Revisiting Network Security Research and Teaching using OpenFlow

Syed Ali Khayam

Wireless and Secure Networks (WiSNet) Research Lab
School of Electrical Engineering & Computer Science
National University of Sciences & Technology (NUST), Pakistan
This work is done collaboratively with Prof. Scott Shenker's group at UC Berkeley. Students involved are:
- Syed Akbar Mehdi
- Junaid Khalid
- Mohsin Sardar
- Murphy McCauley
- Kyriakos Zarifis

Our research is supported by Pakistan National ICT R&D Fund

We thank Wahaj-us-Siraj and Jahenzeb Ahmad of Micronet Broadband for sharing their datasets with us

I am not going to reinvent the wheel and blatantly copy slides (with reference) from Nick McKeown's group at Stanford
NUST was established as an Engineering University in 1991

Since then, it has expanded its scope to include Basic, Medical, Management and Social Sciences as well

The University main campus is located in Islamabad
About NUST: Distributed NUST Campuses
About NUST: New Campus

- Academic Block
- Sport Complex
- Students Hostel
- Faculty & Staff Residency
- 700 Bed (Teaching hospital)
About NUST: NUST Among Top 400 Universities of the World

Declared:
- No. 1 in PAKISTAN
- No. 20 in ASIA
About NUST-SEECS: University Collaborations

Canada

USA

France
Germany
Switzerland
Sweden
UK
Denmark

China
Korea
Japan
Thailand
Singapore

Australia
About WiSNet: Focal Research Areas

- Network and Information Security
- Multimedia Communication
- Wireless Networks

Software Defined Networking
About WiSNet: Funded Projects

National ICT R&D Fund

Nokia Research Center

NRF National Research Foundation of Korea

ETRI Electronics and Telecommunications Research Institute
About WiSNet: Selected Journal Publications
About WiSNet Lab: 4 USPTO Pending Patents

United States Patent Application Publication

Khayam et al.

(19) United States
(12) Patent Application Publication
(43) Pub. Date: Jul. 30, 2009

(22) Filed: Jan. 29, 2009

Related U.S. Application Data

(60) Provisional application No. 61/062,929, filed on Jan. 30, 2008.

Publication Classification

(51) Int. CL
H04L 12/26 (2006.01)

(52) U.S. Cl. -------------------------- 379/252

(57) ABSTRACT

A method is provided for estimating the header of a data packet in a wireless communication system. The method includes: maintaining a list of data packets received without an error at a receiver; receiving at the receiver a corrupt data packet having errors in its header; computing a likelihood score for the header of the corrupt data packet in relation to each entry in the list of data packets; and selecting an entry having the highest likelihood score as an estimated header for the corrupt data packet.

United States Patent Application Publication

Radha et al.

(19) United States
(12) Patent Application Publication
(43) Pub. Date: May 21, 2009

(22) Filed: Jun. 2, 2006

Related U.S. Application Data

(60) Provisional application No. 60/637,512, filed on Jun. 3, 2005.

Publication Classification

(51) Int. CL
G06F 21/00 (2006.01)
G06C 7/02 (2006.01)
G06C 17/10 (2006.01)

(52) U.S. Cl. -------------------------- 726/24; 703/22; 703/2

(57) ABSTRACT

A worm propagation modeling system for use with a mobile ad-hoc network (MANET) includes an infection detection module receiving temporal dynamics information relating to temporal dynamics of worm spread in the MANET and spatial dynamics information relating to spatiality of nodes in the MANET. The infection detection module detects infection in a network segment of the MANET based on the temporal dynamics information and the spatial dynamics information.
Software Defined Networking and OpenFlow
What missing in today’s networks?

Abstraction

The Evolution Of Computer Programming Languages


Some rights reserved by dullhunk under a Creative Commons license.
What missing in today’s networks?

Abstraction
What missing in today’s networks?

- Imagine a computer network which is abstracted using principles similar to those used in OS and programming abstractions.

- An abstracted network simplifies the life of a network admin:
  - Perform network extensions by dragging, dropping and connecting nodes in the network using a visual tool.
  - Apply access and policy controls without even knowing the IP address of a person/host.
  - Look at all the ports in a network as one big logical switch which can be arbitrarily configured.
  - Etc. etc.

