## Course Syllabus for Fall 2026

**Delivery:** 100% Web-Based, Asynchronous

## Instructor Information

Course Creator and Instructor: Dr. Maria Konte, mkonte@gatech.edu
Head TAs: Erick Herring erickherring@gatech.edu;
Ken Westdorp westdorp@gatech.edu

## About This Course

Welcome! This is a graduate-level course on computer networks. The course covers the core principles, protocols, algorithms, and systems that make modern networks function, with an emphasis on both conceptual understanding and hands-on practice.

The course philosophy connects networking theory, real systems, and evidence-based reasoning. Lectures introduce the concepts and design principles behind the Internet, and projects allow students to implement, simulate, measure, and analyze core networking systems.

Guided tutorials extend this learning into modern networking and systems environments. Students work with practical platforms and frameworks such as Kubernetes, Docker-based simulations, NGINX/OpenResty, OpenTelemetry, Containerlab, FRRouting, Mininet, OpenFlow, POX, and BGPStream. These tutorials give students hands-on exposure to tools for observing, debugging, and reasoning about real network behavior.

A central theme of the course is **evidence before answers**. Students learn to explain network behavior using measurements, packet-level observations, routing-table output, logs, telemetry, and system traces. The course also introduces guided use of AI in the loop: students examine how AI-assisted diagnosis can support troubleshooting when it is grounded in tools, observations, and checkable evidence rather than guesses.

The three main objectives of the course are:

1. **Understand how modern computer networks function.**
   Students learn the core principles, protocols, algorithms, and architectures that make the Internet operate at scale, including transport, routing, router design, Software Defined Networking, security, multimedia applications, and Content Delivery Networks.
2. **Apply networking concepts through hands-on projects and tutorials.**
   Students implement, simulate, measure, and analyze networking systems through projects and guided tutorials, including routing algorithms, SDN firewalls, BGP hijacking, BGP measurements, Kubernetes networking, CDN design, performance diagnosis, and AI-assisted network diagnosis.
3. **Develop evidence-based debugging and hands-on systems skills.**
   Students learn how to observe network behavior, diagnose failures, interpret measurements, and explain conclusions using concrete evidence. They also gain hands-on experience with practical networking and systems frameworks introduced in the tutorials, including Kubernetes, Docker-based network simulations, NGINX/OpenResty CDN components, OpenTelemetry observability, Containerlab, FRRouting, and AI-assisted diagnostic workflows.

## Course Learning Objectives

By the end of the course, students will be able to:

- **Explain the architecture and evolution of the Internet,** including the Internet protocol stack, and the design choices that allow the Internet to scale.
- **Describe how transport protocols support end-to-end communication,** including reliable delivery, flow control, congestion control, and the tradeoffs between lightweight and reliable transport services.
- **Analyze routing within and across networks,** including intradomain routing, interdomain routing, link-state and distance-vector algorithms, BGP, peering, transit, route selection, and routing policy.
- **Explain how routers forward packets at high speed,** including forwarding-table lookups, switching fabrics, buffering, queueing, scheduling, packet classification, and traffic control.
- **Describe Software Defined Networking technologies and applications,** including the separation of control and data planes, controllers, OpenFlow-style control, programmable networks, and SDN-based traffic and security policies.
- **Evaluate security and reliability issues in modern networks,** including BGP hijacking, DNS abuse, denial-of-service attacks, blackholing, surveillance, censorship, and Internet disruptions.
- **Explain how modern applications operate over networks,** including voice, video, multimedia applications, Content Delivery Networks, caching, server selection, and performance tradeoffs.
- **Implement, simulate, and measure networking systems through hands-on projects,** including spanning tree, distance-vector routing, SDN firewall rules, BGP hijacking, and BGP measurement analysis.
- **Use practical networking, measurement, and systems frameworks to observe network behavior.** Students will gain hands-on experience with routing and emulation frameworks such as **Mininet, OpenFlow, POX, Containerlab, and FRRouting**; measurement and analysis tools such as **BGPStream**; cloud-native and service frameworks such as **Kubernetes, Docker-based simulations, and NGINX/OpenResty**; and observability frameworks such as **OpenTelemetry**.
- **Diagnose network and system behavior using concrete evidence and AI-assisted workflows.** Students will learn to move from symptoms to evidence, from evidence to manual diagnosis, and from manual diagnosis to AI-guided confirmation and refinement. They will use measurements, packet-level observations, routing-table output, logs, telemetry, tool traces, and AI-assisted diagnostic workflows to explain what happened and why.
- **Communicate technical conclusions clearly,** by explaining what happened, what evidence supports the conclusion, and how the observed behavior connects to networking concepts.

