<table>
<thead>
<tr>
<th>End-system / Application-level</th>
<th># of Receivers</th>
<th># of Senders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalable Data Transfer Applications</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>ftp traditional apps</td>
<td>One</td>
<td>One</td>
</tr>
<tr>
<td>web downloads</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>video conferencing</td>
<td>One</td>
<td>Many</td>
</tr>
<tr>
<td>multiplayer games</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>video on-demand downloads</td>
<td>One</td>
<td>Many</td>
</tr>
<tr>
<td>software distribution</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>chat rooms</td>
<td>One</td>
<td>Many</td>
</tr>
<tr>
<td>video conferencing</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>multiplayer games</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>web downloads</td>
<td>One</td>
<td>Many</td>
</tr>
</tbody>
</table>

More applications:

- server push
- server pull
- client push
- client pull
- chat rooms
- video conferencing
- multiplayer games
- video on-demand
- software distribution
- web downloads
- server push
- server pull
- client push
- client pull
- chat rooms
- video conferencing
- multiplayer games
- video on-demand
- software distribution
- web downloads

Note: The table above represents a platform for building scalable wide-area upload applications.
Scalable Data Transfer Applications

End-system / Application-level chat rooms
video conferencing
multiplayer games
web downloads
software distribution
video-on-demand
server push

One
One
Many
Many
One
Many

Who is Working on Uploads?

To the best of our knowledge, there is no existing work on making many-to-one communication at the application layer scalable and efficient.

Why are there hotspots?

Chat rooms include: Microsoft, Pernot, Yahoo!
Around the world: 150 million users
Use their own network of servers
Relieve web server负担
Many-to-one data transfer

What Are Upload Applications?

Internet-based Computing
Distance Education
Digital Democracy
E-commerce
Internet-based Storage
Digital Government
Data Warehousing
IRS income tax submission
Hard deadlines
Paper submission
Real-life events
Internet-based storage
Data warehousing

Why is Upload Different?

Many-to-one data transfer
Contention for service rather than data
Replication of services and resources for a single event is expensive, inflexible, and not scalable

Traditional Approaches

Increase capacity
Spread the load over time, space, or both
Examples
Data replication
Data replacement
Service replication
Server push
Changing the workload
Examples
FTP mirroring, web caching
Multi-resolution images, video
DNS lookup, NTP
News download, software distribution

Traditional Approaches (Cont...)

Example: Akamai
Relieve web download hotspots through data replication (caching)
Use their own network of servers with strategic placement around the world
Clients include: Microsoft, Paramount, Wired, CBS Sports, NBC, America Online, Apple...

Why are there hotspots?

Real-life events
Availability of new data
More than 2700 servers
More than 45 countries
More than 150 networks

Scalable Data Transfer Applications

# of Receivers

Many

One

# of Senders

Many

One

... multimedia games
... chat rooms
... video conferencing
... video-on-demand
... software distribution
... with downloadable software
... server push
... traditional
... application
... with downloadable application
... one
... many
... one
... many
Our Goals

A single infrastructure (termed **Bistro**) for all data collection needs

- Good performance (for both service providers and users)
- Scalable (shares resources among all service providers)
- Secure (one service provider does not have to trust another)

Current State of Affairs for Uploading

Independent data transfers over the Internet, i.e., TCP/IP

TCP/IP shares bandwidth fairly

Individual clients experience poor performance when the number of clients is large (if transfer time is long enough to see other connections)

TCP/IP is here to stay

Existence of hot spots in uploads is largely due to approaching deadlines

Exacerbated by long transfer times

Problem: too much data ... too little time ...

Submission looks good

We need a commitment of what and when a data was submitted before a specific time

What is actually needed is an assurance that the specific data was submitted before a specific time

Key Observations (cont.)

- we need a commitment of what and when a data was submitted before a specific time
- what is actually needed is an assurance that specific data was submitted before a specific time
- we need a commitment of what and when a data was submitted before a specific time

A solution to uploading with deadlines

Before deadline:

After deadline:

TCP/IP is here to stay

Secure (one service provider does not have to trust another)

Scalable (shares resources among all service providers)

Good performance (for both service providers and users)
Can a fingerprint be forged?
SHA1 is the state-of-the-art electronic fingerprinting algorithm. It generates a 160-bit fingerprint for an any-size document. If you modify a single bit in a document, the new document has a completely different fingerprint. There is no known algorithm that can forge a SHA1 fingerprint while maintaining the integrity of a document.

Can the destination server be under denial-of-service attack?
Yes. That's one weakness of the Internet. However, you can setup mirrors for the destination server by copying the credentials of the destination server onto alternative servers. Nevertheless, in the current Bistro system, this needs to be done ahead of time.

