Computer Networks
CS 552

Badri Nath
Rutgers University
badri@cs.rutgers.edu

About us: Management

- Professor: Badri Nath
  - http://www.cs.rutgers.edu/~badri
  - badri@cs.rutgers.edu
  - Office hours: Wednesday 1:30 : 3:30 PM
- Course info
  - http://www.cs.rutgers.edu/~badri/552.html

Course Web Page

- Course schedule
- Reading list
- Lecture notes
- Announcements
- Assignments
- Project ideas
- Exams

Sakai Web page

- https://sakai
- Will submit reviews online
- Course announcements
- Written Homeworks
Course Goals

- Understand the basic principles of computer networks, in particular the Internet
- Study new concepts, design principles in network protocols and design
- How to do network research
  - How to determine what is important
  - What are the trends
    - Datacenter, cloud, SDN, connected devices/home, connected vehicles
  - What are the economics, technology that is driving innovation
    - Cost, performance, energy, availability, security
    - Sharing Economy: airbnb, uber

Course Materials

- Research papers
  - Links to pdf on Web page
  - Combination of classic and recent work
  - ~30 papers
  - Optional readings
- Recommended textbooks
  - For students not familiar with networking
    - Peterson & Davie (4th edition)
  - Alternative: Kurose & Ross

Reading papers

- Understand /identify the basic idea
  - What is the problem that the paper tackles?
- What kind of a paper?
  - Performance, vision, new direction/protocol paper
- Summarize key idea
- +ve aspects of the paper
  - New, breakthrough, incremental,
- -ve aspects of the paper
  - Readability, Assumptions (valid?), scaling issues (does it scale), implementation (has it been implemented), measurements (problems?)

Books

- Unix Network Programming: Networking APIs: Sockets and XTI (Volume 1) by W. Richard Stevens.
- I or 2 recommended
Grading

- 20% Paper summaries/reviews/HWs based on Papers
  - A subset of the papers will be assigned for submitting summary/critique
  - All papers assigned should be read as quizzes/Hws will be based on these papers
- 30% Programming project (two-person)
- 20% Mid term
- 30% Final

Honor code
- All submitted work should be yours
- You are all grad students!!

Class Coverage

- Quick overview of undergraduate networking
- Pre requisite: 352 or equivalent
- Students expected to know
  - Link layer, basic IP routing, TCP,
- Focus on Advanced topics in networking (from papers in recent SIGCOMM, NSDI)
- Course will deal with:
  - Services and Protocols
  - Investigate protocol trade-offs, cost models
  - New Workloads, new technologies, new services

Class etiquette

- Cell phones in off position
- No FB status updates in class
- If you need to surf while in class (I prefer not), do not disturb your neighbors
- Stop me anytime to ask questions
  - Prof may not know the answer!!
- This is a graduate class, student participation in class is important
  - Challenge the class, the prof, and ideas in papers

Lecture Topics

Traditional
- Layering
- Internet architecture
- Routing (IP, BGP)
- Transport (TCP)
- Protocols (HTTP,DHCP,DNS)

Recent Topics
- Latency, $$, Energy Considerations
- Internet Architecture
- Measurements
- CDN/Video
- Datacenter networking
- Cloud Services, Metrics
- Software defined networking
**What is a Network?**

- Carrier of information between 2 or more entities
- Some carry objects/people (postal, air, surface transport)
- Most important is the services offered
- User expectation of service
  - Latency, cost, reliability, service interface, others
- We focus on computer networks
- Interconnection may be any medium capable of communicating information:
  - Copper wire
  - Lasers (optic fibre)
  - Microwave
  - Cable (coax)
  - Wireless
  - Satellite link
- Example: Ethernet, WiFI, 3G, LTE

**Why Networks?**

- Availability of Resources
  - Resources become available regardless of the user’s physical location (server based, peer2peer)
- Load Sharing/utilization
  - Jobs processed on least crowded machine
  - Resources can be shared
  - Led to cloud services
    - SaaS, PaaS, IaaS
- High Reliability
  - Alternative source of supply (multiple copies)
- Human-to-Human Communication
  - e.g., on-line world, e-commerce

