Towards an Evolvable Internet Architecture

Thomas Lohmüller
lthomas@student.ethz.ch

December 19, 2007
Towards an Evolvable Internet Architecture

Topics:
- How to evolve from IPv(N-1) to IPvN
- How to use overlay networks in legacy applications

Goals:
- Show some nice ideas for evolvability
- Describe needed technologies
- Show *creative* way how to use legacy applications with new network technologies
Outline

1. Evolvability
2. vN-Bone
3. Legacy Apps and Overlays
4. Summary
Today everyone wants to change the Internet.

- Why is it so hard to change?
- Why did the different approaches not work?
- How can we make the Internet changeable?
Once upon a Time...
In the early days of commercial Internet (mid 1990’s)

- Great faith in Internet evolution.
- Many believed ISPs would soon deploy new versions of IP

But it came different...
Today

- Success of the Internet surpassed our wildest imagination
- Deep pessimism about evolutionary architectural change
- ISPs have little incentive to deploy new architectures
- Most operating systems do not support new protocols
- Costs of universally deploying a new architecture are immense
Requirements

What is required to make the Internet evolvable?

- Foster independent innovation
- Enable customer choice
- Allow ISPs some degree of control
Basic Assumptions 1/2

A1: Assume partial ISP deployment.

- Not all ISP’s will deploy a new version of IP at the same time.
- Must work with only a subset of an ISP’s routers implementing it.

A2: Assume partial ISP participation.

Not all ISP’s are willing to participate.
**Basic Assumptions 2/2**

**A3:** Assume the existing market structure and agreements. Clients should not need a contract with an additional provider.

**A4:** Assume revenue flow. Assume that IPvN attracts users.

**Require Universal Access**
All clients can use IPvN if they choose regardless of whether their ISP deploys IPvN or not.
Open Problems

Problems on our way to IPvN:

**Client-Side**
1. Locate IPvN router
2. Move packets to IPvN router
3. Get IPvN address

**Network**
1. Topology construction
2. Addressing
3. Routing

Towards an Evolvable Internet Architecture
Option 1: Application-Level Redirection
Why this is not a good choice.

- Application queries a lookup-service for IPvN router
- Application has to tunnel packets to IPvN router

Problems:
- Who runs this lookup service?
  - Current ISPs
  - Third-party broker
- How to reach lookup-service?
- Who pays for this service?
Option II: Network-Level Redirection
The better solution.

- Every router in network (whether IPvN or not) *knows* how to forward an IPvN packet to an IPvN router
- Works with current market structure
- How can a client in a non-offering ISP be guaranteed access?

Do some "magic" here ...

![Diagram](image)
IP Anycast

- Host transmits to an anycast address
- Network is responsible for delivery to one of possibly multiple servers
- Described in *RFC 1546*
- Today in use for root DNS name servers
- Enables seamless spread of deployment
- Does not require any change in current routing infrastructure
Today, this system does work.

```bash
$ traceroute 192.88.99.1
traceroute to 192.88.99.1, 30 hops max, 40 byte packets
 1 rou-rz-1-service-inf-isg-rz.ethz.ch (129.132.216.33)
 2 rou-ref-hg-service-inf.ethz.ch (10.1.18.38)
 3 rou-fw-cla-service-inf-isg.ethz.ch (10.1.18.34)
 4 rou-fw-rz-fw-cla.ethz.ch (192.33.92.185)
 5 rou-rz-gw-fwrz-gwrz-core.ethz.ch (192.33.92.170)
 6 swiez2.ethz.ch (192.33.92.11)
 7 swiLS2-10GE-1-1.switch.ch (130.59.36.205)
 8 swiEL2-10GE-1-2.switch.ch (130.59.36.70)
 9 swiCE3-10GE-1-3.switch.ch (130.59.37.65)
10 swi6netCE1-G0-0.switch.ch (130.59.35.137)
```
Anycast Routing I
Intra-Domain-Routing

- IPvN router advertises link to anycast address
- Link-state routing (OSPF)
  - High-cost link to prevent routing through anycast address
  - IPvN router can easily identify other IPvN routers
- Distance-vector routing (RIP)
  - Zero-cost link
  - Distance-vector routing routes packets to closest IPvN router
  - Requires additional discovery-mechanism for IPvN routers to see each other
**Anycast Routing II**

**Inter-Domain-Routing**

---

**Option 1: Use anycast address**
- Non-aggregatable addrs.
- Global routes
- Propagate route in BGP
- Requires change in policy
- One additional route per anycast address

**Option 2: Use unicast address**
- Aggregatable addresses
- Default routes
- BGP already knows IP
- No change in policy
- Use an IP of first IPvN provider
- Additional traffic for first provider
IP Anycast Example

