On the Complexity of Verifying Stateful Networks

A. Panda  S. Shenker  Y. Velner  K. Alpernas  A. Rabinovich  M. Sagiv

Berkeley
UNIVERSITY OF CALIFORNIA

European Research Council
Networks provide end-to-end connectivity

- Just contain host and switches
- All interesting processing at the hosts

Classical Networking
Ted Stevens was right
Security & Performance

- Security (firewalls, IDSs,...)
- Performance (caches, load balancers,...)
- New functionality (proxies,...)
Middleboxes

• Middleboxes are intermediaries
  – Interposed in-between the communicating hosts
  – Often without knowledge of one or both parties

• Examples
  – Network address translators (NAT)
  – Firewall
  – Traffic shapers
  – Intrusion detection systems (IDSs)
  – Transparent Web proxy caches
  – Application accelerators
NAT

<table>
<thead>
<tr>
<th>local</th>
<th>prt</th>
<th>global</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>1</td>
<td>138.76.29.7</td>
</tr>
</tbody>
</table>
Firewalls

Trusted Hosts

<table>
<thead>
<tr>
<th>Trusted Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
</tr>
</tbody>
</table>
Learning Switch

A on 1
D on 3

B

C

A

D
Web Clients and Servers

• Most Web applications use client-server protocol
  – Client sends a request
  – Server sends a response
• Proxies play both roles
  – A server to the client
  – A client to the server
Two Views of Middleboxes

• An abomination (toevah)
  – Violation of layering
  – Breaks the functional model
  – Responsible for many subtle bugs

• A practical necessity
  – Significant part of the network
  – Solving real and pressing problems
  – Needs that are not likely to go away
  – Local functionality enhancements
Reachability/Isolation

• Reachability:
  • A packet of type t sent from host A may reach host B
    • packets from an ATM to the bank database

• Isolation:
  • A packet of type t sent from host A never reaches host B
    • packets from a customer's cell phone to the bank database
Firewall Misconfiguration

A is isolated from B
Research Question

Can we algorithmically prove that a given network of middleboxes establishes isolation between two given hosts for certain types of and packets?
Assumptions

- Finite set of hosts $H$ (extended later)
- Finite set of packet types $T$
- Finite set of ports $Pr$ per middlebox
- Finite set of packet headers
  $P = t: T \times src: H \times dst: H \times pr: Pr$
- Fixed set of middleboxes $M$
- Fixed undirected topology
  $E \subseteq (H \times Pr \times M) \cup (M \times Pr \times Pr \times M)$
- The semantics of each middlebox is a function
  - $m: P^* \times P \rightarrow 2^P = P^* \rightarrow (P \rightarrow 2^P)$
  - Packet bodies are unchanged
Middlebox classification

Finite State Middleboxes

Eventually stateless

Increasing

Firewall

Stateless

Switch

Nat

Decreasing

IDS

Learning Switch

Cache

Load Balancer
Stateless Middleboxes

• Behavior independent of the history
• For all $h, h' \in P^*$:
  – $m(h) = m(h')$
  – For all $p \in P$: $m(h, p) = m(h', p)$

• Examples
  – Switches and Routers
  – ACL Firewall
  – Simple load-balancer
Increasing Middleboxes

• Forwarding behavior increased over time
• For all $h, h' \in P^*$ such that $h \sqsubseteq h'$ and $p \in P$:
  $m(h, p) \subseteq m(h', p)$
• Good examples
  – Stateless
  – Firewall
• Bad Examples
  – Learning Switch
  – Cache
Decreasing Middleboxes

• Forwarding behavior decreased over time
• For all $h, h' \in P^*$ such that $h \sqsubseteq h'$ and $p \in P$:
  – $m(h, p) \supseteq m(h', p)$
• Good Examples
  – Stateless
  – Learning switch
  – IDS?
Eventually Stateless Middleboxes

• Forwarding eventually constant for all histories
• There exists some \( k \)
  – For all histories \( h_1, h_2, \ldots, h_k \) such that \( h_i \sqsubseteq h_{i+1} \)
    for all \( p \in P \) if \( m(h_1, p), \ldots, m(h_{k-1}, p) \) are all distinct
    then \( m(h_k, p) = m(h_{k-1}, p) \)
• Good examples
  – Learning switches
  – IDS?
  – Cache
• Bad Examples
  – Round-robin load balancer
Modeling Middliboxes by FSMs

• A Mealy Machine $m = \langle S, s_0, P, f, \delta \rangle$
  where
  – $S$ are the states of the middleboxes
  – $s_0 \in S$ is the initial state
  – $f: S \times P \to 2^P$ is the current forwarding behavior
  – $\delta: S \times P \to 2^S$ is the next state
  – Extend $\delta$ to histories
    • $\delta([\ ])) = \{s_0\}$
    • $\delta(h . p) = \delta(\delta(h), p))$
• $m$ models $m: P^* \times P \to 2^P$ when for all $h \in P^*$ and $P \in P$:
  – $f(\delta(h), p) = m(h, p)$
Partial FSM for Firewall
Middlebox classification

Finite State Middleboxes

Eventually stateless

Increasing

Firewall

Stateless

Switch

Nat

Decreasing

IDS

Learning

Switch

Cache

Load Balancer
Decidability

• When the order of packet arrival must be respected checking isolation is undecidable even for finite state middleboxes
  – Cycles in the topology allows counting

• When the packet are processed in arbitrary order checking reachability and isolation becomes decidable even for finite state machines
  – Reduction to Well-Quasi-Orders
  – But complexity is high
    • EXPSPACE-Complete
Complexity Results

Finite State Middleboxes

Eventually stateless

Increasing

Decreasing

Stateless

Polynomial

NP-complete

NP-complete

EXPSPACE - complete
Abstract Middlebox Definition Language

- Powerful enough to express the behavior of interesting middleboxes
- Succinct
  - Sometimes exponential state saving
- Simple enough for analysis
- Lends itself to classification of middleboxes
  - Same worst case complexity
  - But sometimes exponential saving
Firewall (AMDL)

BEGIN_MBOX(firewall)

#define TRUSTED_PORT 0

INIT

(host) TrustedHosts = {};

RECV(s, d, t, port) ⇒

if( port == TRUSTED_PORT || TrustedHosts (s) )
    if( port == 0 )
        forward(s,d,t, 1);
    if( port == 1 )
        forward(s,d,t, 0);
    TrustedHosts.ADD(d);

/* else – packet is discarded */

END_MBOX
Firewall vs. FSM

BEGIN_MBOX(firewall)
#define TRUSTED_PORT 0
INIT
(host) TrustedHosts = {};
RECV(s, d, t, port) \Rightarrow
  if(port == TRUSTED_PORT ||
    TrustedHosts (s))
    if( port == 0 )
      forward(s,d,t,1);
    if( port == 1 )
      forward(s,d,t,0);
    TrustedHosts.ADD(d);
/* else - packet is discarded */
END_MBOX
The MVer Toolset

Counterexample

AMDL spec

Front-End

First Order Formula

Datalog+ Program

Petri-Net

Z3

μZ

Lola
(Some) Related Work

**Dynamic**
- Veriflow
  - Online verification
  - Handles dynamic networks pretty well
- Header Space Analysis
  - Offline and online verification

**Static**
- Firewall Verification
  - Margrave
    - Statetless
    - Does not handle loops
- SDN
  - Netkat
    - Verification of Reachability
  - Vericon
    - Arbitrary code
    - Undecidability
Summary

• Middlebox classification

• Complexity results

• Initial toolset
Acknowledgments

• The Noun Project
• Nate Foster, Michael Freedman, and Jane Rexford