## Prerequisites & Required Texts

**There are no formal prerequisite courses**, but this is an advanced course. Many topics from an introductory undergraduate networking course will be reviewed only briefly or not at all.

Students are expected to have a basic understanding of networking concepts, including IP addresses, subnets, TCP vs. UDP, the OSI model and encapsulation, TCP three-way handshakes, the basic operation of HTTP, and the difference between client-server and peer-to-peer models.

All programming in this course will be done in **Python 3**. An intermediate level of Python is sufficient for the projects. Students who are comfortable learning new programming languages, or who have completed Python programs beyond basic “Hello World” examples, should be able to learn what they need during the course, though it may take extra time. Students who have never programmed in Python, or who have only completed very basic Python tutorials, are strongly encouraged to learn Python before the course begins.

**There are no required texts for this course**. For the first part of the course, where we review foundational material, *Computer Networking: A Top-Down Approach* (7th, 8th, or 9th editions) by Kurose & Ross is an excellent optional reference. There is also an excellent website that hists support material for the book.

## Lecture Topics, Guided Tutorials, and Ed Discussion Participation

Lectures introduce the core concepts behind modern computer networks. The course begins with Internet architecture, layering, encapsulation, the end-to-end principle, and the protocol stack. It then moves through transport protocols, intradomain routing, interdomain routing, router design, Software Defined Networking, Internet security, multimedia applications, and Content Delivery Networks. Across these topics, the course emphasizes how networks scale, how protocols make forwarding and routing decisions, how failures happen, and how engineers use evidence to explain system behavior.

Topics include:

- **Internet architecture and design principles:** We study why the Internet is organized in layers, how data is encapsulated as it moves through the protocol stack, and how design principles such as the end-to-end principle helped the Internet scale and evolve. We also discuss practical exceptions.
- **Transport protocols and end-to-end communication:** We study how applications communicate across the network using transport protocols. This includes reliable delivery, flow control, congestion control, and the tradeoffs between lightweight and reliable transport services.
- **Intradomain routing:** We study how routers find paths within a single organization or administrative domain. This includes the difference between forwarding and routing, how routers compute paths, how link-state and distance-vector algorithms work, and what can go wrong during convergence or routing instability.
- **Interdomain routing and network relationships:** We study how independently operated networks exchange reachability information across the global Internet. This includes how networks advertise address blocks, choose routes, apply policy, and form business relationships such as transit and peering. We also discuss peering infrastructure (Internet exchange points and route servers).
- **Router design and packet forwarding:** We look inside routers to understand how packets are processed at high speed. Topics include forwarding-table lookup, switching fabrics, buffering, queueing, scheduling, packet classification, rate control, and the performance challenges routers face under load.
- **Software Defined Networking:** We study why traditional networks can be hard to manage and how software-defined networking separates control logic from packet forwarding. We discuss controllers, programmable network behavior, OpenFlow-style control, programmable data planes, and modern uses of SDN in wide-area, cloud, and exchange-point settings.
- **Internet security:** We examine how Internet protocols and infrastructure can be abused. Topics include DNS abuse, network reputation, BGP hijacking, traffic attraction attacks,

denial-of-service attacks, blackholing, filtering, scrubbing, and defenses deployed by providers and exchange points.

- **Multimedia applications:** We study how audio and video applications operate over networks and why they create different performance requirements from traditional web applications. Topics include voice and video delivery, buffering, compression, bitrate adaptation, latency, quality of experience, and the tradeoffs between reliability and timeliness.
- **Content Delivery Networks:** We study how content is delivered at scale by placing servers closer to users. Topics include edge servers, caching, content placement, DNS-based server selection, CDN-ISP interactions, user-perceived performance, and resilience when origins or paths fail.

Lectures go hand in hand with guided tutorials that help students connect these concepts to hands-on system behavior. The goal is to help students observe real network behavior, diagnose failures, interpret evidence, and explain how modern networked systems operate in practice.