Default domain for $A_n$

Support IPvN

Thomas Lohmüller
Towards an Evolvable Internet Architecture
Open Problems

Problems on our way to IPvN:

**Client-Side**
1. Locate IPvN router
2. Move packets to IPvN router
3. Get IPvN address

**Network**
1. Topology construction
2. Addressing
3. Routing
IP in IP-Tunnel

- Very easy...
- Described in RFC 1853 from 1995
- Supported by most routers even today
Open Problems

Problems on our way to IPvN:

**Client-Side**
1. Locate IPvN router
2. Move packets to IPvN router
3. Get IPvN address

**Network**
1. Topology construction
2. Addressing
3. Routing
1. Evolvability
2. vN-Bone
3. Legacy Apps and Overlays
4. Summary

- Topology Construction
- Addressing
- Routing
- Deploying Source-Specific Multicast
Open Problems

Problems on our way to IPvN:

Client-Side
1. Locate IPvN router
2. Move packets to IPvN router
3. Get IPvN address

Network
1. Topology construction
2. Addressing
3. Routing
Intra-Domain Topology Construction

- IPv(N-1) routing protocols
- Every router has complete knowledge
- Easy to locate IPvN routers
Inter-Domain Topology Construction

- Inter-domain tunnels based on peering policies
- Use Anycast to bootstrap
- Prevention of partitions important
  - Check for connection to *default* provider
Open Problems

Problems on our way to IPvN:

<table>
<thead>
<tr>
<th>Client-Side</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Locate IPvN router</td>
<td>1. Topology construction</td>
</tr>
<tr>
<td>2. Move packets to IPvN router</td>
<td>2. Addressing</td>
</tr>
</tbody>
</table>
Addressing

- **Addressing**
  - Format or structure of address
  - Address allocation
  - Advertising into routing fabric

- **Today’s Internet (IPv4 and IPv6)**
  - Address allocation and advertising by local access provider

- **How to get IPvN-address if access-provider does not support IPvN?**
Possible solutions:
- Request address from IPvN router (like DHCP)
- Self-addressing by hosts

Self-addressing in IPv6
- Well-known suffix 2002: and embedded IPv4 address
  IPv4: 100.200.100.200
  IPv6: 2002:64C8:64C8::

How to advertise and route such addresses?
Open Problems

Problems on our way to IPvN:

**Client-Side**
1. Locate IPvN router
2. Move packets to IPvN router
3. Get IPvN address

**Network**
1. Topology construction
2. Addressing
3. Routing

Thomas Lohmüller  
Towards an Evolvable Internet Architecture
Routing

- Between routers
  - Native IPvN routing. No problem here...

- Between endhosts
  - Ingress router easily to reach (IP Anycast)
  - Egress also easy if destination is in IPvN enabled domain
  - What if destination is not in IPvN enabled domain?
Routing to non-IPvN Domains

- Let router advertise temporary IPvN address
  - Let client register itself at ingress router
  - Simple but departure from existing norms
  - Many non-aggregatable routes
- Use IPv(N-1) routing
  - IPv(N-1) address contained in temporary IPvN address
  - Leave IPvN network and use IPv(N-1) routing
  - Simple, but fails to fully exploit IPvN deployment
- Let IPvN routers advertise on-behalf-of IPv(N-1) domains
  - IPvN router advertises some IPv(N-1) prefixes
  - Packet leaves vN-Bone near destination.
Routing Examples: IPv(N-1) Routing

- Client
- ISP-N
- IPv(N-1) links
- IPv(N-1) router
- IPvN router
- IPvN links
- ISP-O
- ISP-M
- Server IPvN

Tom Lohmüller Towards an Evolvable Internet Architecture
Routing Examples: *on-behalf-of* Routing

- **Client**
- **ISP-O**
- **ISP-M**
- **ISP-N**
- **Server**
- **IPv(N-1) links**
- **IPv(N-1) router**
- **IPvN router**
- **IPvN links**
Source-Specific Multicast

- Deploy Source Specific Multicast (SSM)
- Provides one-to-many packet delivery
- Multicast group defined by (S,G)
  - S: Unicast address of source
  - G: Multicast group address
- Simple interface for clients
  - subscribe(S,G)
  - unsubscribe(S,G)
How it Works

1. Use IP Anycast to locate IPvM router
2. Send $\text{subscribe}(S,G)$ to ingress router
   - Encapsulated in IPv4 packet if needed
3. Router adds entry to multicast forwarding table
4. Router sends $\text{subscribe}(S,G)$ to next router if required
SSM Example
Introduction

