Next Two Lectures

- Wednesday:
  - Clark: “The Design Philosophy....”
  - Saltzer, Reed, and Clark: “End-to-end arguments....”
- Monday:
  - Cerf and Kahn: “A Protocol for...”
- Remember to do your summaries!
- Reading list will be finalized over weekend....

Overview

- Will present ~35 short project suggestions
- Legend: based on how well-defined projects are, not necessary how difficult they are
  - ● Well-defined projects
  - □ Less-defined project
  - ◆ You need to define project’s goals
- Need to send us a one page proposal by Feb 7
  - Feel free to talk with us beforehand!

Outline

- Traditional networking
- Slightly nontraditional networking
- Distributed Hash Tables
- New Architectures and Paradigms
- Theory
- Identity-based Architectures

RED: Does it Really Help?

- Random Early Drop is the first and most widely used “active queue management” algorithm.
- Its goal is to promote fairness and decrease queue lengths (delays).
- Does it really help? There have been several contradictory papers on this.
- What is the real story?

Quickstart+TCP vs XCP

- XCP (Katabi et al.) is a recent congestion control proposal (we’ll cover it later) that requires dramatic changes in TCP and routers
- Quickstart is a quick-and-dirty hack:
  [http://www.icir.org/floyd/quickstart.html](http://www.icir.org/floyd/quickstart.html)
- Is XCP significantly better?
Burst Switching

- Two main communication models
  - Datagrams: each packet is individually switched (routed)
  - Circuits: a circuit is set-up and all packets are forwarded

- Hybrid model: burst switching
  - First packet describes how many packets are in a burst
  - Router decides whether to forward all packets in the burst or none of them

- Research
  - Design a burst switching protocol and study its trade-offs

Interdomain Traffic Engineering

- Interdomain traffic engineering is a mess:
  - Ambiguous goals
  - Ad hoc techniques

- The best known paper on this is "Guidelines or Interdomain Traffic Engineering" by Feamster et al.

- Can one come up with a specification language and a coherent set of mechanisms?

Redirection/Selection Strategies

- Akamai redirection uses domain information to redirect requests

- Recent work on server selection uses network coordinates (GNP, Vivaldi)

- Are coordinates significantly better?

Slightly Nontraditional Networking

Disjoint Paths vs Waypoints

- Feedback based routing (Zhu et al.) uses disjoint paths to achieve resiliency

- Would using waypoints work better?
  - Easier (no need to find disjoint path)
  - More choices

Resiliency via Incast

- Send to set of waypoints (in mcast group):

- Each waypoint forwards data toward rcvr

- Incast boxes (one or more along path) strip out extra redundancies (configurable parameter)

- How reliable does that make delivery?

- What is a coherent architecture for this?
  - i3, DOA, etc.?
### Negation Routing

- Recent proposal for Packet Obituaries (Argyraki et al., Hotnets 2004) gives feedback on which AS is dropping packets
- Additional proposal (being written up) allows one to do “negation routing” by saying: “avoid this AS”
- What is the performance of this approach?

### Sensornet Routing

- Point-to-point routing is hard in sensornets and other ad hoc networks
- In a static ad hoc network, one can build up a coordinate system by using recursive pairing
- Challenge: design such an algorithm and analyze its performance

### Reconfigurable Directional Antennae

- Lots of interest in “mesh networking”
  - Many performance problems because of interference
- What if we had reconfigurable directional antennae instead of broadcast?
- Could quickly reconfigure “links” to produce good paths
- Design such a system and analyze it

### Anycast as Evolution Mechanism

- [joint with Sylvia Ratnasamy]
- How can the Internet evolve?
- Need to give incentives for individual ISPs to deploy new versions of IP at least partially
- That requires having packet using IPvX being automatically forwarded to the nearest router supporting IPvX
- Interesting combination of technical and economic requirements

### Distributed Hash Tables

- Each data item and machine (node) in the system has associated a unique ID in a large ID space
- Hash table like interface
  - `put(id, data)`
  - `data = get(id)`
- ID space is partitioned among nodes
- Data items are stored at the node responsible for its ID

### DHTs Overview
Example: Chord

- Circular ID space \([0, 2^m - 1]\)
- Consider two consecutive nodes on the ID circle with IDs \(N_1\) and \(N_2\), where \(N_2\) follows \(N_1\):
  - Node \(N_2\) is responsible for interval \((N_1, N_2]\)
- Node 35 inserts \((37, data)\)
- Node 3 queries data with ID 37

Location Control in DHTs

- Users have NO control over where data items are located
- Advantages:
  - Users don’t need to know about individual nodes
  - System can recover in case of failure without user involvement
- Disadvantages:
  - Not efficient
  - Enforces uniform trust model
- Research: design a system in which users have "some" degree of control on where data items are located
  - E.g., "I want my items to be located only on nodes in US"

Content Routing (DHTs?)

