MAKING MIDDLEBOXES SOMEONE ELSE’S PROBLEM:
NETWORK PROCESSING AS A CLOUD SERVICE

J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, V. Sekar
Presented by: Angela Jiang
Middleboxes

- Improve security
  - Firewalls, intrusion detection systems
- Improve performance
  - Proxies, caches
- Reduce bandwidth costs
  - WAN optimizers
Drawbacks of Middleboxes

- Expensive infrastructure
- Complex to manage
  - Specialized processing
  - Heterogeneous
- Creates new failure modes
Challenge of outsourcing middleboxes

No longer on path or local

- Functionality may be topology dependent
  - E.g. caches, WAN optimization
- Bandwidth consumption
- Increased packet latency
On-path vs. in the cloud

- What are challenges of middleboxes?
  - Surveyed 57 enterprise administrators

- How can they be addressed by the cloud?
Large deployments

☐ Number of middleboxes close to that of routers
Middleboxes today are located on-path middleboxes, while off-path middleboxes for routing control plane functions. The advent of cloud computing offers new, perhaps better, options for supporting middlebox functionality.

We find that all options have natural tradeoffs across the above requirements and settle on a design that we argue is the sweet spot: the redirection options available to enterprises, the footprint of the cloud provider, and the complexity of the outsourcing mechanism. Paralleling arguments for cloud computing, outsourcing middlebox processing can reduce hardware costs: outsourcing eliminates over $50,000.

Before discussing outsourcing designs, we draw on two datasets: NANOG network operators group, and thus does not include many homes and very small businesses. Our data was primarily surveyed from the research community. Our dataset includes 19 small networks with only tens of hosts, 11 large (10k-100k hosts) networks, and 7 very large (more than 100k hosts) networks.

In §2 we explore the design space for outsourcing middleboxes. We implement APLOMB and evaluate, APLOMB imposes an average latency increase of only 1 ms and a median bandwidth inflation of 3.8%.

A systematic exploration of the requirements and design space in §6 and related work in §7 before concluding in §8.

Outsourcing Middleboxes. We implement APLOMB and evaluate our system on EC2 using real end-user traffic and an analysis of traffic traces from a large enterprise network. In our enterprise deployment of a large enterprise, traffic is instead sent on a detour through the proposed architecture, traffic is instead sent on a detour through the APLOMB architecture.

Figure 3: Administrator-estimated number of personnel per network.

Number of Middleboxes

- $1M-50M
- $500K-1M
- $50K-500K
- $5K-50K
- <$5K

5 Year Expenditure

- 2-5
- 10
- 100
- 1000
- 10000

Number of Middleboxes

- 1
- 10
- 100
- 1000
- 10000

Hardware costs

- $1M-50M
- $500K-1M
- $50K-500K
- $5K-50K
- <$5K

5 Year Expenditure

- 2-5
- 10
- 100
- 1000
- 10000

Number of Middleboxes

- 1
- 10
- 100
- 1000
- 10000
Before discussing outsourcing designs, we draw on two datasets: (1) a study of costs and concerns in 57 real-world middlebox deployments, and (2) a case study of how our system would impact the middlebox ecosystem of firewalls, IDSes, web proxies, and other devices.

The design, implementation, and evaluation of the APLOMB architecture.

A systematic exploration of the requirements and design space for outsourcing middleboxes.

To summarize, our key contributions are:

1. Outsourcing Middleboxes. We implement APLOMB and evaluate its performance.
2. Low complexity at the enterprise.
3. Low performance overhead.
4. Promise of substantial cost savings, including hardware and maintenance.

We find that all options have natural tradeoffs across the above requirements and settle on a design that we argue is the sweet spot for outsourcing in the first place.

The redirection options available to enterprises, the footprint of the cloud leading to a potential increase in packet latency and bandwidth consumption. We aim for system designs that minimize this performance penalty.

