Architectural Design of Unified Multiplex Communications for One-Time Use of IP Addresses

Shingo Ata (Osaka City University)
Hiroshi Kitamura (NEC Corporation / Univ. of Electro-Communications)
Masayuki Murata (Osaka University)

Outline
- Research Background
  - What is One-time use IP addresses?
- Unified Multiplex Communication Architecture
  - Overview
    - Two address types (EA and SSA)
    - Technical requirements
- Things to Do to Realize Unified Multiplex?
  - New Technologies for Unified Multiplex Communication
  - Analysis of Interoperability
- Implementation

Motivation
- Machine-to-Machine (Peer-to-Peer, personalized) communication will become more important for future Internet.
- Not easy to deploy servers due to security risks.
- IP address is fixed for a long time so that attacker can have much chance to try cracking.
- Introduce one-time IP address for single communication.
- Design Unified Multiplex Communication Architecture

What are IP addresses used for?
- Currently (from beginning), IP addresses are considered to identify Nodes.
- Connections are identified by using IP addresses + port numbers (+ protocols).
- IP addresses are used (almost) permanently during the node is active.

Problems regarding Security
- Everyone agrees that...
  - Deployment of server w/ global IP address requires to consider many security risks.
  - High-level measures against security are needed.
    - Configuration of firewall, packet filters, loggings, incident reports...
    - Setting up VPN paths, authorization...
    - Careful and frequent maintenance is needed.
  - Is it really possible to all end users? NO!
- Are there really nothing that end users want to deploy servers? NO!
  - Control nodes remotely from outside.
  - Retrieve resources from outside.

Hiding the Existence of Servers from Others
- In case of cell phone: That’s easy!
  - Tell his/her phone number only to trusted people.
- WHY NOT to apply this principle to IP addresses?!
  - Easy to make brute force attack in IPv4 address space.
  - Attack is much faster than phone calling.
  - Hard to regulate attack calls by administrators.
  - However, IPv6 has a HUGE addressing space, i.e.,
    - Hard to brute force attack when IP address is unknown.
  - Shorten the lifetime of IP address as much as possible.
  - Changing IP address session-by-session.

Is there a way for all end users to deploy servers with easy?

NOT easy in IPv4, but possible in IPv6.
One-time IP Addresses

- IP addresses are used to identify nodes, BUT for sessions!
- Assigned just before the communication starts, disposed after the communication ends.

Security Advantages when using One-time Addresses

- IP address is ONLY valid for an associated session.
- Prevention against reverse tracking/attacking.
  - Though the IP address is known by someone, it is no longer used for any communication.
- Prevention against inference of node behavior.
  - Difficult to merge sessions having the same IP address
- Advanced authorization/authentication by using IP address only.
  - Embedding some magic numbers in IP address

Simple way to enhance security without any firewalls!

Unified Multiplex Communication Architecture

<table>
<thead>
<tr>
<th>1 Node-1 Fixed Address</th>
<th>1 Node - Multi-Floating Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Used Addresses</td>
<td>(Current) Legacy</td>
</tr>
<tr>
<td>Information Dealing</td>
<td>Use Only One Address (Basically)</td>
</tr>
<tr>
<td>Service (on Servers)</td>
<td>General and Share Use Same Address</td>
</tr>
<tr>
<td>Information Fluidity</td>
<td>Wait for Anytime (24hour / 365days)</td>
</tr>
<tr>
<td></td>
<td>Fixed (Not Changed)</td>
</tr>
</tbody>
</table>

IP Addresses in Legacy Communication

- Nodes have (typically) a single IP address
- Connections are distinguished by port numbers

IP Addresses in Unified Multiplex Communication

- Nodes have multiple (many!) IP addresses
- Different IP addresses are used for different connections
Two address types on Unified Multiplex

- **EA (Ephemeral Address)** for Clients
  - Like ephemeral ports in current TCP stack.
  - Unless specifying the port number explicitly, the client automatically uses an available port number.
  - Before establishing connection, the client assigns a new IP address, which is not used for any other connections.

- **SSA (Specific Service Address)** for Servers
  - Only valid for a single session.
  - Before the communication, an SSA is generated for a client to communicate with the server and notified to the client.

