Overview

- Web
- Peer-to-peer
  - Motivation
  - Architectures
  - TOR
  - Skype
- CDN
- Video

Tor Anonymity Network

- Deployed onion routing network
  - http://torproject.org
  - Specifically designed for low-latency anonymous Internet communications
- Running since October 2003
  - Thousands of relay nodes, 100K-500K? of users
- Easy-to-use client proxy, integrated Web browser
  - Not like FreeNet – no data “in” TOR
- Really an overlay – not pure peer-to-peer

Based on slides by Vitaly Shmatikov

Tor Circuit Setup (1)

- Client proxy establish a symmetric session key and circuit with relay node #1
  \[ A = K(B)_k \]
- All data sent over the circuit is encrypted
Tor Circuit Setup (2)

- Client proxy extends the circuit by establishing a symmetric session key with relay node #2
  - Tunnel through relay node #1

Using a Tor Circuit

- Client applications connect and communicate over established Tor circuit
  - Datagrams decrypted at each link
- Also want end-to-end encryption – not done by Tor

Tor Circuit Setup (3)

- Client proxy extends the circuit by establishing a symmetric session key with relay node #3
  - Tunnel through relay nodes #1 and #2

Using Tor

- Many applications can share one circuit
  - Multiple TCP streams over one anonymous connection
- Tor router doesn’t need root privileges
  - Encourages people to set up their own routers
  - More participants = better anonymity for everyone
- Directory servers
  - Maintain lists of active relay nodes, their locations, current public keys, etc.
  - Control how new nodes join the network
    - “Sybil attack”: attacker creates a large number of relays
    - Directory servers’ keys ship with Tor code
Overview

- Web
- Peer-to-peer
  - Motivation
  - Architectures
  - TOR
  - Skype
- CDN
- Video

What is Skype?

- Support pc-to-pc, pc-to-phone, phone-to-pc
- VoIP and IM client communication
  - Also: conference calls, video, ...
- Developed by people who created KaZaa
  - Has peer-to-peer features that will look familiar
- A p2p illusion
  - Login server
  - Buddy-list server
  - Servers for SkypeOut and SkypeIn
  - Anonymous call minutes statistic gathering
- Challenges include dealing with NATs, firewalls and scalability

What problems does it solve?

- NAT and firewall traversal
  - Nielsen September 2005 ratings
    - 61.3% of US home internet users use broadband
      (http://www.nielsen-netratings.com/pr/pr_050928.pdf)
    - ‘Most’ users have some kind of NAT
- Calls between traditional telephone and internet devices
  - SkypeOut (pc-to-phone)
  - Terms of service: governed by the laws of Luxembourg
  - Skypeln (phone-to-pc), voicemail
- Scale management of user state

The Skype Network

- Ordinary host (OH)
  - A Skype client (SC)
- Super nodes (SN)
  - A Skype client (SC) that has public IP address, 'sufficient' bandwidth, CPU and memory
- Login server
  - Stores Skype id's, passwords, and buddy lists
  - Used at login for authentication
Host Cache

- IP address and port number of online Skype nodes (SNs)
- Maximum size: 200 entries
- Liang, Kumar and Ross. *Understanding KaZaA*
  - 200 entries for ordinary nodes (ON)
- Login server IP address and port number
- HC Windows location (depends on OS version)
  C:\Documents and Settings\All Users\Application Data\Skype

Login

- Public, NAT
  - Establishes a TCP connection with the SN
    - Keep connection alive by sending refresh message every 2 min.
  - Authenticates with the login server
  - Announces arrival on the network (controlled flooding)
  - Determines NAT type
- Firewall
  - Establishes a TCP connection with the SN
  - Authenticates with the login server

Skype Components: Ports

- No default listening port
- Randomly chooses a port (P1) on installation
- Opens TCP and UDP listener sockets at P1
- Opens TCP listener sockets at port 80 (HTTP) and port 443 (HTTPS)

Call Establishment

- Call signaling always carried over TCP and goes e2e
- Public-public call
  - Caller SC establishes a TCP connection with callee SC
- Public-NAT
  - Difficult case is when callee is behind port-restricted NAT
  - Different solutions based on the nature of the NAT
  - Caller ----> Super node ----> Callee
  - TCP connections established between caller, callee, and more than one Skype nodes
- Firewall-firewall call
  - Same as public-NAT but no UDP packets
- Calls to non buddies = search + call
User Search