**What is Internet Technology?**

- What is an internet?
  - Network of networks
- What is the Internet?
  - A global internet based on the IP protocol
- To what does “Internet technology” refer?
  - Architecture, services, interfaces, and protocols

**Domain name growth**

**What is a Network?**

- Carrier of information between 2 or more entities
- Some carry objects/people (postal, air, surface transport)
- Most important is the services offered
- User expectation of service
  - Latency, cost, reliability, service interface, others
- We focus on computer networks
- Interconnection may be any medium capable of communicating information:
  - Copper wire
  - Lasers (optic fibre)
  - Microwave
  - Cable (coax)
  - Wireless
  - Satellite link
- Example: Ethernet, WiFI, 3G, LTE

**Why Networks?**

- Availability of Resources
  - Resources become available regardless of the user’s physical location (server based, peer2peer)
- Load Sharing/utilization
  - Jobs processed on least crowded machine
  - Resources can be shared
  - Led to cloud services
    - SaaS, PaaS, IaaS
- High Reliability
  - Alternative source of supply (multiple copies)
- Human-to-Human Communication
  - e.g., on-line world, e-commerce

**What is Internet Technology?**

- What is an internet?
  - Network of networks
- What is the Internet?
  - A global internet based on the IP protocol
- To what does “Internet technology” refer?
  - Architecture, services, interfaces, and protocols
Sample Internet Applications

- Electronic mail
- WEB
- File transfer, sharing
- Social networks (FB, linkedin, twitter)
- On-line commerce
- Search
- Resource distribution (hosting)
- Video streaming
- Games

Impact of the Net on People

- Access to remote information
  - HW assignments from sakai
  - Stock quotes from financial web site
  - Corporate video, news clips, virtual tours
  - Virtual tours of homes, tourist spots, virtual globetrotting
  - Cloud services
- Person to person and group communication
  - email, collaborative tools (chat groups), Whatsapp, online social networks (FB), twitter, instagram, snapchat
- Interactive entertainment
  - youtube, netflix, hulu, music, itunes, soundcloud, spotify

Net Usage in Society

- The good
  - Access to information (i-commerce), selling goods and services (e-commerce), incredible productivity tool, unified communication tool
- The bad
  - gossip, too much information, net addicts (FB status updates!)
- The ugly
  - Fraud, pornography, threatening e-mail
- But, it is just a mirror of society
Impact on society

- Net neutrality
- Laws and censorship (SOPA ---)
  - Google vs China
  - Wiki going black -- protest web censorship
- Nations’ laws and Internet
  - Regulation
- Content creation, ownership, distribution, online piracy
- Cyber Warfare – N Korea, Sony

Internet Players

- Users, people who use the applications
  - Everyone (mom and pop, kids)
  - get something done (hopefully useful)
- Designers
  - You: protocol design and implementation
- Service Providers
  - Administrators and ISPs
  - Datacenter operators
  - Provider-customer versus peer to peer
  - Management, revenue, deployment
- Market/business models for the Internet
  - Consumer to consumer (eBay, Match.com, craigslist, airbnb), Business to consumer (Amazon, eBay, Google, Netflix, Hulu), Business to business (getty, Harvest, Google), Consumer to business (hotjobs, monster, LinkedIn)

Internet Growth

- Mary Meeker: Internet trends May 2015
WWW growth

Source: Internet trends May 2013, Mary Meeker

Facebook growth

MAU: Number of unique users in the past 30 days

Social Networking

Gossip beats learning

Source: Experian Hitwise
Initially: Only kids!!

Now: everyone; grandpa, grandma

Social network usage

Messaging Continues to Grow Rapidly...
Leaders = WhatsApp / Facebook Messenger / WeChat

Monthly Active Users on Select Social Networks and Messengers, Global, 2011 – 2015

Source: Social networking sites in this chart include Facebook, LinkedIn, Twitter, Instagram, Pinterest, and Google+. Facebook data per Facebook.
Mobile Phone usage

1995
80MM+ Mobile Phone Users
1% Population Penetration

2014
5.2B Mobile Phone Users
73% Population Penetration

Mobile OS trends

Global Smartphone Operating System Market Share
(by Units Shipped), 2005 vs. 2010 vs. 2013

Communication trends

Can nodes be selected dynamically?

