Thank you!

An hour? <jawdrops>
Well, don’t worry about it because no one is going to listen to anything for an hour, unless it’s a concert.

The WiFi finally works these days. Can we get a 3D printer?

Wait, you just got an award, and as a reward they want you do to more work? Are you sure this is a community you want to be in?
The true honorees: My students and postdocs!

- David Andersen
- Venkat Arun
- Arjun Balasingam
- Magdalena Balazinska
- Deepak Bansal
- Frank Cangialosi
- Tiffany Chen
- Shuo Deng
- Nick Feamster
- Prateesh Goyal
- Songtao He
- Wendi Heinzelman
- Bret Hull
- Peter Iannucci
- Kyle Jamieson
- Jaeyeon Jung
- Katrina LaCurts
- Allen Miu
- Akshay Narayan
- Ravi Netravali
- Amy Ousterhout
- Jonathan Perry
- Raluca Ada Popa
- Bodhi Priyantha
- Asfandyar Qureshi
- Lenin Sivalingam
- Anirudh Sivaraman
- Jayashree Subramanian
- Alex Snoeren
- Arvind Thiagarajan
- Mythili Vutukuru
- Mike Walfish
- Keith Winstein
- Jakob Eriksson
- Lewis Girod
- Ramki Gummadi
- Emre Koksal
- Evdokia Nikolova
- Radhika Mittal
- Rohan Murty
- Calvin Newport
- Srinivas Narayana
- Geoff Werner-Challen
Where do research inspirations come from?

- Technology trends
- Operational expertise and know-how
- Applications
Resilient Overlay Networks

an architecture that allows distributed Internet applications to detect and recover from path outages and periods of degraded performance within several seconds, improving over today’s wide-area routing protocols that take several minutes to recover.

D. Andersen, et al., SOSP 2001
13 years later (2014) @Cambridge Mobile Telematics

Sensor data was not uploading from MTN in South Africa to AWS us-west S3.

Chord Distributed Hash Table

A review of the features of recent peer-to-peer applications yields a long list: redundant storage, permanence, selection of nearby servers, anonymity, search, authentication, and hierarchical naming. Despite this rich set of features, the core operation in most peer-to-peer systems is efficient location of data items. The contribution of this paper is a scalable protocol for lookup in a dynamic peer-to-peer system with frequent node arrivals and departures.

The Freenet peer-to-peer storage system [5], [6], like Chord, is decentralized and symmetric and automatically adapts when hosts leave and join. Freenet does not assign responsibility for documents to specific servers; instead, its lookups take the form of searches for cached copies. This allows Freenet to provide a degree of anonymity, but prevents it from guaranteeing retrieval of existing documents or from providing low bounds on retrieval costs. Chord does not provide anonymity, but its lookup operation runs in predictable time and always results in success or definitive failure.

I. Stoica et al., SIGCOMM 2001
Intellectual Impact of DHTs (& Chord)

- Popularized distributed consistent hashing
- Design simplicity
- Key-value stores as a building block
- Flat names in distributed services
- Flat names in network architecture
- Dynamic stabilization protocols

Untangling the Web from DNS
Michael Walfish\textsuperscript{a}, Hari Balakrishnan\textsuperscript{a}, and Scott Shenker\textsuperscript{b}
IRIS Project
\textsuperscript{a}\{mwalfish, hari\}@csail.mit.edu, MIT Computer Science and AI Laboratory (CSAIL), Cambridge, MA
\textsuperscript{b}shenker@icsi.berkeley.edu, International Computer Science Institute (ICSI), Berkeley, CA

A Layered Naming Architecture for the Internet
Hari Balakrishnan\textsuperscript{\textdegree}
hari@csail.mit.edu
Karthik Lakshminarayanan\textsuperscript{\textdegree}
karthik@cs.berkeley.edu
Scott Shenker\textsuperscript{\textdegree, b}
shenker@icsi.berkeley.edu
Ion Stoica\textsuperscript{\dagger}
istoica@cs.berkeley.edu
Sylvia Ratnasamy\textsuperscript{\dagger}
sylvia@intel-research.net
Michael Walfish\textsuperscript{\dagger}
mwalfish@csail.mit.edu

Accountable Internet Protocol (AIP)
David G. Andersen\textsuperscript{1}, Hari Balakrishnan\textsuperscript{2}, Nick Feamster\textsuperscript{3},
Teemu Koponen\textsuperscript{4}, Daeyoung Moon\textsuperscript{5}, and Scott Shenker\textsuperscript{1}
\textsuperscript{1}Carnegie Mellon University, \textsuperscript{2}MIT, \textsuperscript{3}Georgia Tech, \textsuperscript{4}ICSI & HIIT, \textsuperscript{5}University of California, Berkeley
Congestion Control from an App’s Perspective

Architectural Considerations for a New Generation of Protocols

David D. Clark and David L. Tennenhouse
Laboratory for Computer Science, M. I. T.