How secure is the encryption? Can it be cracked?
The strength of encryption is usually a function of the algorithm and key size. The Bistro system is not tied to a particular algorithm or key size. It lets the event operator choose these at the time an event is setup. As new and more secure algorithms become available, the system will need to be upgraded to support them.

How big a server do we need in order to give out so many timestamped and certified e-tickets in a short period of time?
To certify an e-ticket requires a digital signature, and signing digital signatures is a time-consuming process. But, as it turns out, digital signatures can be batched. We have developed batch signing schemes (please see our publications) to remove this limitation. Now we can sign as many as it comes.

What about client authentication? Do we know, with certainty, who is submitting a tax return?
As in the current system, you do not know who is submitting a tax return at the time of submission. Even with paper submission, it is very difficult to verify a signature. Client authentication is outside the scope of the Bistro system. If a tax payer uses a tax partner's service to submit his/her tax return, it would be easy to authenticate a tax partner. Each tax partner can independently generate a pair of public and private keys (according to the specifications from IRS) and send the public key to IRS. Each submission can be digitally signed with the tax partner's private key. IRS can verify the digital signature using the corresponding public key.

Opportunities to Speed up Data Transfers

| Host X | 1 Unit of Data |
| Host Y | 1 Unit of Data |
| Host D |

Host Z

1. X sends to D, then Y sends to D -- 2 units of time
2. X sends to D // Y sends to Z then to D -- 1.5 units of time
3. X & Y send simultaneously to D -- 2 units of time
4. X & Y send simultaneously to Z then to D -- 3 units of time
5. ??? -- 1.2 units of time
6. X & Y send simultaneously to Z then to D -- 2 units of time

Advantages of Bistro

- Shares resources and a single infrastructure
- Replaces a traditionally synchronized client push solution with a non-synchronized combination of client-push and server-pull
- Eliminates hot spots by spreading most of the demand on the server over time, by making the actual data transfer independent of the deadline
- Deployable today, i.e., no change required inside the network
- Gradual deployment over a public, private, or mixed infrastructure of hosts
- More dynamic and therefore more adaptive to system and network conditions
- A bistro in every administrative domain, e.g., co-located with web servers or mail servers
- Entire network of bistros collects data for one application/agency one day and for another application/agency the next day
- Use the Bistro infrastructure for other large scale data gathering, transfer, and storage needs

Vision
Some Research Problems

- Resource Location and Discovery
- Placement and Assignment
- Security
- Large Scale Data Transfer

Using digital signature to generate real-time timestamp

- Why digital signature?
  - Integrity
  - Authentication
  - Nonrepudiation

- Online Digital Signatures

- Our Approach

- Different batching schemes
  - No batching scheme
  - Simple batching scheme
  - Extra information to be sent to clients

- High cost of modular arithmetic

- Why digital signature?

HTTP://merlot.usc.edu/cs551-712

Bill Cheng

Bistro Improvements

CS551
Why is this different from downloads?

- Referencing clients to download objects
- Choice of object is exogenous
- Placement or selection (plus assignment) problem
- Assignment problem
- No complete solution

Commit Problem

Bistro

Performance Evaluation

Tree-based schemes have considerable advantages but cost.

Any

Bistro

Destination

External Domain

Commit

Exclusion

Our Approach (Cont.)

Batching schemes reduce a server's CPU load (where hashing is not the dominant factor)
Performance Study

Note: seq. uploads to single server should be approx 3000 sec, and avg. transfer time of one client should be approx 100 sec.

Legend

- Replica
- Pub
- Priv
- Random
- Unrealistic heuristic (approx. lower bound)

Simulation setup (using ns2 & GT-ITM)

- Transit-stub graph with 152 nodes
- 2 transit domains, with avg 4 nodes each, edge capacity of transit-transit edge is 1 Mbit/s
- 2 stub domains, with avg 6 nodes each, edge capacity of transit-stub or stub-stub edge is 256 Kbit/s
- 96 simultaneous uploads with files unif. distr.
- between 100 KBytes & 2 MBytes
- low background load (30%)
- high background load (70%)

Performance results

Performance gains mainly due to parallelism
Large-scale Data Collection

Destination server needs to collect data from all other bistros but how?

One-by-one

All-at-once

Poor resource utilization due to non-shared bottleneck link

Several simple approaches

 Longer transfer time

Spread-in-time-GT

Concurrent-G

Network congestion

Application level re-routing

Avoid congested links

Develop a coordinated transfer schedule

Simulation setup (using ns2 & GT-ITM)

7 other bistros, each with a total amount of data unif. distr. between 25 MBytes & 75 MBytes and the total amount of data in bistros is 350 MBytes.