Video Content Growth

Mary Meeker Internet trends 2013
Reimagining shared economy

**Renting a place**
- Airbnb

**Hailing a cab**
- Uber

**Renting your car**
- Flightcar

**On demand food delivery**

**Supply/Demand Exchanges**
Objective of networking

- End-hosts to communicate
  - Applications running on end-hosts
- Different technologies
- Different protocols
- Different Services

How to communicate?

- Circuit switching
  - Establish a connection before communicating
    - POTS (plain old telephone system)
    - Dedicated pipe for the duration of the session
- Packet switching
  - Multiplex communication from different sources
  - Every packet is self contained
  - Efficient use of resources
  - NO guarantees on performance

What’s hot?

- Internet
- Deep Learning
- Data centers
- Edge vs Cloud
- Voice Interface to X
- Software Defined Networking (SDN)
- Internet of Things (IoT)
- Internet Bodily of Things (IoB)
- Vehicular Networking
- Driverless Cars
- Alexa, What’s happening to my network?
How to handle different networks?

- Many differences between networks
- How to translate between various network technologies?
- Have a common protocol for inter network communication
  - IP
  - A set of rules with a well-defined interface

How to locate a node?

- Naming, discovery and routing
  - Network elements needed to support directory
  - Network elements needed to support forwarding towards destination
  - Scalable
  - Reliable

How to meet application demands?

- Corruption?
  - Need error detection and correction
- Reliability
  - Data lost?
- Overload
  - Congestion control
- Security
  - Encryption, authentication

Lots of Functions Needed

- Link
- Multiplexing
- Routing
- Addressing/naming (locating peers)
- Reliability
- Flow control
- Fragmentation
- Etc....
ISO OSI Layering Architecture

- Application
  - Presentation
  - Session
  - Transport
  - Network
  - Data Link
  - Physical

- Host A
  - Application Protocol
  - Presentation Protocol
  - Session Protocol
  - Transport Protocol
  - Network Protocol
  - Data Link Protocol
  - Physical Protocol

- Host B
  - Application Protocol
  - Presentation Protocol
  - Session Protocol
  - Transport Protocol
  - Network Protocol
  - Data Link Protocol
  - Physical Protocol

Problems

- Seven layers not widely accepted
- Standardized before implemented
- Top three layers fuzzy
- Internet or TCP/IP layering widespread

TCP/IP Layering Architecture

- Application
  - Transport (TCP/UDP)
  - Internet
  - Host-to-Net

- A simplified model
- The network layer:
  - Hosts drop packets into this layer, layer routes towards destination—only promise—try my best
- The transport layer:
  - Reliable/unreliable byte-oriented stream

Internet design philosophy

- #1 Functionality at the edge as opposed to core
- In Telephone network it is the opposite:
  - Any new service, the phone company has to provide
  - Edge device is dumb
- Smart device at the edge means programmability
- New services can be supported, drives innovation:
  - VOIP (SIP), IM
- Cathedral vs Bazaar

David Clark, The design philosophy of the DARPA internet protocols, 1998
Implications (cathedral vs bazaar)

- Bazaar
  - Edge is programmable
  - Nimble, novel applications

- Cathedral
  - Core elements still rigid
  - Standards, slow evolution
  - Can’t do anything radical
  - Where is programmability in the system?

# 2 Multiplexed utilization

- Best effort, packet switching
- Keep the network simple
  - Packets may be lost, corrupted, out-of-order
  - Let the end host implement any other requirements
  - Want reliability?
    - Retransmit from sender
- Packets self contained
  - Can take different routes
  - Different transfers on the same link
- Stateless in the core
  - End hosts can maintain state
  - Fate-sharing (If I die, my state will die but will not affect others)

# 3 Support multiple networks

- IP over anything, anything over IP
  - Run over any type of link
  - Build any end-to-end protocol over IP

TCP/IP Layering Architecture

- IP over anything, anything over IP
  - Build any end-to-end protocol over IP

Diagram: TCP/IP Layering Architecture
#4 distributed management

- Need only know local information
- Information distributed over different nodes
  - Scalable
  - No single hot spot
- Distributed functionality -- Roles
  - Different entities manage different parts of the system
  - Impact on naming, routing, addressing
- Local and Global management authorities