- This concept is called Software Defined Networking as a lot of this abstraction would have to be programmed and evolved in software.
Closed to Innovation

Slide from Nick McKeown (Stanford)
Software Defined Networks

- In an SDN, a network-wide software controls all the network nodes
  - This software can be changed as requirements or traffic behaviors evolve

- In the present SDN literature, this software is called by many names:
  - Network Controller or simply Controller
  - Network Operating System
  - Network Hypervisor ....
Software Defined Networks

Network Operating System

Operating System

Specialized Packet Forwarding Hardware

Slide from Nick McKeown (Stanford)
Software Defined Networks

- While we would like to control network behavior in software, we still want to forward packets at line rates (1/10/40/... Gbps)

- To achieve this architecturally, we have to separate the network control and data paths
  - **Data Path**: Simply forwards packets using *rules* defined in a flow table. If no rule is found, forward the packet to the controller.
  - **Control Path**: Decides of what to do with a packet. Once decision is made, puts rule in the switch's flow table.
An interface is needed to allow the controller and switch to talk to each other.

- The OpenFlow protocol provides this interface.
SDN Implementation using OpenFlow

1. Open and standard interface to hardware

2. At least one good operating system
   Extensible, possibly open-source

3. Well-defined open API

App

Network Operating System

Simple Packet Forwarding Hardware

Simple Packet Forwarding Hardware

Simple Packet Forwarding Hardware

Slide from Nick McKeown (Stanford)
OpenFlow: Architecture

Ethernet Switch

Control Path | OpenFlow | OpenFlow Protocol (SSL/TCP) | OpenFlow Controller

Data Path (Hardware)

Slide from Rob Sherwood (Stanford)
OpenFlow: Architecture

Software Layer

Hardware Layer

OpenFlow Firmware

Flow Table

<table>
<thead>
<tr>
<th>MAC src</th>
<th>MAC dst</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP sport</th>
<th>TCP dport</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>5.6.7.8</td>
<td>*</td>
<td>port 1</td>
</tr>
</tbody>
</table>

PC

Controller

1.2.3.4

5.6.7.8

5.6.7.8

port 1

port 2

port 3

port 4

Slide from Rob Sherwood (Stanford)
OpenFlow: Flow Table (Specification 1.0)

- Rule
- Action
- Stats

Packet + byte counters

- Forward packet to port(s)
- Encapsulate and forward to controller
- Drop packet
- Send to normal processing pipeline
- Modify Fields

+ mask what fields to match

Slide from Rob Sherwood (Stanford)
### OpenFlow: Examples

#### Switching

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Eth type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>IP Prot</th>
<th>TCP sport</th>
<th>TCP dport</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>* 00:1f:..</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>port6</td>
</tr>
</tbody>
</table>

#### Flow Switching

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Eth type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>IP Prot</th>
<th>TCP sport</th>
<th>TCP dport</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>port3</td>
<td>00:20..</td>
<td>00:1f</td>
<td>0800</td>
<td>vlan1</td>
<td>1.2.3.4</td>
<td>5.6.7.8</td>
<td>4</td>
<td>17264</td>
<td>80</td>
</tr>
</tbody>
</table>

#### Firewall

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Eth type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>IP Prot</th>
<th>TCP sport</th>
<th>TCP dport</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>22</td>
</tr>
</tbody>
</table>

Slide from Rob Sherwood (Stanford)
### OpenFlow: Examples

**Routing**

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Eth type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>IP Prot</th>
<th>TCP sport</th>
<th>TCP dport</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>5.6.7.8</td>
<td>*</td>
<td>port6</td>
</tr>
</tbody>
</table>

**VLAN Switching**

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Eth type</th>
<th>VLAN ID</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>IP Prot</th>
<th>TCP sport</th>
<th>TCP dport</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>00:1f..</td>
<td>*</td>
<td>vlan1</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>port6, port7, port9</td>
</tr>
</tbody>
</table>
Open Networking Foundation Pursues New Standards

By JOHN MARKOFF
Published: March 22, 2011

MOUNTAIN VIEW, Calif. — Acknowledging that so-called cloud computing will blur the distinctions between computers and networks, about two dozen big information technology companies plan to announce on Tuesday a new standards-setting group for computer networking.

The group, to be called the Open Networking Foundation, hopes to help standardize a set of technologies pioneered at Stanford and the University of California, Berkeley, and meant to make small and large networks programmable in much the same way that individual computers are.

The members of the Open Networking Foundation will include Broadcom, Brocade, Ciena, Cisco, Citrix, Dell, Deutsche Telekom, Ericsson, Facebook, Force10, Google, Hewlett-Packard, I.B.M., Juniper, Marvell, Microsoft, NEC, Netgear, NTT, Riverbed Technology, Verizon, VMWare and Yahoo.
OpenFlow-at-Home: A Network Security Use Case
OpenFlow-at-Home

- Networked devices in a home network are continuously growing

- Home network users do not have the expertise to configure or manage their networks

- Home devices are inflexible, proprietary and difficult to manage

- SDNs offer a natural opportunity to flexibly manage home networks
Securing Home Networks using OpenFlow-at-Home

- Home computers are often compromised with malware, mostly without the knowledge of the user.

