OCALA
Overlay Convergence Architecture for supporting Legacy Applications on Overlays

Dilip Antony Joseph¹, Jayanth Kannan¹, Ayumu Kubota², Karthik Lakshminarayan¹, Ion Stoica¹, Klaus Wehrle³

¹UC Berkeley, ²KDDI Labs, ³University of Tübingen

Motivation

• Many attempts to improve the Internet:
  – i3: mobility, NAT traversal, anycast, multicast
  – DOA: middlebox support
  – OverQoS: quality of service
  – SIFF: resilience against DDoS attacks
• But still no widespread deployment…

• Problem: rewriting/porting popular applications for new architectures a daunting task!
**Goal**

- Support legacy applications (e.g. ssh, Firefox, IE) over new network architectures and overlays
  - Enable users to take advantage of new network functionality using their favorite applications!

**Solution**: Overlay Convergence Architecture for Legacy Applications (OCALA)

Interpose an Overlay Convergence Layer between transport layer and overlay networks

- Legacy Applications (ssh, firefox, explorer, ...)
- Transport Layer (TCP, UDP, ...)
- Overlay Convergence (OC) Layer
- Overlay (DOA, DTN, HIP, i3, RON, ...)

- OC Independent (OC-I) Sublayer
- OC Dependent (OC-D) Sublayer
Simultaneous access to multiple overlays

Which overlay to use?

- IP address and port number:
  - Eg: Forward all packets sent to 128.32.132.223 port 22 over RON

- DNS name:
  - Eg: Forward all packets sent to berkeley.edu.ron over RON
  - Eg: Forward all packets sent to berkeley.edu.i3 over i3
Bridging Multiple Architectures

- Communication across overlays
- Stitch together functionality

Legacy Client Gateways – Demo

- Clients need not run OCALA locally
- Gateway has special Legacy Client IP (LCIP) module
Legacy Server Gateways

- Server need not run OCALA locally
- Special OC-D module called Legacy Server IP (LSIP) at gateway
- LSIP behaves like a software NAT box

Legacy Client Gateways – Demo

- Can access following links:
Overlay Convergence Architecture for Legacy Applications (OCALA)

Interpose an Overlay Convergence Layer between transport layer and overlay networks

- Legacy Applications (ssh, firefox, explorer, …)
- Transport Layer (TCP, UDP, …)
- Overlay Convergence (OC) Layer
- OC Independent (OC-I) Sublayer
- OC Dependent (OC-D) Sublayer
- Overlay (DOA, DTN, HIP, i3, RON, …)

Setting up a new connection

1. DNSreq(foo.ov)
2. setup(foo.ov)
3. resolve(foo.ov)
4. ID_B
5. Overlay specific setup protocol
6. tunnel_d = td_AB
7. OCI-Setup (pd_AB)
8. DNSresp(oc_handle = IP_AB)

Host A

Legacy App.  Transport Layer

Host B (foo.ov, ID_B)

Overlay (DTN, i3, RON)

1.x.x.x
**Data Flow**

**Host A (ID_A)**

- Legacy App.
  - Transport Layer

  - IP_{AB} data
  - "foo.ov" → pd_{AB}
  - pd_{AB} ↔ IP_{AB}
  - pd_{AB} → td_{AB}

- OC-I

- OC-D

  - ID_B pd_{AB} IP_{AB} data
  - td_{AB} → ID_B

**Host B (foo.ov, ID_B)**

- Legacy App.
  - Transport Layer

  - IP_{BA} data
  - pd_{BA} ↔ IP_{BA}
  - pd_{BA} → td_{BA}

- OC-I

- OC-D

  - ID_B pd_{AB} IP_{AB} data
  - td_{BA} → ID_A

---

**Implementation**

- Implemented as a proxy to be run by the user.
  - *tun* device used to capture packets
- Works on Linux and Windows XP/2000
  - Mac almost done…
- OC-D modules
  - Dynamically loadable libraries.
  - Implemented RON, i3, DOA, HIP OC-D modules.
    - 250 lines of glue code in case of RON.
    - HIP/DOA OC-D modules implemented by HIP/DOA researchers
- Configuration GUI
Common functionality

• Functionality required by multiple overlays implemented in the OC-I layer

• Example: Security
  – Similar to SSL
  – Modifications for supporting middleboxes

Conclusion

• Enables unmodified legacy applications to simultaneously access multiple overlays

• Stitch together functionality of different overlays

• Helps network researchers bring functionality of new network architectures to real users