Traditional Internet Model

A new Internet model

Facebook DC architecture by Nathan Farrington et al
## Internet Design Principles

- **Scale**
  - Protocols should work in networks of all sizes and distances
- **Incremental deployment**
  - New protocols need to be deployed gradually
- **Heterogeneity**
  - Different technologies, autonomous organizations
- **End-to-end argument**
  - Networking functions should be delegated to the edges; application knows best

---

## End-to-end argument

- **Saltzer, reed and clark [1984]**
  - End-to-end arguments in system design
  - Main idea
    - If a function can only be completely and correctly implemented with the knowledge and help of the applications standing at the communication end points. Hence providing this function in the subsystem is not possible
    - Complexity at the edges as opposed to the core
    - Simply stated, the argument suggests that functions placed at the low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level.
    - Don’t force feature, service, restriction on the end points

---

## Communication system

- An end system connected by a communication subsystem
- Questions?
- Who is responsible for a given function
  - Subsystem?
  - End units?
  - Or both (redundant) or jointly?

---

## End-to-end argument

- Functions placed at lower level implies specific problems being solved in a general way
- Best aim:
  - Simple lower layer with smart end points
    - Basic and general functions at the lower layers
    - Gives flexibility
**e2e argument**

- Low level system may not have all the information to implement the given functionality
- Implement only for performance (wireless links)
- Low level system shared by all applications – what if the application does not need the feature
- Performance
- Duplicated effort
- Should not impact applications that do not use that functionality

**e2e tradeoffs**

- New business models
  - Network caching, redirection, proxy transcoding
- Wireless Application protocols
  - Gateway provides a box for content translation
- Network redirection
  - Network level switch for load balancing
- Balance between performance, layering, e2e argument

**New metrics**

- Energy/power
  - Always-on system consumes a lot of power
- System Performance/availability
  - How to guarantee performance on shared infrastructure?
  - Latency, throughput, availability
  - How to measure?
- Maintenance
  - Hardware cost is falling, long term human cost of admin is increasing
  - Opex vs capex debate
  - Cloud/Software as a Service (SaaS) models

**Typical Datacenter networking**

The cost of a cloud: Research Problems in data center networks by Albert Greenberg et al., CCR
Data center traffic (Elephant vs Mice)

- 50% of flows < 100 KB
- <5% of bytes

- 5% of flows > 10 MB
- 30% of bytes

Kandula-IMC2009
Alizadeh-pFabric: near-optimal data transport SIGCOMM13,

DC network evolution

Facts and Figures [Quereshi09]

- Servers are power hungry (annual electricity bills)

<table>
<thead>
<tr>
<th>Company</th>
<th>Servers</th>
<th>Electricity (MWh)</th>
<th>Cost (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>eBay</td>
<td>16K</td>
<td>~0.8 x 10^10</td>
<td>~$117M</td>
</tr>
<tr>
<td>Akamai</td>
<td>40K</td>
<td>~1.7 x 10^10</td>
<td>~$10M</td>
</tr>
<tr>
<td>Rackspace</td>
<td>50K</td>
<td>~2.9 x 10^10</td>
<td>~$12M</td>
</tr>
<tr>
<td>Microsoft</td>
<td>&gt;500K</td>
<td>&gt;5 x 10^10</td>
<td>&gt;$63M</td>
</tr>
<tr>
<td>Google</td>
<td>&gt;500K</td>
<td>&gt;6.8 x 10^10</td>
<td>&gt;$83M</td>
</tr>
<tr>
<td>USA (2006)</td>
<td>10.9M</td>
<td>610 x 10^6</td>
<td>$4.5B</td>
</tr>
<tr>
<td>MIT campus</td>
<td>2.7 x 10^6</td>
<td>829M</td>
<td></td>
</tr>
</tbody>
</table>

Datacenter cost

- Servers 45%
- Power 15%
- Infrastructure 25%
- Network 15%

- 50,000 server @ 3K a pop, 5% cost of money, 3YR
- 52.5 MWh cost
- Power cost
- Power to run the IT equipment
- Power to run cooling, UPS etc - Overhead
- PUE=Total power /IT power
- 1.2 ideal – 20% overhead
- Typically 2 to 3 PUE – Air conditioning costs enormous
Energy-proportional metric

- Can we design networks that consume power proportional to utilization?

