple text format and generating individualized PDFs by shuffling questions and answer options per student and per team, reducing copying and streamlining RAT distribution [8]. It also supports rapid grading and reporting by collecting answers in a compact textual format and producing per-question summaries that can guide immediate clarification.

In the application-layer week, the item shared by iRAT and tRAT asks whether HTTP is stateless, with distractors drawn from common misconceptions, and careful preparation is enough to answer it. Figure 2 shows the item added only in the tRAT that week. It embeds the same conceptual material in

an operational scenario and asks teams to weigh configuration options against explicit constraints, so reaching consensus requires justifying a decision rather than recalling an answer.

Additional tRAT-only item (application/analysis level)

A small departmental website serves one HTML page plus 18 additional objects (images/CSS), all from the same host over HTTP/1.1. Persistent connections are disabled, static files carry no Cache-Control, clients are not configured to use any proxy, and the RTT to the server is about 100 ms with negligible loss. Which change will most reduce the total completion time of the initial page load for typical users under these constraints?

(a) Enable HTTP persistent connections (keep-alive) so the browser can reuse one TCP connection for multiple object requests.

(b) Add Cache-Control (e.g., max-age) and validators (ETag/Last-Modified) to all static objects.

(c) Stand up a forward web cache (proxy) near the server, but keep client browsers unchanged.

(d) Enable cookie-based sessions so the server can maintain state across object requests.

Fig. 2: The tRAT-only item from the application-layer week. Unlike the shared iRAT/tRAT item on the same topic, it requires reasoning about a configuration decision under operational constraints.

3.3 Team application activities: discussions and short exercises

After the readiness assurance phase, class time is used for structured team activities that require students to apply concepts and justify their decisions. This corresponds to the TBL principle that readiness assurance should feed into application-focused work, with inter-team discussion facilitated by the instructor [15]. The prompts are drawn from standard introductory Kurose and Ross [9] or designed in-house, and they follow the topical progression of the course. We present a typical activity briefly together with its motivation, teams have about ten minutes to work through it, and we then walk through the proposed solution path and discuss the results together.

3.4 Labs: reproducible environments, formative feedback, and deliverables

Labs are the hands-on counterpart to the in-class workflow. Students configure services, observe protocol behavior and debug with realistic tools.

The lab infrastructure is organized around a “template-and-update-workflow”. Before the course, the staff prepare a Virtual Machine (VM) template containing

the required dependencies, among them GNS3, the supporting system packages and the notebook environment. That template is cloned into a separate VM per lab team, with access restricted to that team. Each team therefore has its own VM and GNS3 topology as an isolated working environment. New lab material is released through a course Git repository, so students update by pulling the latest notebooks rather than reconfiguring the platform each week.

Figure 3 shows a representative topology from the DNS lab. An internal lab network connects clients, a DNS server and a web server through a switch, and NAT and Docker network access let the emulated services reach the surrounding lab infrastructure while staying isolated from other teams. Because every team works from an identical template, observations are comparable across teams, and each exercise keeps its operational character. In the DNS lab, students configure an authoritative DNS service, connect it to the topology and verify that name resolution works from the relevant hosts in their own environment.

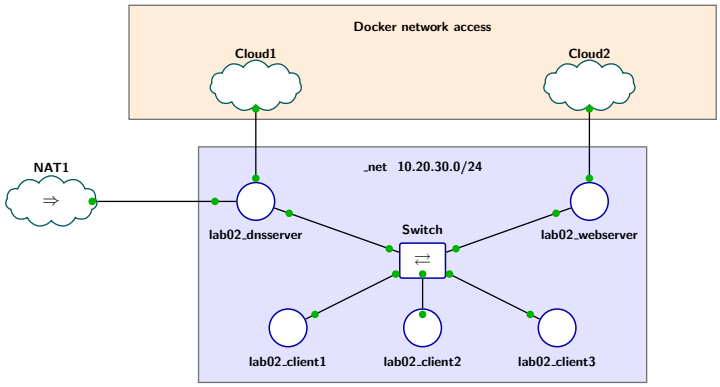


Fig. 3: GNS3 topology for a team-specific DNS lab environment. The internal network contains clients, a DNS server, and a web server connected via a switch.