Paralleling arguments for cloud computing, outsourcing middleboxes is a promising solution for enterprises to address the administrative and management requirements (§2.2), and the need for overprovisioning to guarantee QoS (§2.3). We argue these challenges faced in administering them. Our dataset includes 19 small networks with only tens of hosts, 11 large (10k-100k hosts) networks, and 7 very large (more than 100k hosts) networks. We conducted a survey of 57 enterprise network administrators, including NANOG network operators group, and thus does not include many enterprises with only tens of international sites; we elaborate on this dataset in §5.3.

Our analysis highlights several key challenges that enterprise administrators face with middlebox deployments: large deployments are expensive, requiring expertise, and a substantial size. Our data was primarily surveyed from the most of the infrastructure at the enterprise, and a cloud provider can most of the time provide better services for $5,000-50,000 dollars, and the top third of the small networks spent between $50K-500K on middlebox hardware per network.

Of the very smallest networks (e.g., homes and very small businesses), our system was observed to increase the number of personnel per network (1). Low performance overhead.

Figure 3: Administrator-estimated number of personnel per network.

Figure 2: Administrator-estimated spending on middlebox hardware per network.

Figure 1 shows a box plot of the number of middleboxes deployed in networks of all sizes, as well as the number of routers deployed with them. Across all network sizes, the number of middleboxes increases as the number of routers increases, as does the number of devices. Our analysis highlights several key challenges that enterprise administrators face with middlebox deployments: large deployments are expensive, requiring expertise, and a substantial size. Our data was primarily surveyed from the most of the infrastructure at the enterprise, and a cloud provider can most of the time provide better services for $5,000-50,000 dollars, and the top third of the small networks spent between $50K-500K on middlebox hardware per network.

Of the very smallest networks (e.g., homes and very small businesses), our system was observed to increase the number of personnel per network (1). Low performance overhead.

Figure 3: Administrator-estimated number of personnel per network.

Figure 2: Administrator-estimated spending on middlebox hardware per network.

Figure 1 shows a box plot of the number of middleboxes deployed in networks of all sizes, as well as the number of routers deployed with them. Across all network sizes, the number of middleboxes increases as the number of routers increases, as does the number of devices. Our analysis highlights several key challenges that enterprise administrators face with middlebox deployments: large deployments are expensive, requiring expertise, and a substantial size. Our data was primarily surveyed from the most of the infrastructure at the enterprise, and a cloud provider can most of the time provide better services for $5,000-50,000 dollars, and the top third of the small networks spent between $50K-500K on middlebox hardware per network.

Of the very smallest networks (e.g., homes and very small businesses), our system was observed to increase the number of personnel per network (1). Low performance overhead.
Large deployments with high cost

- Outsourcing eliminates infrastructure
- Cloud provider has economy of scale
- Time multiplexing of hardware
Complex management

- Broad expertise needed
- Upgrades and vendor interaction
- Monitoring and diagnostics
  - Cloud monitors utilization and failures
- Configuration
  - Cloud manages *appliance* configuration
  - Enterprise focuses on *policy* configuration
Common causes of failure

- Misconfiguration
  - Cloud reduces management complexity
- Overload
  - On-demand scaling
- Physical failures
  - Stand-by devices
Bounce redirection

(a) “Bounce” redirection inflates latency.
IP redirection

(b) Direct IP redirection in multi-PoP deployments cannot ensure that bidirectional traffic traverses the same PoP.
IP redirection

- Cloud can announce enterprise IPs
- Multi-PoP for improved performance
  - No guarantee of bidirectional visibility
  - Little control of PoP selection
DNS redirection

(c) DNS-based redirection minimizes latency and allows providers to control PoP selection for each request.
DNS redirection

- Control of PoP selection
- Bidirectional visibility
- Legacy applications may only provide IP addresses, not DNS names
Minimize latency with PoP choice

- Choose optimal PoP for each client and site \((c,e)\) combination
- Before: \(\min_P \text{Latency}(P, e)\)
- Smart redirection:

\[
\arg \min_P \text{Latency}(P, c) + \text{Latency}(P, e)
\]
Redirection performance

