CrossTalk: Scalably Interconnecting IM Networks

Marti Motoyama
George Varghese
UC San Diego
Problem

• Growth of social sites relying on “network effect”

• Consequences of the “network effect”:
  – Quality does not necessarily drive adoption
  – Popularity within social circles matters most

Mechanisms to mitigate vendor lock-in for social apps?
Introduction/Motivation

• Case Study: IM Networks
  – Simple social application, provides insight

  • CrossTalk
  – Scalably interconnects IM networks while allowing users to continue utilizing the client of their choice

• Benefits:
  – Allows users to switch to new innovative clients
  – Permits smaller IM networks to interoperate with larger IM services
  – May rekindle interest in 3rd party applications by allowing inter-IM network message exchanges
  • Examples include Chat Translator, Twitter Synch, etc
Outline

• Existing Approaches
• New Idea: Bypass Gateways
• Implementation
• Evaluation
• Future Work:
  – Encryption
  – General Architecture
Approach 1: Ignore Problem

- Cannot communicate across boundaries
Approach 2: Client Consolidation

- **Main problem:** Feature Subtraction
- **Multiple identities**

Reverse engineered protocols, features lost in translation
Client Consolidation Limitations

- Feature subtraction

- Multiple Identities

- Software/Network Overhead

"Never worked for me either"

"..unable to use transfer files using Trillian"

13.7 MB vs 1.3 MB
Approach 3: Standard Gateway

- Main problem: Scalability
- Server can easily detect/disable

Alice and Gateway 1 MUST be buddies

Diagram:
- AIM Server
- Gateway 1
- Yahoo Server
- Gateway 2
- Alice
- Bob
Standard Gateway Limitations

• Gateways subject to user restrictions:
  – Limits on buddy list size (e.g. < 512)
  – Limits on concurrent sessions (e.g. 1 for Skype)
• Suppose IM network with 50 million users:
  – If maximum size of buddy list 500:
    • 100k Gateways to cover all users
• Need **massive** gateway replication to interconnect millions of users
• Thus, standard gateways **do not scale**
• Gateways employ awkward semantics:
  – Ex: IM gtalk:user@domain.com hello
Outline

• Existing Approaches
• New Idea: Bypass Gateways
• Implementation
• Evaluation
• Future Work
Bypass Gateways

Traffic flow through server causes scalability problems

Gateway to Gateway Network

AIM Server

Gateway 1

AIM

Gateway 2

Yahoo Server

Yahoo

Alice

Bob
Bypass Gateways

- Scalable and no feature subtraction (in base network)
- But, *both* ends must be behind gateway
Outline

• Existing Approaches
• Our New Idea: Bypass Gateways
• Implementation
• Evaluation
• Future Work
Crosstalk Components

• Unmodified clients
  – Each user must have the ability to specify a proxy
  – Simple naming convention
    • ex: AOL user *alice* identified as *alice@aol*

• Bypass gateways
  – Interpose between clients and servers

• Gateway-to-Gateway network
  – Logical network
  – Gateways connected via DHT
• **3 major functions:**
  - Translation between AIM, MSN, Yahoo, Jabber
  - Bifurcating Presence Information
  - Merging Buddy Information
Gateway Processing Steps

For each user:

1. Identify protocol, wait for user to authenticate to base server
2. Update user’s state in DHT
3. Retrieve foreign buddy list from DHT
4. Repeatedly: pass-through or intercept & translate
Lessons Learned

• Can use protocol version numbers to allow time for reverse engineering
• Must merge protocol packets carefully
  – Ex: AOL embeds sequence numbers in messages that must be modified to merge information
• Must scale to many TCP connections per user
  – Ex: MSN creates per conversation TCP connection
Applications Built on CrossTalk

• Built 2 example applications:
  – IP geolocation
  – Last.fm information

• Suppose AOL user A wants to know Yahoo user Y’s location or listening habits:
  – User A types “/music” or “/location” as an IM
  – User Y’s client responds with “Shakira” or “San Diego”

• Works because bypass gateways allows IMs to reach other networks
Outline

• Existing Approaches
• New Idea: Bypass Gateways
• Implementation
• Evaluation
• Future Work
Evaluation Questions

• Two metrics

1. Latency: How much delay for translation?
   • Baseline delays vary from 5msec – 230msec

2. Throughput:
   • What size enterprise can a single Bypass Gateway support?

Latency: < 25 msec
Throughput: 1 desktop PC, greater than 4000 person workload
Guo et al monitored MSN, AIM traffic in an enterprise network with > 4000 employees

**Heaviest load** usage characteristics:
- ~130 online users per protocol
- ~20 concurrent conversations per user
- Message length of ~150 characters
- ~1 second interval between successive chat messages
Evaluation Methodology

- Modeled cross/same-to-same IM traffic using market shares
- Implemented mock IM servers to handle same-to-same traffic
- Spread clients/servers across local VMs
- Gateway run on P4 3.2 Ghz, 1Gb
Translation Latency Results

Latency experienced by Yahoo receiver

Max cross traffic latency < 25 msec
Outline

• Existing Approaches
• Our New Idea: Bypass Gateways
• Implementation
• Evaluation
• Future Work
Future Work: Handling Encryption

G2G Network

Skype

Gateway Box

Skype Binary
Gateway Shim
Skype UI
alice
dial: bob

SIP

Gateway Box

SIP Binary
Gateway Shim
SIP UI
bob
Future Work: General Architecture

• Migrate to a **new abstract client layer**
  – Ask for services using abstract calls
    • Send IM, Dial Call, Get File

• Insulate users from changes in technology
  – Protocol intelligence in the cloud

• Make use of bypass gateways or shim layers to achieve interconnectivity for unmodified clients

• Use a DHT to store mappings on behalf of all applications

• Enable **service composition** by modeling services as nodes in graph
Conclusion

- Introduced new technique (bypass gateways) that overcomes vendor lock-in and avoids scaling limits of standard gateways
  - May foster third-party innovation
  - Larger providers cannot easily hinder and smaller providers have incentives to join
  - IM gateway prototype can support throughput for enterprise IM benchmark