Application Level Framing

The way to avoid this problem is for the lower layers such as presentation and transport to deal with data in units that the application specifies. In other words, the application should break the data into suitable aggregates, and the lower levels should preserve these frame boundaries as they process the data. This proposal will call these aggregates Application Data Units, or ADUs. ADUs will then take the place of the packet as the unit of manipulation. We call this design principle Application Level Framing.
Congestion Manager: Improving Apps

However, Internet traffic patterns have been changing rapidly and are certain to be very different in the future. First, Web workloads stress network congestion control heavily, and in unforeseen ways. Typical Web transfers are characterized by multiple concurrent, short TCP connections. These short Web transfers do not give TCP enough time or information to adapt to the state of the network, and concurrent connections between the same pair of hosts compete rather than cooperate with each other for scarce resources. Perhaps most importantly, several increasingly popular real-time streaming applications run over UDP using their own user-level transport protocols for good application performance, but in most cases today do not adapt or react properly to network congestion. Furthermore, there are applications such as teleconferencing where multiple concurrent streams co-exist (e.g., audio, video, whiteboards, text), that will benefit from efficient multiplexing and sharing of bandwidth.

HB, S. Seshan, et al., SIGCOMM 99, OSDI 00
Congestion Control Plane: Getting CC out of the Data Path

A. Narayan, F. Cangialosi, P. Goyal, et al., SIGCOMM 18

![Diagram](image)
Twenty Years Ago…
Cricket: Accurate Indoor Location

- RF + ultrasound with self-configuring network of ceiling and wall beacons
- Use time difference of arrival between RF and ultrasound signals
- 3-5 cm accuracy

Fast forward to 2020…

SonicPACT: An Ultrasonic Ranging Method for the Private Automated Contact Tracing (PACT) Protocol

John Meklenburg\textsuperscript{1}, Michael Spector\textsuperscript{2}, Michael Wenz\textsuperscript{1}, Hari Balakrishnan\textsuperscript{1}, Anantha Chandrakasan\textsuperscript{1}, John Cohn\textsuperscript{1}, Gary Hatake\textsuperscript{1}, Louise Ivers\textsuperscript{1}, Ronald Rivest\textsuperscript{1}, Gerald Jay Sussman\textsuperscript{1}, Daniel Weitzner\textsuperscript{1},

\textsuperscript{1}Corresponding Authors. \textsuperscript{1}MIT Lincoln Laboratory, \textsuperscript{1}Massachusetts Institute of Technology, \textsuperscript{1}MIT-IBM Watson AI Lab, \textsuperscript{1}Massachusetts General Hospital

Priyantha et al., MobiCom 00, MobiCom 01, Infocom 05
MIT CarTel Project (2005-2011)

A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks
Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden
MIT Computer Science and Artificial Intelligence Laboratory
(vladb, bwhull, akmiu, hari, madden)@csail.mit.edu
http://cartel.csail.mit.edu

CarTel: A Distributed Mobile Sensor Computing System
Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan and Samuel Madden
MIT Computer Science and Artificial Intelligence Laboratory
cartel@nms.csail.mit.edu

Sensys 2006

MobiCom 2006

CarTel Vehicular Network Testbed

MobiSys 2008

The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring
Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, Hari Balakrishnan
MIT Computer Science and Artificial Intelligence Laboratory
{jakob, girod, bwhull, newton, madden, hari}@csail.mit.edu
Score Driving by Computing Risk Factors

Validate risk factors based on millions of trips and many billions of miles

- Phone Motion
- Screen Interaction
- Cornering
- Hard Braking
- Speeding
- Near/Prior Collisions
- Harsh Acceleration
- Road Type
All I wanted to do was to audio-Skype with my students while driving home (2012)
Sprout in a Nutshell

- Maximize throughput, but
- Bounded risk of delay > D (e.g., D = 100 ms)
- Infer rate from interarrival distribution
- Predict future link rate and convey prediction to sender (before congestion)
- Control: Send fast, but with 95% probability all packets will arrive within D

K. Weinstein, A. Sivaraman, B, NSDI 2013
Example Result
Protocol Design Contests

Anirudh Sivaraman, Keith Winstein, Pauline Varley, João Batalha, Ameesh Goyal, Somak Das, Joshua Ma, and Hari Balakrishnan
Massachusetts Institute of Technology, Cambridge, Mass.
{anirudh, keithw, paulinerv, johnmysb, ameeshg, das, joshma, hari}@mit.edu

ABSTRACT

In habits like data mining and natural language processing, design contests have been successfully used to advance the state of the art. Such contests offer an opportunity to invoke the excitement and challenges of protocol design—one of the core intellectual elements of research and practice in network systems—to a broader group of potential contributors, whose ideas may prove important. Moreover, it may lead to an increase in the number of students, especially undergraduates or those learning via online courses, interested in pursuing a career in the field.