- **BGP with Containerlab Tutorial:** Students build and inspect a small multi-AS BGP topology using Containerlab and FRRouting. They configure eBGP sessions, inspect BGP neighbor state, examine routing tables, observe route advertisements and withdrawals, and connect command output to routing behavior. The tutorial teaches interdomain routing, BGP control-plane behavior, route selection, troubleshooting, and evidence-based interpretation of router state.
- **Kubernetes Networking Tutorial:** Students work inside a small Kubernetes cluster to understand how cloud-native networking works. They inspect pods, Services, Endpoints, DNS, packet forwarding, NAT, and NetworkPolicies. Students learn how to diagnose Kubernetes communication failures using evidence from multiple layers.
- **Content Distribution Network "BuzzStream" Tutorial:** Students build a small content delivery network one layer at a time. They start with a slow origin server and then add edge caching, cache keys, TTL behavior, load balancing, multiple edges, consistent hashing, stale-cache behavior, timeouts, and a backup origin. Students learn how CDNs improve performance, routing, cache locality, and resilience.
- **Network Performance Diagnosis Tutorial:** Students investigate whether slow responses are caused by the network path or by the server application. They use network measurements, packet-level evidence, server-side timing, and observability data to distinguish network-side delay from application-side delay. The tutorial teaches performance debugging, bottleneck localization, latency analysis, throughput analysis, packet loss, retransmissions, and cross-layer reasoning.
- **AI-Assisted Network Fault Diagnosis Tutorial:** Students study network fault diagnosis as an evidence-based workflow. They first diagnose faults manually, then compare their reasoning with an AI-assisted diagnostic. The tutorial uses a reproducible network environment, controlled fault injection, diagnostic evidence, tool traces, and ground-truth evaluation. Students learn how to move from symptoms to evidence, from evidence to diagnosis, and from manual diagnosis to AI-assisted confirmation.

During the semester, there will be six separate Ed Discussions. These discussions will require either that you read or watch some material and provide substantive comment on it or that you complete a challenge. You are required to participate in at least 3 of the 6 discussions.

## Projects

**Spanning Tree Protocol project. (Focuses on Layer 2 / Data Link Layer)**
Students implement a distributed algorithm that builds a loop-free forwarding structure over a Layer 2 topology. The project helps students understand how switches prevent forwarding loops, exchange local messages, converge on a shared tree, and adapt when the topology changes.

**Distance Vector routing project. (Focuses on Layer 3 / Network Layer)**
Students implement a distributed Bellman-Ford routing algorithm. The project focuses on how routers exchange distance vectors, update path costs, handle weighted links, reason about convergence, and detect negative-cycle behavior in simulated network topologies.

**SDN Firewall project. (Focuses Layers 2–4 / SDN and Packet Filtering)**
Students build a configurable firewall using Mininet, OpenFlow, and the POX controller. The project introduces Software Defined Networking by asking students to translate firewall policies into packet-matching rules based on MAC addresses, IP addresses, protocols, and ports.

**BGP Hijacking project. (Focuses Interdomain Routing / Network Layer)**
Students recreate a BGP hijacking attack in a simulated multi-AS topology. The project shows how a rogue autonomous system can advertise another AS's prefix, attract traffic through BGP path selection, and disrupt or redirect communication. *Note: Not offered during summer sessions.*

**BGP Measurements project. (Focuses Internet Measurement / Control Plane)**
Students use BGPStream and PyBGPStream to analyze historical BGP routing data. The project introduces Internet measurement by having students study routing-table growth, advertised prefixes, autonomous systems, AS-path length evolution, announcement-withdrawal events, and RTBH events.

**Extra Credit Project: Demo.** For extra credit, students have the opportunity to complete a creative project and present a demo during an asynchronous demo session on Ed Discussion during the final weeks of the semester. This project is a chance to bring together several tools, techniques, and concepts from the course and apply them to a networking problem of interest.

## Course Assignments and Grading

| Assignments | Regular Semester | Summer Semester |
|---|---|---|
| Spanning Tree Protocol (STP) | 10% | 10% |
| Distance Vector Routing (DV) | 10% | 15% |
| SDN Firewall (SDNFW) | 10% | 15% |
| BGP Hijacking (BGPH) | 10% | --- |
| BGP Measurements (BGPM) | 15% | 15% |
| **Projects Total:** | **55%** | **55%** |
| Exam 1 | 10% | 10% |
| Exam 2 | 10% | 10% |
| Content Quizzes | 5% | 5% |
| Tutorials Quizzes | 15% | 15% |
| Ed Discussion Participation | 5% | 5% |
| **Exam/Quizzes/Tutorials Total:** | **45%** | **45%** |
| Extra Credit Opportunities | 5% | 5% |
| **Extra Credit Total:** | **5%** | **5%** |

## Grading Scale

The final grade will be assigned as a letter grade according to the following scale. There is no rounding, and no curve is applied.