Requirements for Unified Multiplex Communication

- Easy to deploy
  - Enable unified communication by updating OS software in end nodes only (no router replacements needed).
  - Support application without any updates (source modification, re-compiling).
- Co-existence with Legacy Communication nodes
  - Support gradual migration to Unified Multiplex Communication.

Comparisons of TCP communication between Unified and Legacy

- Setting up sockets
  - Timings of binding IP address to socket are completely different.
  - IP addresses for sockets are not determined at bind() call.
- Connection identification in kernel
  - Legacy: IP addresses, port numbers, protocol.
  - Unified: IP addresses ONLY.
- Address lifetime
  - Legacy: Used permanently.
  - Unified: Invalid after the connection ends.

Procedure for Establishing TCP Connection (Legacy)

- Pre-determined address+port number are used for binding sockets.

Procedure for Establishing TCP Connection (Unified)

- Port number is abolished.
- Addresses for sockets are not determined at bind(), but assigned just before accept().

Functionality needed for Unified Multiplex Communication

- Delayed address setting (DAS) at connection establishment
  - Set an IP address for the socket after bind() is called.
  - Introduce new (Uncertain) state of IP address.
  - Automatic address setting for supporting applications without modification.
  - Address generation methods.
  - Introduce new TCP state for sockets without assigned IP address.
- Ignoring port numbers
  - Port number is no longer used for connection identification in PCB (Protocol Control Blocks) of kernel.
- Releasing addresses at the end of session.
Two choices for address setting:
- There is no procedure to assign EA/SSA in current applications.
- New state to represent:

## Auto Address Settings

- There is no procedure to assign EA/SSA in current applications.
- Automatic assignment mechanism is needed for applications without any modifications.
- Two choices for address setting:
  - Automatic (Auto Set): Suitable for most applications.

### Auto Address Settings

<table>
<thead>
<tr>
<th>Mode Level</th>
<th>Use of port number for distinguishing packets</th>
<th>Generation of port number available</th>
<th>Available Communication Style</th>
<th>Port Ignoring Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port ignore 0</td>
<td>No</td>
<td>No</td>
<td>Any</td>
<td>USSA, ’<em>’ — UEA, ’</em>’</td>
<td>Port ignore 0</td>
</tr>
<tr>
<td>Port ignore 1</td>
<td>No</td>
<td>No</td>
<td>Source port number of received packet</td>
<td>USSA, ’<em>’ — UEA, ’</em>’ USSA, ’*’ — LUEA, EP</td>
<td>Port ignore 1</td>
</tr>
<tr>
<td>Port ignore 2</td>
<td>No</td>
<td>Yes</td>
<td>Source port number of received packet</td>
<td>USSA, ’<em>’ — UEA, ’</em>’ UUEA, ’*’ — LUEA, EP</td>
<td>Port ignore 2</td>
</tr>
<tr>
<td>Port aware</td>
<td>Yes</td>
<td>Yes</td>
<td>Source port number of received packet</td>
<td>LUEA, EP — LUEA, EP</td>
<td>Port aware</td>
</tr>
</tbody>
</table>

## Interoperability

- Four levels of treating port numbers for communication with Legacy nodes.
  - Port Ignore Mode Level 0
    - Completely ignore port numbers in PCB.
    - Port number fields in packet header are not used.
  - Port Ignore Mode Level 1
    - Completely ignore port numbers in PCB.
    - Set port number of source node for return packet.
  - Port Ignore Mode Level 2
    - Use port numbers in PCB of client.
    - Set port number of source node for return packet.
  - Port Aware Mode
    - Same as Legacy mode.
    - Set fixed (LEGACY_COMPAT) port number for compatibility.

## Implementation and Verification Status

- Unified Multiplex Communication Architecture functions have been implemented on the following:
  - FreeBSD 6.2.R FreeBSD 6.8.R
  - Linux kernel 2.6.24 (Implemented functions are limited)
- Without modifications of communication applications.
- Only with the kernel replacement.
- It has verified that basic functions work correctly as they are designed.
Conclusion

- One-time use of IP address is promising for reducing security risks of nodes easily and its benefit is applicable to all end users (non-experts).

- Unified Multiplex Communication Architecture
  - Realizes one-time IP address communication style with ease.
  - Has interoperability with Legacy nodes.

- Design and implementation have done in FreeBSD and Linux (partially).

- Further refinement and validation of applications are needed for wide deployment.