- From the Skype website
  - Guaranteed to find a user if it exists and logged in in the last 72 hours
- Search is similar to search in KaZzA
  - But were unable to trace messages beyond SN
- User does not exist. How does search terminate?
  - Skype contacts login server for failed searches
- SN searches for a user behind UDP-restricted firewall

Media Transfer

- No silence suppression
- Silence packets are used to
  - Play background noise at the peer
  - Maintain UDP NAT binding
  - Avoid drop in the TCP congestion window
- Putting a call on hold
  - 1 packet/3 seconds to call-peer or Skype node
  - Same reasons as above
- Codec frequency range
  - 50-8,000 Hz (total bw of 3 kilobytes/s)
- Reasonable call quality at (4-5 kilobytes/s)

Summary

- Selfish application
  - Uses best CPU and bandwidth resources
  - Evades blocking
  - Change application priority to ‘High’ after call establishment
- Code obfuscation, runtime decryption
- Login server and super nodes - not strictly peer-to-peer
- STUN and TURN equivalent functionality (handle NATs)
- Combination of hashing and controlled flooding
- Multiple paths for ‘in-time’ switching in case of failures
- Search falls back to login server

Overview

- Web
- Peer-to-peer
- CDN
  - Motivation
  - Edge servers
  - Content delivery
  - Mapping
  - Impact on Internet
- Video
Case Study on Reliability and Scalability: The 2000 Election

Without Akamai this site could not have served customers above their crash zone

Content Distribution Networks (CDNs)

- CDN company installs hundreds of CDN servers throughout Internet
  - Close to users
- When provider updates content, CDN updates servers
- CDNs differ in their "reach"
  - Number of sites, closeness to customers
  - Some content providers have started to deploy their own CDNs

Potential Benefits

- Very good scalability
  - Near infinite if deployed properly
- Good economies at large scales
  - Infrastructure is shared efficiently by customers
  - Statistical multiplexing: hot sites use more resources
- Avoids congestion and long latencies
  - Through mapping to closest server
- Can be extremely reliable
  - Very high degree of redundancy
  - Can mitigate some DoS attacks

Edge Caches

- Region – set of caches managed as a cluster
  - May have a specific function: http, streaming, …
- Availability is a major concern in architecture
- Redundancy at the network level
  - See next slide
- Dealing with server failures
  - Servers do fail occasionally
  - Each server has a “buddy” which is constantly trading hellos
  - If hellos stop, buddy starts to respond directly to requests for primary server
  - Users in the middle of a download may have to hit “reload”
  - No one else will notice any interruption

Example Configuration

Content Delivery:
Possible Bottlenecks

Process Flow

1. User wants to download distributed web content
2. User is directed through Akamai’s dynamic mapping to the “closest” edge cache.

3. Edge cache searches local hard drive for content.

3b. If requested object is not on local hard drive, edge cache checks other edge caches in same region for object.

3b. If requested object is not cached or not fresh, edge cache sends an HTTP GET the origin server.
3c. Origin server delivers object to edge cache over optimized connection

4. Edge server delivers content to end user

1. User requests content and is mapped to optimal edge Akamai server

2. If content is not present in the region, it is requested from most optimal core region
3. Core region makes one request back to origin server

Core Hierarchy Regions

4. Core region can serve many edge regions with one request to origin server

Core Hierarchy Regions

Core Hierarchy Features

Reduces traffic back to origin server
- Reduces infrastructure needs of customer
- Provides best protection against flash crowds
  • Especially important for large files (e.g. Operating System updates or video files)

Improved end-user response time
- Core regions are well connected
- Optimized connection speeds object delivery

Mapping: Server Selection

- Which server?
  - Lowest load → to balance load on servers
  - Best performance → to improve client performance
    • Based on Geography? RTT? Throughput? Load?
  - Any alive node → to provide fault tolerance

- How to direct clients to a particular server?
  - As part of routing → anycast, cluster load balancing
    • Not covered
  - As part of application → HTTP redirect
  - As part of naming → DNS
Application Based

- HTTP supports simple way to indicate that Web page has moved (30X responses)
- Server receives Get request from client
  - Decides which server is best suited for particular client and object
  - Returns HTTP redirect to that server
- Can make informed application specific decision
- May introduce additional overhead → multiple connection setup, name lookups, etc.
- While good solution in general, but...
  - HTTP Redirect has some design flaws

