Computer Networking Research and Where to Go From Here

This document contains my view of what is interesting in computer networking research today, how you can learn more, and where you can contribute to the state of the art.

Courses

Here are a number of courses within the CS and ECE departments that you may want to consider taking:

- CS 343 Operating Systems (Bustamante): On a typical host, the network stack (all of Minet) lives inside of an OS kernel.
- CS 213 Computer Systems (Dinda): An introduction to all aspects of computer systems using Intel PCs running Linux as the example.
- CS 395 Building Internet Services (Dennis): focuses on the application layer and teaches you how to build high performance, high utility servers.
- CS 395 Practicuum on Intelligent Information Services (Hammond/Birnbaum): innovative applications that exploit networks
- CS 395/495 Dynamic Behavior of Hosts and Networks (Dinda): looks at how to use statistical and signal processing tools to measure, characterize and predict network behavior.
- CS 396/495 Advanced Operating Systems (Bustamante)
- CS 399/499 Independent Projects (various): I have many networking related projects about which I would be happy to talk to you. I would also be happy to entertain any of your ideas.
- ECE 222 Fundamentals of Signals and Systems: core theory underlying signal processing
- ECE 307 Communication Systems: how to use signals and systems to communicate information.
- ECE 333 Introduction to Computer Networks: takes a hardware up approach to teaching about networks and covers lots of technologies other than the TCP/IP + Ethernet triumvirate we focused on. Queuing theory intro
- ECE 485 Local Area Networks: concentrates on LAN hardware.
- ECE 486 Queuing Models for Computer Communication

Conferences and journals

Most networking research results are first published in annual conference proceedings, although some good work also appears in journals. Here are some of the conferences and journals to check out. As a student, you can usually get these proceedings quite inexpensively. Membership in an ACM SIG, which typically costs less than $20 for a student, includes a copy of the proceedings. ACM SIGCOMM is an especially good deal as the newsletter (Computer Communications Review) publishes good material. Most authors will also place their papers on their web sites.

- ACM SIGCOMM: This is the premier networking conference.
- IEEE INFOCOM: This is a HUGE networking conference that includes work in a wide variety of areas. The proceedings have been known to inflict blunt force traumas.
- ACM SIGMETRICS: This is the premier performance analysis conference, so it publishes a lot of good work on network performance and characteristics.
- IETF Meetings: anyone can participate in an IETF meeting
- ATM Forum Meetings
- Grid Forum Meetings: anyone may participate
- IEEE HPDC: This is a high quality conference on high performance distributed computing – lots of material on how to use networks to do real work.
- IEEE ICDCS: Distributed computing, more theoretical focus.
- ACM SOSP: The premier operating systems conference. The ACM SIG is SIGOPS.
- SPAA (Symposium on Parallel Architectures and Algorithms) : "Parallel Architectures" is the theoretical study of network topologies and what they are good for from a parallel algorithms perspective.
- NOSDAV workshop: Early results on innovative networking research.
- IEEE Globecom
- IEEE ICC
- ACM Multimedia conference
- IEEE Network: survey articles and tutorials
- IEEE Communications magazine: survey articles
- IEEE/ACM Transactions on Networking: research articles
- IEEE Transactions on Communications
- Computer Communications Review
- IEEE Journal on Selected Areas in Communications

Wireless Networking

- Ad hoc networks: How can a collection of wireless hosts self-assemble into a network? How can that network evolve as they move?
- Antennas, encoding, and modulation: How can we make the physical layer better?
- Power management: How can we avoid sending extra packets, or sending packets with unnecessary power?
- Forward Error Correction: What are the characteristics of wireless errors and what kind of encoding scheme is best to ameliorate their effects?
- Properties of wireless traffic: How is it different from wired traffic? How can we characterize it? Can we predict it? What can we do with such knowledge?
- Reservations and Quality of Service: How can we support high levels of service on channels where we don't have complete control?

Multicast

- Scalable reliable multicast: Currently, IP multicast only supports UDP. How can we provide reliable many-to-many connections? Systems: STORM, SRM

- Self-assembly: How can multicast trees better self-assemble?
- Multicast QoS: How can we provide quality of service or reservations for multicast traffic? What does quality of service even mean in this context?

Routing

- Emergent behavior: is ad hoc networking a good model for the Internet as a whole?
- QoS routing: can we change routing on the fly to meet quality of service?
- Predictive source-based routing: Can packets route themselves based on predictions of the performance of different paths?
- Active networks: Are packets programs?

Parallel computing on networks

- Can commodity networks and hosts replace specialty parallel supercomputers?
- Grid computing: Can we do useful computing on collections of geographically disbursed resources?
- Parallel communication: How can we efficiently map parallel communication primitives such as tree reduction, all-to-all, and nearest-neighbor onto the irregular topology of a network?
- Parallelizing compilers and run-time systems: How can we compile for networks?

Performance

- Hardware: How do we make NICs, switches, and routers that operate at very high bandwidths with low latencies?
- Zero-copy: How can we avoid making unnecessarily many copies in the data path of a network stack? Are new software abstractions or API semantics necessary?
- Virtual interfaces: How can we efficiently expose the NIC to the application so that it can drive it directly?

Network measurement, characterization, and prediction

- Measurement: How can we measure aggregate network properties in a scalable manner?
- Sampling Theory: How can we sample the temporal behavior of a network property well? What are the band-limits?
- Statistical properties: How can we summarize the properties of a network? How can we present that summarization to an application?
- Prediction: Is the dynamic behavior of networks predictable? To what degree? Using what kind of predictive models?
- Predictability: Can we heuristically determine the predictability of a given network property independent of a predictive model?
- Prediction systems: Can we build on-line measurement and prediction systems that scale and do not place an undue amount of load on the system?
- How do we use predictions from such a system in an application?

Queuing theory

- Modern distributions and self-similarity: Real networks have analytically difficult arrival processes and size distributions. How do we incorporate them? What new theory is needed?
- What are the implications of these new processes and theory?
- How do we test these analytical results with reality?

Quality of service and reservations

- Implementing reservations: How can a network infrastructure (the routers) implement reservations for queue slots and other resources?
- What is the nature of such reservations? Are the deterministic? Stochastic?
- How should such reservations be exposed to applications and users?
- Priority scheduling of packets: Can we achieve many of the benefits of reservations through priorty scheduling?
- How can we map end-to-end application requirements to priorities?
- Minimizing work and state: How can we achieve guaranteed levels of service with a minimum amount of work in the routers?
- Game theory: Can a router implement a game in which the optimal strategy for an individual flow is to play fair?
- Real-time: How can a network provide real-time service, either soft or hard?

Qualitity of service and predictive adaptation

- Adaptation mechanisms: By what means can an application adapt to changing network characteristics?
- Adaptation control: How can we control these mechanisms (schedule processes, modulate quality, etc) ?
- Emergent behavior: Can fairness and a good global policy emerge from the decisions of many independently adapting applications?

Distributed systems

- Distributed shared memory and RPC: How can we make a networked environment as easy to program as a single host?
- Distributed Objects: How can we extend the object model out to the network?
- Network file systems and caching
- Distributed algorithms

Security and fault tolerance

- Resilience: How can we make it difficult or impossible for a nefarious individual or organization to take down the network? How can we make it "heal"?
- Fairness: How can we make the network "fair" to all users?
- Privacy: How can we assure that our traffic can not seen by others?

And many other areas I am no doubt forgetting about…