Formative feedback via interactive notebooks To provide frequent formative feedback without excessive staff overhead, lab tasks are accompanied by interactive notebooks with checkpoints and structured prompts. Checkpoints let students verify intermediate states such as service reachability or configuration properties before progressing, and they let staff focus support on teams that are blocked.

Because the notebooks run from the corresponding team VM, the checkpoints can test the operational state rather than only checking static answers. A DNS checkpoint can, for example, query the team's DNS server, verify that the expected name resolves, and return feedback that points students toward likely configuration errors. This makes the notebook a bridge between the written lab instructions and the emulated network shown in Figure 3.

Multi-session deliverables Lab work is organized as weekly tasks that pair practical configuration with the discussion questions belonging to it, and teams check in with their assistant on these tasks as they go, explaining what they built and why it works. Every few weeks the accumulated task work is submitted as a single deliverable, three across the term, cf. Figure 1, instead of weekly reports. This encourages synthesis across lab topics and keeps the grading workload manageable. The notebook checkpoints give immediate feedback on local progress during a session. The deliverable asks students to explain what they configured, why it worked and how they verified it, with selected configuration excerpts, command output, packet traces or topology observations and a short interpretation. The weekly check-in keeps a team on track and makes them articulate their reasoning out loud, and the deliverable is the summative component, assessing their ability to synthesize and communicate a completed body of work.

The full question sets for all weeks, the notebooks and their checkpoint implementations, and the deliverable descriptions with their marking expectations are all part of the released material [18].

3.5 Teaching resources and support structure

The main practical constraint on scaling flipped TBL and hands-on labs is staffing, because frequent feedback is central to the design. In the 2026 edition, a single instructor was responsible for course teaching and overall grading. The weekly operational tasks are running the readiness assurance process, reviewing its outcomes to identify misconceptions, delivering targeted clarifications and keeping up the dialogue with students. Counting teaching, lab supervision, coordination and examination, the course amounts to roughly 20% of a full-time position for the instructor, of which the four contact hours and some three hours of weekly RAT preparation are the recurring core. Each teaching assistant contributes on the order of 120 hours across the semester.

Lab teaching assistants and near-peer mentoring Lab activities were supported by four student teaching assistants and a PhD teaching assistant (TA), with responsibilities separated. The student TAs provided *formative* support during lab sessions, that is guidance, troubleshooting and the weekly task check-ins. The PhD TA coordinated the lab programme and wrote the feedback on the deliverables, keeping it consistent across teams. The 50 students were organized into 17 lab groups of 2–4, each with a fixed student TA mentor as its first point of contact, which lowers the threshold for asking questions and lets students in this second-semester course form early connections to senior students.

Scalability and planned staffing reductions Near-peer mentoring remains resource-intensive in TA time. For the next iteration we plan to run the same workflow with fewer TAs, by enlarging the cluster each mentor covers and formalizing practices for peak demand such as queueing and prioritization. Combined with the structured checkpoints and self-service verification described above, we expect this to support larger cohorts without a proportional increase in staffing; we treat this as a design hypothesis to be validated in the next edition.

4 Assessment Design

The course uses a portfolio-based assessment model that combines continuous formative assessment with an individual oral examination. The goal is to emphasize learning and engagement over fine-grained point accumulation, in line with the principles described by Edström [5]. The final course grade combines the portfolio (40%) with the individual oral examination (60%).

4.1 Portfolio Assessment

The portfolio comprises three components, the iRATs, the tRATs and the lab work including its written deliverables. Within the portfolio, the RATs and the multi-session lab deliverables are weighted equally at 50% each.