Naming Based

- Client does name lookup for service
- Name server chooses appropriate server address
  - A-record returned is “best” one for the client
- What information can name server base decision on?
  - Server load/location → must be collected
  - Information in the name lookup request
    - Name service client → typically the local name server for client

Mapping Algorithms

- Three main components to finding “closest” edge cache to end user from a Network point of view:
  - Packet Loss + Throughput + Latency
  - Listed in order of importance (roughly)
- Mapping also takes into account edge cache performance
  - Does a server have an object on its hard drive?
  - Uses consistent hashing algorithm
  - Does the edge cache have CPU, RAM, bandwidth, etc. available to serve end-user?

Access to Web Site without CDN
Access Using Akamai

HTTP request
user enters standard URL

Client’s Servers

HTTP request
for embedded content

Akamai Server

HTML code
contains Akamai URLs (ARL)

Content Served Locally

Example ARL: img src="a1000.g.akamai.net/.../www.customer.com/images/logo.gif"

How Akamai Works

cnn.com (content provider)

DNS root server

Akamai server

Get foo.jpg

1 2 3 4

Get index.html

6 7 8

Akamai high-level DNS server

Akamai low-level DNS server

Nearby matching Akamai server

9 10

Get /cnn.com/foo.jpg

Steps in Content Retrieval

• Clients fetch html document from primary server
  – E.g. fetch index.html from cnn.com
• URLs for replicated content are replaced in html
  – E.g. `<img src="http://cnn.com/af/x.gif"> replaced with `<img src="http://a73.g.akamaitech.net/7/23/cnn.com/af/x.gif">`
  – Note that modified name includes original file name
  – Drawback: must modify the content
• Client is forced to resolve aXYZ.g.akamaitech.net hostname
• Initial focus was on static content
  – Has shifted to services
Resolution of Modified Name

- Root server gives NS record for akamai.net
- Akamai.net name server returns NS record for g.akamaitech.net
  - Name server chosen to be in region of client’s name server
  - TTL is large
- G.akamaitech.net nameserver chooses server in region
  - Should try to chose server that has file in cache - How to choose?
  - Uses aXYZ name and hash
  - TTL is small → why?

Benefits

- End user never communicates with origin server – high reliability
  - Thousands of servers backing each other up
  - If one geographic area is disabled, no other area is affected
  - Mitigates some DoS attacks
- Uncacheable content is tunneled back to origin
- Persistent TCP connections increase performance
  - Helps with downloading of objects to end caches

Further Optimizing Performance

Customer CNAME’s (aliases) www.customer.com
  - Anyone looking up www.customer.com will be redirected to an Akamai hostname - "customer.d4p.net"
  - No, I do not know why we use “d4p.net”. 😊
  - customer.d4p.net is CNAME’d to aXXX.g.akamai.net
  - Standard Akamai mapping magic sends returns the closest edge server for aXXX.g.akamai.net

Overview

- Web
- Peer-to-peer
- CDN
  - Motivation
  - Edge servers
  - Content delivery
  - Mapping
  - Impact on Internet
- Video
Change in Traffic with CDN Edge Caches

Impact of CDN Growth

- Flattening of the Internet
  - More content is served from the edge of the network
  - Also seeing an increase in peering
  - More traffic remains at edge – reduced load on core
- Changes in the economic relationships
  - Caches benefit users: better performance, reliability
  - Happy customers benefit the CDN
  - ISP benefits since more content is served locally
    - Reduces traffic from provider – direct economic benefit
  - CDNs sometimes place caches in “eye ball” ISPs for free, but economic models change all the time

Some Recent Trends

- If CDN’s can deploy caches, why can’t I?
- Content providers have started to deploy CDNs
  - Reduce cost, assuming you are large enough
  - Optimize caching to their specific requirements
  - Can still use CDNs, e.g., in certain regions, ….
- Internet Service Providers also try deploy CDNs
  - Sometimes difficult to build the business relationships with content owners – too many ISPs!
  - How about the know-how?
  - Hybrid solutions are emerging, e.g., ISPs install hardware and license software from CDNs

Summary

- Caching improves web performance
- Caching only at client is only partial solution
  - Not enough locality
- Content Delivery Networks move data closer to user, balance load, increase availability, …
  - Is having impact on structure of the Internet
  - No longer just a solution for static content