Throughput proportional fabric

- Does throughput offered rise/fall in proportion to traffic sources/sinks
- Fat-free topologies paper in Hotnets 2016

Energy Proportional Computing

Figure 1. Average CPU utilization of more than 5,000 servers during a six-month period. Servers are rarely completely idle and seldom operate near their maximum utilization, instead operating most of the time at between 10 and 50 percent of their maximum.

Throughput proportional fabric

Energy Proportional Computing

Energy Efficiency = Utilization/Power

Figure 2. Server power usage and energy efficiency at varying utilization levels, from idle to peak performance. Even an energy-efficient server still consumes about half its full power when doing virtually no work.
Energy Proportional Computing

Can we do this for Networking Infrastructure?

Energy Efficiency = Utilization/Power

Figure 4. Power usage and energy efficiency in a more energy-proportional server. This server has a power efficiency of more than 80 percent of its peak value for utilizations of 30 percent and above, with efficiency remaining above 50 percent for utilization levels as low as 10 percent.

Energy/ Power metric

- If Cost/Green Conscious
- Make network elements less power hungry
- Take advantage of lower power rates elsewhere, time-of-day

What is the energy consumption of the internet?

- Energy consumed by networking equipment such as routers, switches, hubs etc
- Does not include hosts
- Internet energy consumption controversial data

Facts and Figures [Gupta & Singh-03]

<table>
<thead>
<tr>
<th>Device</th>
<th>Approximate Number Deployed</th>
<th>Total AEC TW-h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hubs</td>
<td>93.5 Million</td>
<td>1.6 TW-h</td>
</tr>
<tr>
<td>LAN Switch</td>
<td>95,000</td>
<td>3.2 TW-h</td>
</tr>
<tr>
<td>WAN Switch</td>
<td>50,000</td>
<td>0.15TW-h</td>
</tr>
<tr>
<td>Router</td>
<td>3,257</td>
<td>1.1 TW-h</td>
</tr>
</tbody>
</table>

\[ P/E = 1W = 1 J/S 1 Kw-h = 3600000 joules = 10 x100W bulbs for 1 hr \]
More Numbers...

- Total energy consumed by networking devices annually in 2000 (US): 6.05 TW-h
- Amounts to about 0.07% of total U.S. energy expenditure
- Expected increase: +1 TW-h by 2005
  > Note: This does not include energy consumed by hosts, UPS supplies or cooling equipment.

Enterprise networks

- Network Switches hubs routers under utilized
- Many units not energy proportional
- Turn off ports in proportion to b/w demands
- Timescales?

So, why bother saving energy?

3 reasons:
1. Current energy inefficiencies
   a) Wired vs. wireless energy costs
   b) 6 TW-h ~ 1 nuclear reactor
   c) Extrapolate to World ~140 nuclear reactors
2. Enable greater deployment
   a) Similar connectivity in India would require 4.75% of total energy budget
3. Enable longer operation times during events of disaster
   a) Recent Grid failure in NE US/Canada
   b) Frequent power outages in most of the world

How to save energy?

- Low-Energy Hardware Design:
  - Use hardware components with low power modes of operation
  - Lower the clock frequency of the components, use DVS and other methods during low demand
  - Energy-Aware APIs – give control to software
  - Architecture that allows selective powering off

- Energy-Aware Protocol Design:
  - Node-level algorithms for sleeping
  - Route aggregation and other global techniques to inform devices when and for how long to sleep
Where to save energy in a device?

- Memory
- Main CPU
- Switch fabric or bus backplane
- Line cards (designs ranging from simple to complex with ASICs or network processors to process packets)
- Bang for the buck

### Key Questions

- For how long can components sleep?
  - state transition times, energy spike
- How is the decision to sleep taken?
  - traffic activity level, in isolation (uncoordinated), global (coordinated), edge or backbone device, transit or stub network
- How to distinguish sleep vs. failure?
  - should not trigger network reconfiguration in sleep state vs. failure state
- How to wake up a device?
  - at fixed intervals, on packet arrival, account for protocol timers
- Impact on protocol behavior?
  - long sleep times, slower propagation of topology changes

More on sleeping...