Both components use a *coarse-grained* scheme instead of detailed numerical scores. A component scores 2 points when the learning outcomes are clearly achieved, 1 for partial achievement and 0 otherwise, with half-steps on the deliverables where more differentiation proved useful. For the RATs the thresholds follow the number of correct answers, for instance 2 points from 7 of 10 correct and 1 point from 3 of 10. The three weakest RAT results are dropped, which absorbs illness and bad weeks without a formal procedure. RATs and deliverables then enter as percentages weighted equally, and the portfolio percentage is mapped onto whole grades on a scale stricter than the national reference scale but not punitive, with an A from about 95% and a B from about 88%.

The thresholds are discussed openly with students and may be raised during the term when performance is consistently strong, for instance from 7 to 8 correct answers for 2 points. Any such change applies only prospectively. The reference group, cf. Section 2.1, is the formal channel for this calibration. It lets students signal during the semester whether they feel adequately challenged, so the level can be adjusted mid-course and not only between editions.

The portfolio described here is itself a product of that channel. For several iterations the portfolio included individual written reflections as a graded component. Successive reference groups reported the same difficulty. The expectations were unclear, and the task caused confusion more than learning. The published reports show that this recurred across years and was not one bad cohort [17], and the component was removed for the 2026 edition in favour of the weighting above. The channel therefore affects the design and not only its thresholds. A graded component is fixed in the course description and can only be revised for a following year, which delays such a change and also leaves time to judge a new component across more than one cohort.

4.2 Individual Oral Examination

Because most portfolio activities are team-based, the individual oral examination is the primary instrument for assessing each student's personal attainment of the intended learning outcomes. The oral exam follows the assessment principles outlined by Edström [5], particularly the reversal of the burden of proof. The

student, not the examiner, is responsible for demonstrating that the learning outcomes have been met.

Each examination occupies a 30-minute slot and runs in three stages. The student first presents and summarizes their understanding of selected course topics for ten minutes, using the whiteboard to structure the explanation and to demonstrate achievement of the learning outcomes. The examiners then ask follow-up questions for fifteen minutes, addressing both what was presented and relevant aspects that were omitted, and focusing on conceptual understanding, reasoning and transfer to new situations. In the remaining five minutes the examiners agree on the grade after the student leaves the room, and the student is then invited back for immediate feedback and grade disclosure. For a cohort of 50 this amounts to some 25 hours of examination, spread at about eight students per day to keep the daily load manageable.

Students receive the assessment rubric in advance [18]. A strong performance shows structure and prioritization in the opening summary, breadth with the ability to zoom into detail on request, reasoning about why a mechanism exists and which trade-offs it carries, integration across layers instead of siloed topics, and the use of lab observations and traces as evidence.

The oral examination requires students to justify their reasoning and transfer it to new situations, so it assesses what the weekly readiness assurance, team discussion and labs practice. It provides an individual measure of attainment while the rest of the portfolio remains team-based.

5 Student Perspective

This section summarizes student feedback from the institution's quality assurance material. We report only aggregated responses and anonymized excerpts. The underlying course quality reports, including the course survey results, are published openly in NTNU's Studiekvalitetsportalen [16].

5.1 Data sources

We primarily draw on the course survey reported in the Spring 2026 course quality report [16]. It received 32 responses from the cohort of 50 students (64% response rate) and combines Likert-scale items with two free-text questions on the best aspects of the course and on suggested improvements, answered by 24 and 17 respondents, respectively. We cross-check recurring themes against the published reports from earlier iterations [17]. Those describe a portfolio that still included written reflections, cf. Section 4, so we use them for preparation, readiness assurance and labs only.

5.2 What students valued

On the Likert items, perceptions of the learning design were uniformly positive. All 32 respondents agreed or strongly agreed that the course involved many good

learning activities requiring active work rather than just listening (16 / 16) and that these were well prepared and clearly structured (17 / 15). 31 of 32 agreed that the course required preparation for the lessons (9 / 22), and all 32 that they were happy with the course overall (10 / 22) [16].

