A Testbed for GRMP Protocol
(draft-wang-forces-grmp-01.txt)

Ligang Dong (presenter), Lei Shi, and Weiming Wang,
(donglg@mail.hzic.edu.cn)

College of Information & Electronic Engineering
Hangzhou University of Commerce, P.R.China

March 2 2004
Content

- Introduction to Testbed
- Experiments on Testbed
Introduction to Testbed - Deployment

- **CE host**
- **FE host 1**
- **FE host 2**
- **FE host n**
- **Hub**
- **SmartBits**

**Control plane packets**

**Forwarding plane packets**

**analysis host (SmartBits software installed)**
The CE located in the CE host takes the responsibility to control FEs.
The messaging system between the CE and the FEs follows GRMP.
Control plane packets can be sent between CE and FEs via TCP, UDP, or IP socket connection.
• The CE is realized using a **VC++** software that runs in **Windows OS.** (software)
• The CE can **send** various kinds of messages, whose parameters can be **specified via dialog boxes.**
Introduction to Testbed - CE software
The FE is realized using an adapted Linux kernel, which can forward packets and communicate with the CE.
Introduction to Testbed - Smartbits

• The smartbits is used to generate various rates of packets and measure the rates of received packets.
Experiments on Testbed - Purpose

- Check the implementation feasibility of GRMP
- Currently testify the strategies for DoS attack protection policy
GRMP: FE architecture

Transmitter/Receiver

FE Slave Model

Scheduler

FE Model (about the definition of various LFBs and datapath which connects multiple LFBs)

Encapsulator of redirection info.

Redirection msg.

Msg. to be send

Info./Policy database

Control msg.

Msg. interpreter

Msg. generator

Redirection info.

Received msg.

FE

Fp

CE

Packets

Messages

Control Info.
Experiments on Testbed - Setting

- The smartbits generates and sends common data packets and redirection data packets to FEs.
- The CE generates and sends pseudo control packets to FEs.
  - As the rate of pseudo control packets can be controlled, we use pseudo control packets instead of "real" control packets for performance comparison.
Experiments on Testbed - Setting (cont.)

- When the **FE** receives packets from the **CE**, the FE sends back *reply packets* according to the test demand.
- When the **FE** receives packets from the **smartbits**, the FE *forwards or redirects* received packets to the smartbits or CE respectively according to the setting status of the classifier LFB.
Experiments on Testbed - Experiment I

**Content**

- **Before uploading a classifier LFB,** *All* the packets sent by the *smartbits* are received by smartbits. *(screen record)*
- The CE sends control messages to the FE and inform a FE host to **upload the classifier LFB.** *(screen record)*
- **After uploading,** the redirection data packets are received by the **CE** while the *common* data packets are still received by the *smartbits.* *(screen record)*

**Result**

- It testifies that we implemented the *messaging system* and **dynamic loading** of the classifier LFB.
Experiments on Testbed - Experiment II

- **Content**

  - Control messages → $Q_c$ → Scheduler → Tran./Recv
  - Redirection messages → $Q_r$ → Scheduler

- **We change the scheduling strategy (for DoS Protection Policy) in the FE in terms of the control from the CE according to GRMP**
  - FCFS strategy
  - Priority-Based strategy
  - Rate-Based strategy
Experiments on Testbed - Metrics

• DoS attack from the redirection packets may congest the communication channel
• Currently used performance metric: the loss rate of packets.
  • Packets may be lost in the waiting queues when congestion happens. The more the packets lost, the more serious the congestion is.
## Experiments on Testbed – Experiment II

### Table 1: Loss rate of Control and Redirection packets in the case of one TCP (when the rate of redirection packets varies)