Figure 6: Round Trip Time (RTT) inflation when redirecting traffic between US PlanetLab nodes through Amazon PoPs.
Provider footprint

- Amazon can be used for **non-location dependent** boxes

![Graph](image)

**Figure 7:** PlanetLab-to-PlanetLab RTTs with APLOMB redirection through Amazon and Akamai.
Location dependent boxes

- **Wan optimization, caches, proxies**

![Figure 8: Direct RTTs from PlanetLab to nearest Akamai or Amazon redirection node.](image_url)
Location dependent boxes

- Minimize latency
  - Need Akamai-like provider footprint
- Minimize bandwidth
  - Compress traffic before reaching enterprise access link
  - APLOMB+
APLOMB can be outsourced. Authors suggest that this is not a typical requirement and thus that load balancing techniques. For example, to load balance traffic across commodity hardware.

We enable protocol-agnostic redundancy elimination [21]. Note while DNS-based services are the common example, applications may require legacy services that require fixed IP addresses. In the simplest case, enterprise may require legacy services that require fixed IP addresses. For these IP addresses, the administrator requests DNS service in the address manifest, listing the private IP address of the service, the relevant APLOMB gateway, and a DNS name. The cloud provider then manages the DNS records for this address on the enterprise client's behalf. When a DNS request for this service arrives, the cloud provider (dynamically) assigns a public IP from its own pool of IP addresses and 10.4.5.6, and the cloud provider knows that the enterprise traffic to traverse the cloud infrastructure (multihomed) can outsource close to 50% of the appliances in very large enterprises. This suggests that small and medium enterprises can achieve almost all outsourcing benefits with a basic APLOMB architecture using today's commodity hardware.

APLOMB involves an initial registration step in which administrators provide the cloud provider with an address manifest of their address allocations. The cloud provider allocates a static public IP address at a single PoP for the enterprise's gateway router, and enterprise administrators supply the cloud provider with a manifest of their address allocations. The cloud provider itself with the cloud controller (§4.3), which supplies it with a list of cloud tunnel endpoints in each PoP and forwarding rules that remain in an average small, medium, and large enterprise under different outsourcing deployment options. This suggests that small and medium enterprises can achieve almost all outsourcing benefits with a basic APLOMB architecture using today's commodity hardware.

Figure 10: Architectural components of APLOMB.

<table>
<thead>
<tr>
<th>Cloud Instances</th>
<th>APLOMB Control Plane</th>
<th>APLOMB</th>
<th>Enterprise Site</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Client Registration D</td>
<td>Middlebox Monitoring &amp; Invocation</td>
<td>Redirection Client ↔ PoP</td>
</tr>
<tr>
<td></td>
<td>DNS → IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enterprise configuration: addresses

- Enterprise provides *address manifest*
- Protected services
  - Private IP addresses
- DNS services
  - Hosts that accept incoming traffic
- Legacy IP services
  - Require fixed IP addresses
Enterprise configuration: policies

- Cloud provides policy configuration interface
  - Specify types of middlebox processing
- Enterprise specifies policy chains
- Provider may also export device-specific configuration parameters
APLOMB gateway functionality

- Maintain persistent tunnels to cloud PoPs
- Direct outgoing traffic to appropriate PoP
- APLOMB+ provides compression
APLOMB gateway
Cloud functionality

- Address resolution
- Apply middlebox functions
- Tunnel traffic to and from APLOMB gateways
- Interface for policy configuration
APLOMB architecture - cloud

APLOMB Gateway
- Local Routing Table
- Populate Routing Table
- Registration Service
- OpenVPN
- Generic RE

EC2 Datacenter
- Vyatta
- Vyatta
- Vyatta
- NAT
- OpenVPN
- Generic RE
- Policy Routing
- Measurement Node
- EC2

Cloud Controller
- MySQL
- Monitor Clients
- Calculate Routing Tables
- Update Measurements
- Launch Middlebox Instances
- Monitor Middleboxes

APLOMB Providers have a great deal of computing applications like web servers; as such we do not go into detail here.

