OpenBox – Enabling Innovation in Middlebox Applications

Yotam Harchol
Hebrew University of Jerusalem, Israel

Joint work with:
Anat Bremler-Barr (Interdisciplinary Center Herzliya, Israel)
David Hay (Hebrew University of Jerusalem, Israel)

ACM SIGCOMM HotMiddleboxes 2015

This research was supported by the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement no 259085.
Software-Defined Solutions

Forwarding plane (switches, routers):
- High cost
- Limited management
- No multi-tenancy
- Limited functionality and limited innovation
- Complex distributed algorithms

Solution: SDN / OpenFlow

Middleboxes:
- Higher cost
- Limited and separate management
- Limited provisioning and scalability
- No multi-tenancy
- Limited functionality and limited innovation
- Similar processing steps, no re-use

Our solution: OpenBox
**OpenBox**

- **OpenBox: A new protocol**
- Decouples middleboxes control from their data plane
- Unifies data plane of multiple middleboxes

**Benefits:**
- Easier, unified control
- Better performance
- Scalable
- Flexible
- Multi-tenancy
- Innovation
A Different View of Middleboxes

• **Previous works:** Middlebox = monolithic closed unit
  – Middlebox Traffic Steering (e.g., SIMPLE [Sigcomm ‘13], Stratos)
  – Middlebox Virtualization (e.g., ComB [NSDI ’12])
  – Middlebox State Management (e.g., OpenNF [Sigcomm ‘14])
  – Middlebox Runtime Platform (e.g., xOMB [ANCS ‘12], SDM [INFOCOM ‘14])

• **OpenBox:** Middlebox = logical application
  – Most processing steps are shared among many types of middlebox applications
  – Some steps can be done once for multiple applications
Unified Processing in Data Plane

- **L7 Load Balancer**
  - Header Lookup
  - Session Analysis
  - Payload Lookup
  - Modify Header

- **Firewall**
  - Header Lookup
  - Alert / Log

- **Intrusion Prevention**
  - Header Lookup
  - Session Analysis
  - Protocol Analysis
  - Decompression
  - Payload Lookup
  - Alert / Log

- **OpenBox Service Instance**
  - Virtual or Physical
• Each application defines:
  – Processing path (of processing stages)
  – Rules:
    <priority, header match, metadata match, payload match, actions>

• Northbound API basic building blocks:
  – Specify processing path and rules
  – Customize specific processing stages
  – Metadata: information passing between processing stages
  – Session based storage
    • E.g., gzip window for decompression
  – Content storage
    • E.g., for caching and quarantine
(Logically-)Centralized Control

- **Aggregates** multiple OpenBox applications
  - Creates a *Processing Graph* that aggregates multiple processing paths
  - Aggregates *rules* based on *priority* and network location

- Provides a central *management*
  - *Multiple tenants* run *multiple applications* for *multiple policies* in *the same network*
OpenBox Service Instance (OBI)

- Receives processing graph and rules from controller
  - executes them on data path

- Hardware or software based
  - Room for innovation
  - Proactive rule spawning

- Centralized
  or

- Distributed
  - E.g., performs all or just part of the processing stages
  - Results of each stage are passed as metadata for next stages
    - Use NSH / Geneve / VXLAN-GPE (IETF drafts) with jumbo frames for metadata passing

Can re-use:
- SDN switches for header lookup
- Programmable packet processors (P4)
- Click / DPDK platforms
Time for most processing stages is sublinear with # of rules (or even constant)

OpenBox Service Instance
2000 Rules
\( \log(2000) \) latency
\(< 2 \times \log(1000)\)
Also improves throughput when re-using resources

Was shown for DPI in [Bremler-Barr et al., CoNEXT ‘14]
Innovation in Data Plane

- OpenBox provides a set of **required** processing steps
- Vendors can provide **enhanced OBI**s along with their OpenBox applications:
  - Can place these enhanced processors only where needed
  - No need the whole network to comply with a specific vendor
  - Vendors may cooperate
Scalable & Reliable Data Plane

• Easy provisioning of new OBIs by simply running more VMs (VNFs)
  – Even for hardware-assisted tasks, can use software VMs as a backup for peak times
• OBIs can report load information to controller, to allow centralized control over provisioning
• Increases reliability – quickly respond to crashes
Conclusions

• Middleboxes are currently a real challenge in datacenter, operator, and enterprise networks

• OpenBox decouples the data plane processing from middlebox applications logic and:
  – Reduces costs of management
  – Enhances performance
  – Improves scalability
  – Increases reliability
  – Provides multi-tenancy
  – Allows easier innovation and lets new players into the game
Current Status:
• Ongoing development of control plane and data plane
  – Along with some sample applications such as NIPS, Load Balancer, NAT
  – Looking at Click/DPDK as a possible platform
• Working on defining the OpenBox protocol

Future Work:
• Algorithms for smart aggregation of OpenBox applications
• Algorithms for smart placement and provisioning of OpenBox service instances
• Distributed data plane, possibly cooperating with SDN/P4 switches
• New OpenBox applications
Questions?

Thank you.