**Uncoordinated sleeping:**
- Decision to sleep based on individual traffic levels alone
- Inform nearest neighbors only
- Sleep time limited by protocol hello message timer
- May trigger network reconfiguration in case of missed protocol messages

**Coordinated sleeping:**
- Decision taken on a network-wide basis
- Need algorithm to pre-compute the optimal sleep time, but computation costs increase
- Hello message frequency can be adjusted, may take longer to detect changes in network topology
- Sleep longer intervals, but forwarding tables may be outdated
- Reroute all traffic through one route, shutdown other routes
- Introduces delay and packet loss in case of sudden traffic burst

Computation Placement

- Energy cost varies by location
- Energy cost varies by hour of the day
- Can we push computation to a geographically distant place to save energy?
- Can we exploit time difference?
- Peak vs non-peak power rates
Exploit spatial-temporal price

- Prices vary from place to place
  - Coal vs nuclear
- Price varies with time of day
  - Peak vs off-peak
- Prices vary from place to place
- Coal vs nuclear
- Price varies with time of day
- Peak vs off-peak

Research

- Modification of protocols at layers 2 and 3 to incorporate sleep modes
- Study impact of modifications on end-to-end delay and performance
- Develop energy models for routers and switches
- Study the algorithmic problems of how/when/how long to sleep
- Load migration: tolerable latency vs cost savings

The tail at scale

- Curse of the long tail
- 99% of the requests finish < 10 msec
- 1% of the request > 1 sec
- A large fan out, more request will need > 1 sec
- Touch 100 servers: prob not one of them is cursed = 0.99^{100} \approx 0.37
- 63% of the requests take \geq 1 sec

Variability in latency

- Shared resources
- Background jobs
- Queuing
- Maintenance jobs (check pointing)
- Large fanout multiplier effect
Use redundancy

- Redundancy effectively used to improve availability (RAID, replicated servers)
- Now use to reduce variability in latency
- Response time important in many servers
  - Word completion, spell check, document selection
- Idea: Send requests to multiple servers
- Pro: Hit one server with low latency
- Con: more resources consumed due to duplicate work (increase in queuing delay)
- Challenge: reduce latency without undue overhead

Hedged requests

- Send request to multiple servers
- Cancel the other after first response
- Techniques to reduce overhead:
  - Send second after some delay (d > percentile)
  - Send second at lower priority

Tied requests

- Send request to multiple servers (with id of replicas)
- Server cancels other requests when the request is scheduled
- Still both requests may be executed (queue empty)
- Techniques to reduce overhead
- Send second after some delay (d > RTT of message in network)

Results

- Hedged requests: 100 servers
  - Send hedged request after 10 msec delay
  - Reduced 99.9 percentile latency from 1800 msec to 74 msec
  - Sends only 2% more requests
- Tied requests
  - 40% reduction at the 99.9 percentile latency
- Key takeaway: Predictability from unpredictable parts
- Can we do this for network security?
Computing cost declining: own or rent

- Rutgers pays about 30K a month for ISPs

Cloud services

- Amazon web services
- Google cloud
- Microsoft Azure

Amazon cloud pricing

- **On-demand**: pay by the hour, no upfront payment (0.023 per hour to 0.094 per hour)
  - #CPUs, memory size, disk space etc varies
- **Spot pricing**
  - Bid for space capacity, bid price< spot price, process is preempted
- **Reserved Pricing**
  - 1 Yr to 3 yr commitment; upfront payment
  - Large instances $549 plus 0.063 per hour

Cost of cloud

- Many cost models
- Storage (Amazon S3)
  - Lease: 1 TB/Month 3$ regular; 1.25$ infrequent; Glacier (archive) 0.70$
  - Buy: 1 TB 60$
- Cost models for Database, management, storage, network, computation
- Data mining/analytics (rapid miner)
  - Price proportional to number of tuples used
New Metrics