A driving design principle for APLOMB is to keep the new components to steer packets between the above components.

In practice, the control plane is realized in a logically centralized controller.

The cloud controller is also responsible for determining the cloud PoPs, the cloud controller pushes the current best (as discussed in §3.2) tunnel selection strategies to the APLOMB gateways.

For PoP selection and utilization statistics from each middlebox to each device to verify its continued activity. In addition, the cloud controller manages and pushes middlebox policy configurations, and dynamic middlebox instances, and tunnel endpoints. It is responsible for determining this end, the APLOMB Control Plane manages the relevant network transitions in the presence of hardware and network failures. To reduce the enterprise's administrative overhead but also enables seamless access to internal private address, (2) apply middlebox processing services to the enterprise (and to encrypt/decrypt and compress/decompress if enabled) etc.

In the presence of hardware and network failures, the cloud controller launches new middleboxes and instances of that middlebox being utilized for a client's traffic.

Enterprise administrators can specify different middlebox processing policies, when they can specify different middlebox instances, and configure firewall policies. If appropriate, the provider may also require all egress traffic to go through a firewall proxy. For example, an enterprise could require all egress traffic to go through a firewall proxy. and require that all ingress traffic traverse a firewall proxy. and require that all ingress traffic traverse a firewall proxy.

Using data from heartbeat health checks on all middleboxes, the cloud controller periodically updates these tunneling configurations based on the measurement.

Middlebox instances are provided by the cloud infrastructure provider (e.g., [41, 46, 13, 42]). Our architecture is agnostic to these choices and accommodates a broad range of deployment scenarios (e.g., [34, 25, 8]). These so-called cloud middleboxes over an existing infrastructure provider.

This paper runs as a third-party service using software-based middleboxes described above. The specific implementation we present in this paper is long as there is some feasible path to implement the four components.

Standard virtual machines, providers using hardware middleboxes are simpler if all middlebox processing is performed in software on a cloud provider (e.g., [17], and (2) provider (e.g., Amazon) or by third-party cloud service providers (3) middlebox services are: (1) provided by the cloud infrastructure provider (for dynamic scaling under load are well-known for cloud computing applications like web servers [12]; as such we do not go into detail here.

In APLOMB, the APLOMB gateway, middlebox, tunneling logic to steer packets between the above components.

In practice, the control plane is realized in a logically centralized controller.

For simplicity, this discussion refers to a single logically centralized controller.
**Cloud controller**

- Manages relevant network state
  - Periodic “heartbeat” checks to all devices
- Determine optimal redirection
- Implement middlebox policy configurations
- Dynamically scales middlebox capacity
  - Network load: monitors utilization
  - Failure: heartbeat checks
APLOMB architecture - controller
Performance – HTTP page loads

![Graph of HTTP Page Load Times](image)

- **APLOMB**: Blue line
- **No Redirection**: Red dotted line

- **Y-axis**: Fraction of Websites
- **X-axis**: HTTP Page Load Time (s)

**Key Observations**
- APLOMB reduces page load times significantly compared to no redirection.
- The graph shows the cumulative distribution function (CDF) of page load times for Alexa top 1,000 websites.
- APLOMB results in faster load times, indicated by the higher CDF values for APLOMB compared to the no redirection case.

**Performance Metrics**
- **95th Percentile**: APLOMB with 0.72 seconds, No Redirection with 0.82 seconds.
- **Average**: APLOMB is faster, with lower average load times.

**Conclusion**
- APLOMB demonstrates practicality for middlebox outsourcing, improving page load times and overall performance for enterprise traffic.
Performance - BitTorrent

- Bulk transfer over long period of time
- Simultaneous connections

- Residential network
  - 2.8% average increase of download time
- University network
  - 5.5% increase
Performance - VoIP

- Demands low \textit{jitter} – variation in latency
  - 30ms target for maximum jitter
- Residential
  - With APLOMB: 2.49ms
  - Without: 2.3ms
- Public WiFi
  - With APLOMB: 13.31 ms
  - Without: 4.41 ms
  - \(~300\%\) increase
Scaling and failover

![Graph showing network load and number of instances over time.](image)
How many boxes outsourced?