In the free-text responses, three value drivers recur. Students most frequently highlighted the *hands-on labs*, emphasizing that they made the theory concrete and relevant. One student described this as "the material and practical work/labs showing the relevancy of what we learn in theory". They repeatedly noted that the *weekly readiness assurance* created a productive study rhythm by requiring continuous preparation. One wrote that "the RATs are really important to keep up with the content of the course, because it requires us to read and understand each week's pages". And they valued *team discussion and clarification*. The tRAT phase and the clarifications that follow it resolve misconceptions and deepen understanding, as in "team RATs are also great" and "the discussions and taking the time to make things that are unclear understandable".

5.3 Friction points

Improvement suggestions in the Spring 2026 survey concern operational fine-tuning more than structural issues [16]. Some students reported that labs and lessons were not always sufficiently aligned, as in "some labs and lessons were not always aligned, so sometimes I felt a little lost in the labs". Several requested clearer wording and better contextualization of difficult RAT items, noting that some questions felt unexpectedly difficult or not clearly enough connected to the preparation material. A recurring concern was waiting time for assistance during labs, "we had to wait a long time when we needed help". One respondent found the weekly reading too concentrated, having worked through it over the weekend before the Monday RAT. The material is released well in advance, so this concerns the scheduling of preparation and not its volume. Some students also wished for a clearer picture of what to expect in the oral examination.

The weakest Likert item concerns feedback, with only 15 of 32 respondents agreeing or strongly agreeing that they received many valuable comments on their submissions. The same respondents rated the other two feedback items much higher, 30 of 32 for the learning assistants' help and 25 of 32 for timely answers [16]. The item asks about submissions, and written feedback in this course arrives only with the three deliverables, cf. Section 3.4, so it captures a real limit. It does not capture the tRAT result, the clarification session, the notebook checkpoints or the weekly check-in, which reach students immediately and orally and are apparently not counted as feedback.

A related pattern appears in two further items. 29 of 32 respondents had a clear picture of what was required of them, while 23 found it clear what quality was expected and 22 that the assessment criteria were clear [16]. The coarse-grained scale of Section 4 gives the RATs an explicit threshold in correct answers and leaves the corresponding step on the deliverables to a judgement about the quality of an explanation.

6 Discussion and Conclusions

We close with three observations from several iterations of the course that we expect to apply elsewhere.

The first one concerns generative AI. In networking, the relevant state is observable, so a claim can be tested against the system it describes. The course uses this property at the three points described in Section 2.3 where the work has to be owned in person. None of them detects AI use, and we did not measure it. Their purpose is to make the question irrelevant to the grade. Disciplines without such an external arbiter of truth face a harder problem.

The second concerns the scaling of feedback. A tRAT serves the whole cohort within the same twenty minutes, while a team blocked in a lab occupies one assistant. Automated checkpoints reduce part of that load, since verifying operational state can be scripted, and we would automate that part first. The survey also shows that the feedback students do receive is only partly registered as such, cf. Section 5.3. Naming immediate oral feedback as feedback when it is given is a cheap way to address part of this.

The third concerns the pace of change. Teaching can be adjusted within the semester, while a graded component is fixed in the formal course description and is revised a year ahead. This is partly a constraint and partly appropriate, because a new component needs more than one cohort to evaluate it. We dropped the written reflections, cf. Section 4, only after several cohorts had reported the same difficulty. The next revision concerns the lab deliverables, for which a more explicit rubric would address the uncertainty reported in Section 5.3.

Whether any of this makes students learn more we cannot say. We report perceptions from one course and one cohort, without a comparison group. Our claim is limited to the integration itself. Existing materials serve each element well on their own, and the effort went into assembling them into a single weekly rhythm. We release ours so that others can reuse that assembly.

Acknowledgments. We thank the PhD teaching assistants and student teaching assistants of all course iterations, whose dedicated support is a cornerstone of the course, the student reference groups of this and previous years for shaping the course design, and all students whose engagement and survey responses inform this experience report. Generative AI (Anthropic's Claude) was used throughout the preparation and revision of this paper as a writing and editing aid, in line with the AI-as-coach principle of Section 2.3; all content and claims are the authors' own.