Opportunities

Concurrent improvement due to re-routing around congestion

Simulation setup (using ns2 & GT-ITM)

No significant effect on throughput of other traffic

Transfer To Destination

Performance improvement due to

PathMerge

PathSync95

PathSync100

PathDelay

Makespan (sec) x 10

0 50 100

# of bg FTPs

0.0 0.5 1.0 1.5

# bg FTPs

Performance improvement due to

PathMerge

PathSync95

PathSync100

PathDelay

Makespan (sec) x 10

0 50 100

# of bg FTPs

0.0 0.5 1.0 1.5

# bg FTPs

Opportunities (Cont...)
Contributions Thus Far

- First effort to study many-to-one communication problem at the application layer & attempt at stating fundamental obstacles
- Proposed a reasonably general framework
- Proposed solutions to all parts of the problem
- Suggested some open problems

Related Work

- Akamai and other content distribution networks
- Napster
- A variety of server selection problems
- Internet security

Dynamic: e.g.,
- [Carter & Crovella 97]
- [Sayal et al. 98]
- [Dykes et al. 00]

Application layer multicast: e.g.,
- [Chu et al. 00]

Client-side server selection
- Statistical: e.g.,
- [Seshnm et al. 97]

Wide area applications
- Wide-area download applications: e.g., Akamai
- [Karger et al. 97]
- Napster type systems, e.g.,
- [Kong & Ghosal 99]
- Gathercast
- [Badrinath & Sudame 98]
- Concast
- [Calvert et al. 00]
- Many-to-one communication at IP level & within Active network framework
- Alternate paths
- [Savage et al. 99]
- Detour
- [Savage et al. 99]
- Application level re-routing
- RON: resilient overlay network
- [Andersen et al. 01]
- Modification on cryptographic algorithm
- [A. Fiat 89]
- One-time signatures used in secret key system
- [Lamport 79, Merkle 88]
- Online batch-based digital signature schemes

Vision

A bistro in every administrative domain
- co-located with web servers
- Entire network of bistros collects data for one application one day & for another application the next day
- E.g., co-located with school data
- A bistro in every administrative domain

Participants Contact Information

Prof. Leana Golubchik
Computer Science Department
University of Southern California
Los Angeles, CA 90089
Email: leana@cs.usc.edu
Voice: (213) 740-4524
Fax: (213) 740-7285
URL: http://cs.usc.edu/~leana

Faculty Members:
- Leana Golubchik
- Samir Khuller (UMD)

Research Staff:
- William C. Cheng
- Yung-Chun Wan (UMD)
- Cheng-Fu Chou (NTU)

Students:
- Chun-Chen Chiu (NCSU)
- Chun-Chen Chiu (UMD)
- Shari Miller (UMD)
- Leana Golubchik
- Research Staff:
- Faculty Members:
- Participants Contact Information

Related Work (Cont.)

- Internet security
- A variety of server selection problems
- Register
- Dynamic: e.g.,
- Application layer multicast: e.g.,
- Client-side server selection
- Wide area applications
- Application level re-routing
- RON: resilient overlay network
- Modification on cryptographic algorithm
- One-time signatures used in secret key system
- Online batch-based digital signature schemes

Related Work (Cont.)

- Internet security
- A variety of server selection problems
- Register
- Dynamic: e.g.,
- Application layer multicast: e.g.,
- Client-side server selection
- Wide area applications
- Application level re-routing
- RON: resilient overlay network
- Modification on cryptographic algorithm
- One-time signatures used in secret key system
- Online batch-based digital signature schemes

Related Work (Cont.)

- Internet security
- A variety of server selection problems
- Register
- Dynamic: e.g.,
- Application layer multicast: e.g.,
- Client-side server selection
- Wide area applications
- Application level re-routing
- RON: resilient overlay network
- Modification on cryptographic algorithm
- One-time signatures used in secret key system
- Online batch-based digital signature schemes

Related Work (Cont.)

- Internet security
- A variety of server selection problems
- Register
- Dynamic: e.g.,
- Application layer multicast: e.g.,
- Client-side server selection
- Wide area applications
- Application level re-routing
- RON: resilient overlay network
- Modification on cryptographic algorithm
- One-time signatures used in secret key system
- Online batch-based digital signature schemes

Related Work (Cont.)

- Internet security
- A variety of server selection problems
- Register
- Dynamic: e.g.,
- Application layer multicast: e.g.,
- Client-side server selection
- Wide area applications
- Application level re-routing
- RON: resilient overlay network
- Modification on cryptographic algorithm
- One-time signatures used in secret key system
- Online batch-based digital signature schemes