We describe the creation of the infrastructure and our experience with a protocol design contest conducted in MIT’s graduate Computer Networks class. This contest involved the design and evaluation of a congestion control protocol for packet traversing cellular wireless networks. Our key to the success of a design contest is an unambiguous, parameter-restricted to compare protocols. To practice, protocol design is the art of trading off conflicts goals with each other, but in this contest, we specified that the goal was to maximize (throughput) (delay). This goal is a good match for applications such as video streaming or voice over IP that are able to bear high throughput and low latencies.

Some studies on protocol performance were better than others in designing such goals. Furthermore, the contest’s hallmark of the set of all protocol protocols was that it was parameter-restricted to compare protocols. To this end, we specified that the goal was to maximize (throughput) (delay). This goal is a good match for applications such as video streaming or voice over IP that are able to bear high throughput and low latencies.

Categories and Subject Descriptors:
C.2.2 [Computer-Communication Networks]: Network Protocols

Total Pages: 2

Omniscient

Sprout

Skype

Delay (ms @ 95th percentile)

Throughput (Mbps)
Fundamental Design Issues for the Future Internet

Scott Shenker, Member, IEEE
JSAC 1995

By what criteria do we evaluate a particular network architecture? The Internet was designed to meet the needs of users, and so any evaluative criteria must reduce, in essence, to the following question: how happy does this architecture make the users? Network performance must not be measured in terms of network-centric quantities like utilization, dropped packets, or power, but rather should be evaluated solely in terms of the degree to which the network satisfies the service requirements of each user's applications. For instance, if a particular application cares more about throughput than delay, or vice-versa, the network service to that application should be evaluated accordingly.
Measurable App-level Objectives

\[ \max U = \log(\text{throughput}) - \delta \log(q\text{delay}) \]

Remy (TCP ex Machina, Winstein & B, SIGCOMM 13)
Copa: Inspired by aiming to beat the computer

\[ \text{max } U = \log(\text{throughput}) - \delta \log(q\text{delay}) \]

- Intuition: target rate of \( 1/(\delta \cdot q\text{delay}) \) maximizes \( U \) under certain conditions
- Hard to estimate \( q\text{delay} \) in real networks (see NSDI 18 paper)
- Works well in production networks (Facebook)

V. Arun & B, NSDI 18
ABC: Accel-Brake Control

- Base station marks 1 on ECN bit if rate < target else marks 0 on ECN bit
- Sender increases by 1 on ECN “1” and decreases by 1 on ECN “0”

P. Goyal, R. Netravali, et al., NSDI 20
Interactive Gaming

Cubic
Game lags

ABC
Game runs stall free!
Fastpass

- Burst Control
- Low Latency
- Multiple Objectives
- Multipath

Centralize all control via arbiter
Aim for zero queueing

SIGCOMM 14
How Resilient are Applications?

I’m still waiting for the MIT CSAIL email!
Freezing video calls among Top 10 Zoom Bugbears

Work from home Brits have had to put up with nearly a quarter of their video calls freezing and 15 per cent cutting out altogether – just because of bad Wi-Fi, a study has found.
Whither Apps?

- App architectures have become extremely complex over the past 25 years
- Software evolution
  - Modules, became
  - Components, became
  - Microservices, which then
  - spread not only across one, but multiple, Datacenters, and then
  - and over the network to the “Edges”,
  - and to the Ends (mobile, desktop, IoT, cyber-physical systems)
Death by a 100,000 distributed microservices?

- Each call to a microservice is a potential fault
  - Ironically, handling others’ errors makes each microservice more complex
- Components and services are distributed across multiple datacenters, across the network, and across organizations
- How can we achieve resilience?
Solutions?