- Gritter and Cheriton proposed a technique for "Content Routing"
- This was before the days of DHTs
- Would DHT technology (put "on-path") provide a better solution to this problem?

Circle Checks for DHTs

- DHTs provide no correctness guarantees
- However, there may be ways to check whether the DHT is giving consistent results
  - Based on a circulating packet
- Flesh out this design, and analyze
- Generalize this to other problems:
  - Don’t ensure correctness, but check it
  - Can one evade the CAP theorem?

DHT as Library vs DHT as Service

- Traditionally DHTs have been used as libraries
  - Lots of flexibility
  - But requires separate DHT for each application
- Can also use DHT as service
  - Rigid interface
  - ReDIR is client-side library that helps make this more effective
- OpenDHT (www.opendht.org) is a public DHT service

OpenDHT Projects!!!

- Reliable mcast (srn, wb)
- Top k clients
- Coral (NYU project) over OpenDHT
  - Just location-aware store
- Suspend/Resume (IRP project, but on DHTs)
- Auxiliary boxes:
  - NAT traversal (Skype?)
  - Permanent store
Peering DHTs

- For a DHT service to make sense, it needs to be commercially viable
- That means that there must be a way to provide a DHT-dialtone
- Wallfish et al. have a proposal for peering DHTs
- Many details need to be ironed out, and its performance needs to be analyzed.

New Architectures and Paradigms

DoS Prevention

- DoS Resilient Architecture
  - Separate clients from servers
  - Only servers can be directly contacted
  - Clients can be contacted only if it allows this explicitly
- Research:
  - Other alternatives to implement such architecture?
  - How well can you do in the context of the current Internet?
  - Note: can use DOA, i3 like architectures

Checkable Protocols

- Protocols that check correctness but do not guarantee it, e.g.,
  - ECN-nonce
  - Listen and Whisper
  - SV-CSFQ
- Develop other applications, e.g.,
  - Differentiated services: make differentiated service more robust to malicious/misconfigured ingress nodes

Annotation Layer

- Many protocols require nodes along path to exchange information:
  - IP traceback, quality of service, authentication
- Today’s solutions not good enough
  - Hijack bits in the packet header (e.g., fragment offset)
  - IP options: slow to process
- Proposed solution:
  - Insert annotation layer between network and transport
- Produce design and examples

Theory
CAP vs CAS

- The famous CAP theorem (easy to read) states that one cannot achieve:
  - Consistency
  - Availability
  - Ability to function while Partitioned
- Partitioning is no longer necessary
- What we really care about is C, A, and Scaling!
- Can we formulate and prove a CAS theorem?

Overlay Routing

- Assume
  - A network topology T
  - A routing algorithm running on top of T
  - You control a fraction \( f \) of nodes in T
- Question:
  - How well can you approximate an “arbitrary” routing metric as a function of \( f \) and topology T?
- Example:
  - T uses \# of hops to implement shortest path
  - You know delay distributions along links in T
  - How well can you approximate lowest latency routing metric assuming a power-law topology and \( f = 10\% \)?

Geographic Routing

- Consider a stationary ad hoc network
- Design a compact routing scheme (small routing tables)
- Require that this scheme have low incremental costs when nodes and links come/go
- Is geographic routing the only such scheme?

Identity-Based Architectures

“ID-based” Architectures

- Decouple the identity of an end-host/service from its address
- At transport level, sender sends packet to an ID, not an address
- Examples
  - Delegation Oriented Architecture (DOA) [http://nms.lcs.mit.edu/DOA]
  - Host Identity Protocol (HIP) [http://www.ietf.org/html.charters/hip-charter.html]
  - Internet Indirection Infrastructure (I3) [http://i3.cs.berkeley.edu]

“ID-based” Architectures (cont’d)

- Both the sender and receiver should be able to explicitly control the service-path
- Example:
  - Sender wants its packets to go through S1 before they reach destination
  - Receiver wants all packets to go through S2 before it receives them
**Example realization in i3 and DOA**

![Diagram showing i3 and DOA realization](image)

**Authentication and Encryption**

- Resolution: name → IDs
  - Use DNS to return receiver’s ID and eventually its public key (HIP)
  - Slow update, as secure as DNS
  - Use an address book (i3)
  - Secure but hard to maintain
  - OpenDHT
  - Highly scalable; security needs to be addressed
- Authentication and encryption
  - Public key cryptography (HIP, i3)
- Research
  - Use Identity Based Encryption (IBE); other alternatives?
  - Study trade-offs between various techniques