- It is well-known that the best location to detect threats is close to the source (i.e. home networks).

- But ISPs have little or no visibility into or control over home networks' traffic.

- SDNs can be used to delegate the task of network security to the home network without bothering the home user with complex security management tasks.
Traffic Anomaly Detection using OpenFlow-at-Home

- We use SDNs to revisit a network security concept that has failed in the network core: Traffic Anomaly Detection

- ADSs in the network core have been plagued by:
  - Low accuracies
  - Line rate complexity

- Both problems can be solved if anomaly detection is performed close to the source
Anomaly Detectors

**TRW [Oakland’04]**
- Statistical detector
- Specific for TCP scanning infections
- Operates on per host basis

**Rate Limiting [Usenix’03]**
- Rate throttling and monitoring
- Specific for scanning infections
- Operates on per host basis

**Maximum Entropy [IMC’06]**
- Information theoretic detector
- General purpose
- Operates on time-windowed stats

**NETAD [SAC’03]**
- Anomalous value detector
- General purpose
- Operates on per packet basis
## Benign Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Active Hosts</th>
<th>Total Connections</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>8</td>
<td>3,422</td>
<td>21hrs 13mins</td>
</tr>
<tr>
<td>SOHO</td>
<td>29</td>
<td>50,082</td>
<td>5hrs 28mins</td>
</tr>
<tr>
<td>ISP</td>
<td>639</td>
<td>304,914</td>
<td>9 mins</td>
</tr>
</tbody>
</table>

## Attack Datasets

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Active Hosts (Home, SOHO, ISP)</th>
<th>Attack rates (pkts/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Portscan, TCP SYN Flood, UDP Flood</td>
<td>3, 8, 24</td>
<td>0.1, 1, 10, 100, 100</td>
</tr>
</tbody>
</table>
Implementation

NOX Box

Average: 687 lines of code/anomaly detector

**NOX Box Specs**

- OpenFlow 1.1
- NOX Controller
- Voyage Linux
- 500 MHz AMD Processor
- 512 MB RAM
Accuracy

TCP Portscans

High rate attacks

Low rate attacks
TCP SYN Flood

High rate attacks

Low rate attacks
Accuracy

UDP Flood

High rate attacks

Low rate attacks
## Efficiency

<table>
<thead>
<tr>
<th></th>
<th>Total pkts handled by controller</th>
<th>Avg flow table size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HOME</strong></td>
<td>Min-TRW: 1%</td>
<td>Min-Rate Limiting: 16</td>
</tr>
<tr>
<td></td>
<td>Max-NETAD: 3.46%</td>
<td>Max-Rate Limiting: 40</td>
</tr>
<tr>
<td><strong>SOHO</strong></td>
<td>Min-TRW: 0.37%</td>
<td>Min-Rate Limiting: 38</td>
</tr>
<tr>
<td></td>
<td>Max-MaxEnt: 1.26%</td>
<td>Max-Rate Limiting: 172</td>
</tr>
</tbody>
</table>

## CPU Usage

<table>
<thead>
<tr>
<th>Traffic rate</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mbps</td>
<td>1.9% (TRW)</td>
<td>3.1% (MaxEnt)</td>
</tr>
<tr>
<td>50 Mbps</td>
<td>17.5% (TRW)</td>
<td>28.2% (MaxEnt)</td>
</tr>
</tbody>
</table>
OpenFlow as a Teaching Tool
OpenFlow in teaching

- Computer networks teaching has become extremely boring and monotonous over the years

- A fundamental bottleneck is the inability to experiment with existing and new protocols

- SDN can be used to revamp networking classrooms

- We are planning to do that in the upcoming fall semester
Lab Experimentation on an OpenFlow Campus Slice

Marvell OpenFlow switches are being deployed in:

– EE Research Labs at NUST
– NOC School of EECS
– NUST Central NOC
Assignments and Projects on NOX-on-a-Plug

We are porting a wireless OpenFlow switch and the NOX controller on Marvell’s Plug Computer

Students will do all their assignments and projects on this platform
Contact Information:

- Email: ali.khayam@seecs.nust.edu.pk
- Research Lab Webpage: http://wisnet.seecs.nust.edu.pk