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: 0-59%

## Course Calendar

| Week | Module(s) | Assignments |
|---|---|---|
| **Week 1**<br>Aug 24 – 30 | Welcome Module<br>Lesson 1: Intro, History, and Internet Architecture | Assigned: Spanning Tree **Protocol** (STP) |
| **Week 2**<br>Aug 31 – Sep 6 | Lesson 2: Transport and Application Layers | **Due: Lesson 1 Quiz**<br>**Due: Lesson 2 Quiz** |
| **Week 3**<br>Sep 7 – 13 | Lesson 3: Intradomain Routing | **Due: Lesson 3 Quiz**<br>**Due: Spanning Tree Protocol**<br>**Due: Ed Discussion 1** |
| **Week 4**<br>Sep 14 – 20 | Lesson 4: AS Relationships and Interdomain Routing<br>*Suggested: Network Performance Diagnosis Tutorial* | Assigned: Distance Vector (DV)<br>**Due: Lesson 4 Quiz** |
| **Week 5**<br>Sep 21 – 27 | Lesson 5: Router Design and Algorithms (Part 1) | **Due: Lesson 5 Quiz**<br>**Due: Ed Discussion 2** |
| **Week 6**<br>Sep 28 – Oct 4 | Lessons 6: Router Design and Algorithms (Part 2)<br>*Suggested: BGP Tutorial* | **Due: Lesson 6 Quiz**<br>**Due: Distance Vector** |
| **Week 7**<br>Oct 5 – 11 | Exam 1 | **Due: Exam 1** |
| **Week 8**<br>Oct 12 – 18 | Lesson 7: SDN (Part 1)<br>*Suggested: Kubernetes Tutorial* | Assigned: SDN Firewall (SDNFW)<br>**Due: Lesson 7 Quiz**<br>**Due: Ed Discussion 3** |
| **Week 9**<br>Oct 19 – 25 | Lesson 8: SDN (Part 2) | **Due: Lesson 8 Quiz**<br>**Due: SDN Firewall** |
| **Week 10**<br>Oct 26 – Nov 1 | Lesson 9: Internet Security<br>*Suggested: Network Fault Diagnosis Tutorial* | Assigned: BGP Hijacking (BGPH)<br>**Due: Lesson 9 Quiz**<br>**Due: Ed Discussion 4** |
| **Week 11**<br>Nov 2 – 8 | Lesson 10: Internet Surveillance and Censorship | **Due: Lesson 10 Quiz**<br>**Due: BGP Hijacking** |
| **Week 12**<br>Nov 9 – 15 | Lesson 11: Applications: Video & Multimedia<br>*Suggested: BuzzStream CDN Tutorial* | Assigned: BGP Measurements (BGPM)<br>**Due: Lesson 11 Quiz**<br>**Due: Ed Discussion 5** |
| **Week 13**<br>Nov 16 – 22 | Lesson 12: Applications: CDNs & Overlay Networks | **Due: Lesson 12 Quiz** |
| **Week 14**<br>Nov 23 – 29 | Lesson 13: Looking Ahead: AI and the Internet | **Due: BGP Measurements** |
| **Week 15**<br>Nov 30 – Dec 6 | Exam 2 | **Due: Exam 2** |
| **Week 16**<br>Dec 7 – 13 | | **Due: Tutorial Quizzes**<br>**Due: Ed Discussion 6**<br>**Due: Extra Credit Project** |

* **Note**: For registration, withdrawal, semester start dates, and other deadlines and important dates, please consult the current calendar: [https://registrar.gatech.edu/info/current-academic-calendar](https://registrar.gatech.edu/info/current-academic-calendar)

## Accommodation for Students with Disabilities

If you need accommodations for a disability or learning need, please contact the Office of Disability Services as soon as possible at **(404) 894-2563** or **http://disabilityservices.gatech.edu/** to request an accommodation letter.

Please also send a private message to **"Instructors"** on Ed Discussion as soon as possible. The TA team cannot provide accommodations or extensions without an official accommodation letter, and accommodations are not applied retroactively.

## Student/Faculty Expectations

Georgia Tech expects students and faculty to maintain an environment of mutual respect, responsibility, and professionalism. In this course, we ask everyone to communicate cordially, respect the learning process, value hard work, and uphold the standards and ideals of Georgia Tech. http://www.catalog.gatech.edu/rules/22/

## Additional Information on Canvas

Additional information, policies, and procedures will be made available within the Welcome Module on Canvas. All such information, policies, and procedures are syllabus material and it is the student's responsibility to ensure they understand all course information, policies, and procedures.

## Subject to Change Statement

The syllabus and course schedule may be subject to change. Changes will be communicated via email, Ed Discussion, and/or the Canvas announcement tool. It is students' responsibility to check email messages, Ed Discussion, and course announcements to stay current in their online courses.