- Close to 60% outsourced

<table>
<thead>
<tr>
<th># Middleboxes</th>
<th>Firewall</th>
<th>NIDS</th>
<th>Media GW</th>
<th>Load Balancer</th>
<th>Proxy/Cache</th>
<th>WAN Opt.</th>
<th>Voice</th>
<th>VPN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Today</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>APLOMB+</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cost reduction of elastic scaling

- On average, sites provision 2X the typical load

![Graph showing peak-to-mean ratio for different locations and protocols](chart.png)
Latency penalty suffered

- Pinged hosts at 11 sites from EC2
- Latency almost identical for >60% pairs
- Inflated latency from sites in Asia
- Inflation latency weighted by traffic volume
  - Typical packet has 1.13ms inflation
Bandwidth inflation

- Small enterprise used
- APLOMB: 6.2% increase
- APLOMB+: 28% reduction
- APLOMB+ improves APLOMB bandwidth usage by 32%
As more IT infrastructure moves to the cloud (see §6), worms that preferentially scanned internal prefixes, these were installed in the past to protect internal servers against.

From discussions with the network's administrators, we learned that the enterprise deploys over 100 internal firewalls.

The enterprise can outsource close to 60% of the middleboxes under a CDN.

We set out with the goal of outsourcing as many middleboxes as possible and reducing enterprise costs, all the while without incurring loss of an entire datacenter.

We measured redirection latency for inter-site traffic. Experiment used low-capacity media GW.

We found that for more than 60% of inter-site pairs, the latency with APLOMB+ is almost identical to the direct RTT.

We measured bandwidth inflation. We ran a test with experimental traffic from Internet-originated traffic, but from traffic originated within the enterprise. Internal firewalls protect a host or subnetwork not only from Internet-originated traffic, but from traffic originated within the enterprise.

Internal firewalls were installed in the past to protect internal servers against.

Figure 14 shows that the large enterprise can outsource close to 60% of the middleboxes under a CDN.

Medan inflation: 3.8%
## Paying twice for bandwidth used

- **APLOMB+** reduces bandwidth by 30%
  - Increase bandwidth when using WAN optimization
- Compression only possible with redirection
  - Still paying for uncompressed traffic leaving the cloud
  - Pay twice for compressed traffic vs. once for uncompressed
Cloud bandwidth costs

<table>
<thead>
<tr>
<th>Pricing Model</th>
<th>Total Cost</th>
<th>$/GB</th>
<th>$/Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard EC2</td>
<td>30003.20</td>
<td>0.0586</td>
<td>17.58</td>
</tr>
<tr>
<td>Amazon DirectConnect</td>
<td>11882.50</td>
<td>0.0232</td>
<td>6.96</td>
</tr>
<tr>
<td>Wholesale Bandwidth</td>
<td>6826.70</td>
<td>0.0133</td>
<td>4.00</td>
</tr>
</tbody>
</table>
Cloud bandwidth costs

- Amazon DirectConnect
  - Charges per hours in addition to bandwidth out
  - APLOMB tunnels are persistent

- Wholesale bandwidth
  - Priced by transfer rate
  - Must overprovision
Added security vulnerabilities

- Cloud has unencrypted access to traffic flows
- Existing issue with cloud computing
- Encrypts traffic too and from enterprise
- Each client has it’s own set of VMs for middlebox processing
Discussion: Is it worth it?

- Bandwidth cost vs. infrastructure savings
  - Will still be paying for hardware
  - More cost analysis

- Location-dependent middleboxes
  - Can cache and proxy functionality be preserved

- How much management complexity remains?
  - Details on policy configuration

- What are the research contributions of outsourcing to “middlebox experts”
  - Better to focus efforts on decreasing middlebox complexity?