<table>
<thead>
<tr>
<th>Packet Rates (% of 10Mbs)</th>
<th>( r_c=50% )</th>
<th>( r_c=50% )</th>
<th>( r_c=50% )</th>
<th>( r_c=50% )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( r_r=20% )</td>
<td>( r_r=50% )</td>
<td>( r_r=80% )</td>
<td>( r_r=150% )</td>
</tr>
<tr>
<td><strong>FCFS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>control</td>
<td>0.3%</td>
<td>0.3%</td>
<td>21%</td>
<td>49%</td>
</tr>
<tr>
<td>redirect.</td>
<td>0.3%</td>
<td>0.3%</td>
<td>21%</td>
<td>49%</td>
</tr>
<tr>
<td><strong>Priority-Based</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>control (H)</td>
<td>0.3%</td>
<td>0.4%</td>
<td>21%</td>
<td>21%</td>
</tr>
<tr>
<td>redirect. (L)</td>
<td>0.2%</td>
<td>0.4%</td>
<td>22%</td>
<td>67%</td>
</tr>
<tr>
<td><strong>Rate-Based (with RR)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control (no rate limit)</td>
<td>0.3%</td>
<td>0.2%</td>
<td>0.1%</td>
<td>0.1%</td>
</tr>
<tr>
<td>redirect. (50% rate limit)</td>
<td>0.3%</td>
<td>0.2%</td>
<td>38%</td>
<td>67%</td>
</tr>
</tbody>
</table>
Analysis of Table 1:
- FCFS scheduling strategy cannot prevent the DoS attack
- Priority-Based strategy can prevent the DoS to some extent.
- Rate-Based (with Round Robin) strategy can prevent the DoS attack quite well.
Experiments on Testbed – Experiment II

• Table 2: Loss rate of Control/Redirection packets in the case of one TCP (when the rate of control packets varies)

<table>
<thead>
<tr>
<th>Packet Rates (percentage of 10Mbs)</th>
<th>$r_c=20%$</th>
<th>$r_c=50%$</th>
<th>$r_c=80%$</th>
<th>$r_c=150%$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r_r=50%$</td>
<td>$r_r=50%$</td>
<td>$r_r=50%$</td>
<td>$r_r=50%$</td>
</tr>
<tr>
<td>Priority-Based</td>
<td>control</td>
<td>0.2%</td>
<td>0.4%</td>
<td>13%</td>
</tr>
<tr>
<td></td>
<td>redirect.</td>
<td>0.2%</td>
<td>0.4%</td>
<td>52%</td>
</tr>
</tbody>
</table>

It shows that the control packet may congest the redirection packet too. In particular when we use priority-based strategy, it is possible that all of redirection packets can be congested.
Experiments on Testbed – Experiment II

**Table 3: Loss rate of C and R packets in the case of TCP channel for C packets and UDP channel for R packets**

<table>
<thead>
<tr>
<th>Packet Rates (percentage of 10Mbs)</th>
<th>$r_c=50%$</th>
<th>$r_c=50%$</th>
<th>$r_c=50%$</th>
<th>$r_c=50%$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r_r=20%$</td>
<td>$r_r=50%$</td>
<td>$r_r=80%$</td>
<td>$r_r=150%$</td>
</tr>
<tr>
<td>Scheduling strategy from Linux kernel</td>
<td>Control (TCP channel)</td>
<td>0.3%</td>
<td>2.4%</td>
<td>67%</td>
</tr>
<tr>
<td></td>
<td>Redirect. (UDP channel)</td>
<td>0.2%</td>
<td>0.3%</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

It shows that control packets in the TCP channel will **be affected** by redirection packets in the UDP channel, meaning a DoS attack cannot be prevented purely by such separation.
Experiments on Testbed - Conclusion

• Current experiments testify the following facts
  • GRMP is feasible for implementation of
    • messaging, LFB uploading, ...
  • Scheduling at the ForCES protocol level is effective for DoS attack protection.
    • Moreover, in this case, the scheduling strategies can be flexibly selected.
  • DoS attack cannot be avoided ONLY by separating Control packets and Redirection packets to different channels.
  • It is possible that Control packets may congest Redirection packets in some cases.
The End