Design and Implementation of a Consolidated Middlebox Architecture

Ashutosh Mittal (E-mail: amittal@kth.se)

I. Paper summary

THIS paper introduces *CoMb* (Consolidating Middleboxes), a novel architecture for building, deploying, and managing middlebox applications that tackles the inefficiencies by systematically exploiting opportunities for consolidation. It decouples hardware and software, thus enabling software-based implementations of middlebox applications to run on a consolidated hardware platform. The management of different middlebox applications is at a single (logically) centralized controller that takes a unified, network-wide view generating configurations and accounting for policy requirements across all traffic, all applications, and all network locations. This architecture stands in contrast to todays approach where each middlebox application and/or device is managed independently.

CoMbs network controller assigns processing responsibilities across the network that satisfies a given set of policy requirements. Each CoMb middlebox receives this configuration and allocates hardware resources to the different applications. The network controller takes three inputs: 81) AppSpec: policy constraints; (2) NetworkSpec: description of end-to-end routing paths and the location of the middleboxes; and (3) BoxSpec: hardware capabilities of middlebox hardware and the per-packet resource footprints.

Packet processing within a CoMb box comprises three logical stages. An incoming packet must first be classified, then the packet is handed to a policy enforcement layer responsible for steering the packet between the different applications corresponding to the packets traffic class, in the appropriate order. Finally, the packet is processed by the appropriate middlebox application(s).

II. Significant Contributions

- The middlebox infrastructure has developed in a largely uncoordinated manner. There is a new form of middlebox typically emerging as a one-off solution to a specific need, patched into the infrastructure through ad-hoc and often manual techniques. This paper presents a novel attempt to consolidate them into one common hardware with logical divisions and optimal resource utilization.
- Software-based solutions reduce the cost and development cycles to build and deploy new middlebox applications. Consolidating multiple applications on the same physical platform reduces device sprawl and allows the administration to have network wide view. Thus, pareto-optimal decisions can be taken and overall optimization can be achieved.
- CoMb allows for Application multiplexing, Reusing software elements and Spatial distribution. These techniques of optimal utilization caused reduced aggregate resource consumption by a factor 1.82.5 or reduced maximum per-box load by a factor 225, for a range of real-world scenarios. Also, it imposed little or minimal overhead for existing middlebox applications in the worst case, a 0.7% performance drop was recorded.

III. Unresolved Issues

Running multiple middlebox applications on the same platform raises concerns about isolation with respect to performance (e.g., contention for resources), security (e.g., the NIDS/firewall must not be compromised), and fault tolerance (e.g., a faulty application should not crash the whole system).

With respect to performance, concurrent work shows that contention has minimal impact on throughput on x86 hardware for the types of network processing workloads we envision. However, further stress testing would be required to show if its actually true in all situations.

In terms of fault tolerance and security, process boundaries already provide some degree of isolation and techniques such as containers can give stronger properties. However such sandboxing might cause high context switching overheads. Also, it would hamper the reuse capabilities of CoMb which is quite an important property.

Not only technically, but business-wise as well, this would cause some issues. Such consolidation and decoupling software and hardware in middlebox deployment would cause a ruffle in the market and force the hand of vendors to change their business models.