- Overlays provide new features without changing the Internet
  - Resilient Overlay Network (RON)
  - Internet Indirection Infrastructure (i3)
  - ...
- No widespread deployment
  - Users unwilling to shift to new application programs
  - No interoperability between different overlays
OCALA

New Internet Architectures

1. Build
2. Benefit
3. Feedback

Networking Researchers

Users with existing applications

Source: http://ocala.cs.berkeley.edu/
Goals of OCALA

- Enable legacy applications to work over overlays
  - All applications which use IP
  - No changes at application needed
  - Users choose the best overlay for a particular application
- Enable hosts in different networks to talk to each other
  - Interoperability between hosts in different overlays
  - Interoperability between overlay hosts and pure IP hosts
- Factoring out common functionality
  - Concentrate on architecture; not on supporting legacy applications
  - Factor out common functionality
Add new layer below Transport Layer

- Applications
  - (ssh, Firefox, ...)
- Transport Layer
  - (TCP, UDP, ...)
- Overlay Convergence Layer (OC)
- Overlay
  - (i3, RON, DOA, ...)
- OC Independent (OC-I) Sublayer
- OC Dependent (OC-D) Sublayer
Expressing which Overlay to Use

- DNS-like names to identify machines (or services)
  - Supported by most legacy applications (but not all!)
  - Needs at least a configuration change in application
- Append a new part after top-level-domain overlayspecific.overlayname
  - ucb.i3: connect to host ucb over i3
  - ucb interpreted by overlay-specific OC-D
  - i3 overlay to use
Connection Setup in OCALA

1. $\text{DNSreq}(\text{foo.ov})$
2. $\text{setup}(\text{foo.ov})$
3. $\text{resolve}(\text{foo.ov})$
4. $\text{ID}_B$
5. overlay specific setup
6. $\text{tunnel} = \text{td}_{AB}$
7. $\text{OCIsetup}(\text{pd}_{AB})$
8. $\text{DNSresp}(\text{handle} = \text{IP}_{AB})$
Implementation

- As user-level proxy
- Uses *tun* device to capture packets
- Implemented for Mac OS X, Linux and Windows XP
- about 40k SLOC of C++
- OC-D modules
  - Dynamic loadable libraries
  - Simple 5 function call interface
  - i3 and RON-modules written internally
  - Many more modules written by others
- GUI for configuring OCALA
Requirements to Evolve to IPvN

- **IPvN Hosts**
  - Must be able to create temporary addresses
  - Has to contain IPv(N-1) address

- **IPvN routers**
  - Should be able to advertise IPv(N-1) routing information
  - Have to participate in IPv(N-1) unicast and anycast
  - Perform IPv(N-1) forwarding
  - Participate in vN-Bone construction
  - Perform IPvN forwarding *(of course...)*
Current State of IPv6

- Routing to ingress router by IP Anycast
  - Operational at 192.88.99.1, RFC 3068
  - Transparently used by some home-routers
- Temporary IPv6 address containing IPv4 address
  - Standardized in RFC 3056
  - Implemented in current operating systems
- v6-Bone
  - Operational. See http://go6.net/ipv6-6bone/
Towards an Evolvable Internet Architecture
Evolvable Internet: My Opinion

- Good introduction to IPv6 deployment
- Summary of many other papers (60 papers referenced!)
- All ideas pretty obvious. No surprises...
OCALA: Summary

- User-level proxy: Simple to deploy
- Ready-to-use
  - Simplify implementation of new overlay-module
  - Real users, real applications
- Does not work with packet rewriting

Applications
(ssh, Firefox, ...)

Transport Layer
(TCP, UDP, ...)

Overlay Convergence
Layer (OC)

Overlay
(i3, RON, DOA, ...)

OC Independent
(OC-I) Sublayer

OC Dependent
(OC-D) Sublayer
OCALA: My Opinion

- *Just add another layer*
- Helps to simplify implementation of new overlays
- Is’s a hack
  - Misuse of DNS hostname
  - Bad implementation (uses `tun` device)
  - *Library preloading* with external configuration would be a much cleaner solution
- Only useable in testing environments
Discussion

Topics:
- Evolution to IPvN
  - IP Anycast
  - Addressing
  - Routing in vN-Bone
  - IPvM (example)
- Evolution to IPv6
- OCALA
Further Reading

Towards an evolvable internet architecture
S. Ratnasamy, S. Shenker S. McCanne

OCALA: An Architecture for Supporting Legacy Applications over Overlays.
Dilip Joseph, Jayanthkumar Kannan, Ayumu Kubota, Karthik Lakshminarayanan, Ion Stoica, Klaus Wehrle