**Signaling Protocol for Middleboxes**

- Design a signaling protocol to accommodate middleboxes/services
- Research issues:
  - Authentication of middleboxes
  - Transparent recovery when one middlebox fails another equivalent middlebox can take over
  - Challenge: recovery transparent to end-hosts at transport layer

**ID based Transport Protocols**

- Design a congestion control mechanism (e.g. TCP) such that it is possible to change the receiving machine in the middle of the transfer!
- Scenario:
  - A and B open a connection (A receiver; B source)
  - A changes to A’
  - B continues to send data to A’ without creating a new connection
- Research: refactor transport such that
  - Congestion control state binds to address
  - Data transfer state binds to ID

**Service Differentiation via Middleboxes**

- Problem: flow isolation (UDP can kill TCP)
- Solution outline:
  - Run RR
  - Application level congestion control

**Anonymous File Sharing**

- IDs may be chosen such that not to reveal end-host identity
  - E.g., pick random IDs
- Sender doesn’t know the IP address of receiver
- You can simply use web to share files!
- Questions:
  - Anonymous search engine
  - Anonymity vs. performance
Event Notification System

- Users specify events in which they are interested as a conjunction of attributes, e.g.,
  - (stock="msr") and (share_price > 60)
  - (source=Berkeley) and (destination=North Lake Tahoe) and (time < 3.5 hours)
- Research: create an efficient delivery tree
  - Users with the same interest grouped under the same tree
  - Users in the same geographic region grouped under the same tree

Anycast / Service Location

- Direct a client to a nearby server/proxy
- Two alternatives:
  1) Unmodified client
     - Make selection based on the DNS request (at the DNS server)
     - … similar to Akamai
  2) Modified client
     - Select a “good” server/proxy in the context of OpenDHT or i3
     - Consider both the proximity and load

Efficient Multi-Level Naming

- The LFN proposal requires several layers of names.
- Done naively, this requires many lookups for a single transaction
- Can one devise techniques, such as hints, write-through, etc. to make this performance adequate?

Internet-Scale XML Dissemination Service
[Due to Yanlei Diao – next 5 slides]

- YFilter: Specification of data interests using an XML query language
- XML streams → YFilter → query results

- User queries: Specification of data interests using an XML query language
- Data sources: Continuously publish XML data items
- The service: Delivers to each user the XML data items that match her data interests. The results are presented in a format required by the user
  - Applications: News feeds, stock tickers, sports tickers, mobile services, online auction, network monitoring...

ONYX: Large-Scale XML Dissemination

- ONYX: Operator network using YFilter for XML Dissemination
- An overlay network of information brokers running YFilter
- Underlying infrastructures:
  - A dedicated network
  - Peer-to-peer
  - Collaboration among administrative domains

Content-based Routing

- Broker 2: /article/@edition.area="NY"
- Broker 4: /article/@edition.area="SF"
- Broker 6: /article/@subject/@type="Stock" or /article/@subject/@number="Timing"
- Broker 6: /article/@subject/@name="Tide Forecasts"
- Q1: /article[@@edition.area="NY"]
- Q2: /article[@@edition.area="SF"]
- Q3: /article[@@edition.area="SF"]
  .series[@@series.name="Tide Forecasts"]
Content-based Routing with I3?

- Use I3 as an alternative to content-based routing
- Basic approach:
  - On queries: Function $F_q: (Q_i) \rightarrow (ID_j)$
  - On documents: Function $F_d: (D_i) \rightarrow (ID_{j1, j2, \ldots, jn})$
- Goal: an adaptive algorithm to balance false positives delivered and routing speed.
- Also, report strengths and weaknesses of using I3

Dissemination of Results

- When every user requires customized results (e.g., my Yahoo! in a push-based fashion), how do you deliver the results?
- Bandwidth:
  - Peak load = 5000 documents per second, 1kB each
  - 1 million queries, query selectivity is 10%, result size reduction factor 10%.
  - Total result size = $5000 \times (10^6 \times 10\%) \times (1000\times8\times10\%) = 400\text{Gb/s}$
- MSDN RSS is facing a similar problem!
- Connectivity: consider mobile users
- Solutions:
  - Unicast vs. multicast?
  - Fragmenting and merging results for sharing?
  - Caching for disconnected results?
- Let Microsoft know when you solve the problem!

Next Step

- You can either choose one of the projects we discussed during this lecture, or come up with your own
- Pick your partner, and submit a one page proposal by February 7. The proposal needs to contain:
  - The problem you are solving
  - Your plan of attack with milestones and dates
  - Any special resources you may need