References

1. Anderson, L.W., Krathwohl, D.R. (eds.): A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Longman, New York (2001)
2. Biggs, J.: Enhancing teaching through constructive alignment. Higher Education **32**(3), 347–364 (1996). https://doi.org/10.1007/BF00138871

3. Bishop, J.L., Verleger, M.A.: The flipped classroom: A survey of the research. In: 2013 ASEE Annual Conference & Exposition. Atlanta, GA (2013). https://doi.org/10.18260/1-2--22585
4. Bonaventure, O., De Coninck, Q., Duchêne, F., Gégo, A., Jadin, M., Michel, F., Piraux, M., Poncin, C., Tilmans, O.: Open educational resources for Computer Networking. ACM SIGCOMM Computer Communication Review **50**(3), 38–45 (2020). https://doi.org/10.1145/3411740.3411746
5. Edström, K.: The teaching trick: How to improve student learning without spending more time teaching. Presentation slides, https://www.janleenkloosterman.nl/reports/icab_edstrom_20161109.pdf (2016), accessed 2026-05-12
6. Farmer, T., Johnson, M.C., Larsen, J.D., Davidson, L.E.: Exploring a modification to the readiness assurance process in Team-Based learning. Advances in Physiology Education **49**(2), 366–373 (2025). https://doi.org/10.1152/advan.00062.2024
7. Holterbach, T., Bühler, T., Rellstab, T., Vanbever, L.: An open platform to teach how the Internet practically works. ACM SIGCOMM Computer Communication Review **50**(2), 45–52 (2020). https://doi.org/10.1145/3402413.3402420
8. Kraemer, F.A.: Teampy: Tools for Team-Based learning (source code). https://github.com/falkr/teampy (2025), accessed 2026-05-07
9. Kurose, J.F., Ross, K.W.: Computer Networking: A Top-Down Approach. Pearson, Harlow, UK, 8 edn. (2021)
10. Kurose, J.F., Ross, K.W.: Computer networking: A Top-Down approach – instructor slide decks (8E/9E). https://gaia.cs.umass.edu/kurose_ross/ppt.php (2025), accessed 2026-05-07
11. Michaelsen, L.K., Knight, A.B., Fink, L.D. (eds.): Team-Based Learning: A Transformative Use of Small Groups in College Teaching. Stylus Publishing, Sterling, VA (2004)
12. Mininet Project: Mininet: An instant virtual network on your laptop (or Other PC). https://mininet.org/ (2022), accessed 2026-05-07
13. Norwegian University of Science and Technology: TTM4200 Computer Networks – course description. https://www.ntnu.edu/studies/courses/TTM4200 (2026), accessed 2026-07-02
14. Pahl, M.O.: iLab@Home: Hands-on networking classes without lab access. In: 2020 ACM SIGCOMM Education Workshop: Teaching and Learning Computer Networking During the Pandemic and Beyond (2020), https://s2labs.org/download/publications/2020_SigComm_iLab_at_home.pdf, white paper, informal proceedings
15. Team-Based Learning Collaborative: Definition and core elements of Team-Based learning. http://www.teambasedlearning.org/definition/ (2026), accessed 2026-05-07
16. TTM4200 Teaching Team: Course quality report TTM4200 Computer Networks, spring 2026. NTNU Studiekvalitetsportalen, https://innsida.ntnu.no/studiekvalitetsportalen/emner/TTM4200/2026/spring (2026), includes the reference-group report and course survey results; accessed 2026-09-15
17. TTM4200 Teaching Team: Course quality reports TTM4200 Computer Networks. NTNU Studiekvalitetsportalen, https://innsida.ntnu.no/studiekvalitetsportalen/emner/TTM4200 (2026), published course quality and reference-group reports for all iterations; accessed 2026-09-15
18. TTM4200 Teaching Team: TTM4200 Computer Networks: Course materials, lab specifications and notebooks. Zenodo (2026). https://doi.org/10.5281/zenodo.21218218