• Massive logging, tracing, and telemetry?
• Software-Defined X?
• AI to the rescue?
• Formal verification?
• Hyperscalar giants?
Looks like a large network!
“Internet Thinking”

- Scale: size, heterogeneity
- Feedback-based networked control and routing systems
  - With clever and simple signals
  - Placement of function principles
- Principles for competitive cooperation ("co-opetition")
Internet Thinking to Improve Application Architectures

- Each node produces a status update periodically and makes it available
- Key idea: a small number of bits (1, 2, 4, maybe 8)
- Intentionally not expressive about semantics: don’t say what the state means exactly, but convey high-level meaning (“red”, “green”, “yellow”)
- Narrow interface for information exchange, but millions of ways of computing it
- I think this is an attribute shared by ideas that succeed in large networked systems
Hide Complexity Behind Narrow Interfaces

• The IP layer is itself an example of a narrow universal interface
• But the control side is more illustrative
• Take, for example, Explicit Congestion Notification (ECN) v. eXplicit Congestion-control Protocol (XCP)
  • ECN as a mechanism can be used in any many ways (RED, PIE, CoDel, DCTCP)
  • XCP is certainly richer in information
  • But fewer bits don’t imply worse performance: cf. XCP v. ABC example from earlier
• Fewer bits of information can hide wider range of component behavior (cf. V. Arun’s talk in the next session)
The SIGCOMM Community

- Exceptionally welcoming and supportive
- The Internet is a massive success, so a sense of being the “gatekeepers”
  - Even though this hasn’t been true for years, and perhaps was never true
- Excessively high bar for deployability in papers
  - Hard to achieve both innovation and deployability
Our Community is Expanding Fast

All SIGCOMM-sponsored events

SIGCOMM conference
Paper Acceptance Rates Have Increased

More authors per paper
The SIGCOMM Community

- Getting more difficult for students – the next gen is our future
  - Acceptance rate of 33% means 3 attempts before publishing
  - Reject? Retransmit. And maybe submit something else too.
  - Capacity hasn’t changed much
  - Reviews don’t have memory
  - For a community that invented congestion control, we could do better!
- Research should favor ideas over whether it is useful today
  - PCs really can’t predict the future; only time will tell
- Be more forgiving about whether a paper solves a “real problem”
The SIGCOMM Community: Some Ideas

• PCs should favor an interesting ideas over whether it is useful today
  • PCs can’t predict the future; only time will tell
  • Be more forgiving about whether a paper solves a “real problem”, but of course correctness is important
• Review with memory
• Remove the end-January stress – rolling deadlines *a la* VLDB
• Accept more papers, now that we can record and store everything
Many thanks to my students and postdocs!

- David Andersen
- Venkat Arun
- Arjun Balasingam
- Magdalena Balazinska
- Deepak Bansal
- Frank Cangialosi
- Tiffany Chen
- Shuo Deng
- Nick Feamster
- Prateesh Goyal
- Songtao He
- Wendi Heinzelman
- Bret Hull
- Peter Iannucci
- Kyle Jamieson
- Jaecon Jung
- Katrina LaCurts
- Allen Miu
- Akshay Narayan
- Ravi Netravali
- Amy Ousterhout
- Jonathan Perry
- Raluca Ada Popa
- Bodhi Priyantha
- Asfandyar Qureshi
- Lenin Sivalingam
- Anirudh Sivaraman
- Jayashree Subramanian
- Alex Snoeren
- Arvind Thiagarajan
- Mythili Vutukuru
- Mike Walfish
- Keith Winstein
- Jakob Eriksson
- Lewis Girod
- Ramki Gummadi
- Emre Koksal
- Evdokia Nikolova
- Radhika Mittal
- Rohan Murty
- Calvin Newport
- Srinivas Narayana
- Geoff Werner-Challen

And to Sheila Marian, Dorothy Curtis, Michel Goraczko
And to my collaborators and mentors!

- Randy Katz
- Srini Seshan
- Venkat Padmanabhan
- Bob Brodersen
- Sally Floyd
- Elan Amir
- Mark Stemm
- Steve McCanne
- Suchi Raman
- Frans Kaashoek
- Mohammad Alizadeh
- Sam Madden
- Robert Morris
- David Karger
- Anantha Chandrakasan
- John Guttag
- Seth Teller
- Dina Katabi
- Arvind
- Devavrat Shah
- Scott Shenker
- Ion Stoica
- Victor Bahl
- Vern Paxson
- Jennifer Rexford
- Sylvia Ratnasamy
- Bruce Maggs
- Jeff Mogul
- James Mickens
- Aditya Akella
- Tom Anderson
- Deborah Estrin
- Nick McKeown
- Amin Vahdat
- Bruce Davie
- Manya Ghobadi
- Fadel Adib
- Sachin Katti
Conclusion: Apps Inspire Research

- App-layer overlays can improve resilience
- Flat names and key-value are powerful for distributed applications
- Frame congestion control as an enabler for application adaptation, rather than as an approach to resource management
- Get out of the data path as much as possible!
- Narrow interfaces enable scale and resilience – apply to app architectures?
- Over-index on ideas; question your prior on whether problem is “real”