Cost $\$

- Cost of Computing
  - Trying to figure how much it is going to cost
  - Bidding for cloud services
  - Computation (AWS, AZURE, B/W, Storage (box, dropbox))
  - How to bid the cloud [Liang, Zheng et al. SIGCOMM 2014]
  - Need protocols that reduce burden on end-user

Selling Scale

Buy

Rent

Cost continuum tools

- Can we build tools that can answer questions like
  - What is the cost for doing X?
  - What should I own, rent, or bid?
  - How should I distribute my computation for a given cost model?
  - What is the performance impact?
    - Cost vs accuracy, cost vs latency, cost vs availability
  - Replace CPU with any other resource: storage, DB, B/W, management etc
  - Edge vs Cloud — “cache” to save dollars

Bid strategies

- One time bids
  - Present once, exit if it falls below current market price
  - Risk: job may not complete,

- Persistent bids
  - Resubmit each time period until job finishes

- Hybrid
  - Important nodes one time bid and slaves persistent bids (e.g., map-reduce)
  - Requires transparent provider pricing

Current Network

- Routing, management, mobility management, access control, VLANs, ...

- Million of lines of source code
- 5400 RFCs
- Barrier to entry

- Billions of gates
- Blasted

- Power Hungry

Many complex functions baked into the infrastructure

- OSPF, BGP, multicast, differentiated services,
- Traffic Engineering, NAT, firewalls, MPLS, redundant layers, ...

An industry with a “mainframe-mentality”, reluctant to change

Slides from Stanford Site: Nick McKeown, Martin Casado, Scott Shenker et al,
The SDN Approach

Separate control from the datapath
- i.e. separate policy from mechanism

Datapath: Define minimal network instruction set
- A set of “plumbing primitives”
- A vendor-agnostic interface: OpenFlow

Control: Define a network-wide OS
- An API that others can develop on

Flow Table

Flow Table Entry
"Type 0" OpenFlow Switch

Conventional | SDN
---|---
Application Network OS ASIC | Application
North bound API | Controller
Controller: OpenFlow, POE, OpenSwitch
Switch OS | Switch I/U

Flow Table

<table>
<thead>
<tr>
<th>Match fields</th>
<th>Priority</th>
<th>Counters</th>
<th>Instructions</th>
<th>Timeouts</th>
<th>Cookie</th>
</tr>
</thead>
</table>

At each table, action set is constructed; at the end action set is executed

Rule | Action | Stats
---|---|---

Packet + byte counters

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline

Switch Port | MAC src | MAC dst | Eth type | VLAN ID | IP Src | IP Dst | IP Prot | TCP sport | TCP dport |
---|---|---|---|---|---|---|---|---|---|

Source: Open flow switch specification Open network Foundation
### Examples

**Switching**

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Ether Type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP sport</th>
<th>TCP port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>port6</td>
</tr>
</tbody>
</table>

**Flow Switching**

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Ether Type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP sport</th>
<th>TCP port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>port3</td>
<td>00:20..</td>
<td>00:1f..</td>
<td>0800</td>
<td>vlan1</td>
<td>1.2.3.4</td>
<td>5.6.7.8</td>
<td>4</td>
<td>17264</td>
<td>80</td>
</tr>
</tbody>
</table>

**Firewall**

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Ether Type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP sport</th>
<th>TCP port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>22</td>
</tr>
</tbody>
</table>

### Routing

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Ether Type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP sport</th>
<th>TCP port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>port6</td>
</tr>
</tbody>
</table>

**VLAN Switching**

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Ether Type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP sport</th>
<th>TCP port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>port6</td>
</tr>
</tbody>
</table>

### SDN use cases

- Global traffic engineering
- B4, SWAN
- Network function virtualization
  - Allow multiple functionalities implemented on Hardware
  - Made programmable by SDN

### Boxes, boxes everywhere

- Load balancer
- Firewall
- Wireless Router
- Controller
- NAT box
- Component Flow Tables
SDN issues

- Correctness, Consistency, Configuration
- Policy to flow rules
- Controller robustness
  - Security, availability
- Management plane

Research

- Machine Learning in networks
  - Network functions
  - Edge learning
  - Smart control plane
  - Deep learning
- Querying vs look up
  - Bots for network management
  - Small data vs Big data
- Network defined economy
  - Shared resources
  - Smart cities