RTG:
A Scalable SNMP Statistics Architecture

NANOG 27

Robert Beverly
February 10, 2003
Background: What’s the Problem?

• SNMP: Simple Network Mgmt Protocol
• Despite “Simple,” Many Issues:
  – Scaling in Large Installations
  – Storage Retention (Length/Granularity)
  – Report Generation Time (Interactivity)
  – Reporting Flexibility
  – Robustness, statistics as a critical component:
    • Legal (Culpability)
    • Billing
Service Provider Requirements

• Service Providers and Large Enterprises have unique SNMP requirements

• High-Level Requirements:
  – Support for 100’s of devices each with 1000’s of objects (high speed)
  – Disjoint polling, storage and reporting
  – Ability to retain data indefinitely
  – Provide an abstract interface to data in order to generate complex and/or custom reports
  – Flexibility (distributed polling, sub-minute polling, no averaging, etc.)
Possible Solutions

• Commercial Packages:
  – Typically large, bloated, expensive, difficult to manage
• Generally, open-source packages can not complete polling within 5-minute interval
• MRTG:
  – Scaling Problems
  – Little flexibility
• RRDtool/Cricket:
  – Good scaling (can we do better?), no abstract data interface
RTG History

- Name: too busy writing software to be inventive!
- Motivation: limitations of available tools (both open-source and commercial packages)
- RTG began as an experiment at a large service provider
- Possible to develop fast, scalable and flexible SNMP statistics tool?
- First implementation suffered from speed and architectural problems
- Opportunity to completely rewrite from scratch
RTG: Real Traffic Grabber

- Flexible, modular and scalable high-performance SNMP monitoring system
- Runs as a daemon on UNIX platforms
- All data inserted into a relational database
- Intelligent database schema to retain long-term data without speed degradation
- Traffic reports, plots, web-interface
- Can poll at sub-one-minute intervals
- Clean separation of polling, storage and reporting to easily distribute load
- RTG designed as a foundation to build upon
RTG Operation

• Auxiliary Perl script queries network for new interfaces and changed ifIndex or description.
• Generates an RTG “target list”
• RTG poller randomizes objects in the “target list”
  – Limits SNMP query impact on network devices
  – Improves performance
• All data is inserted into a MySQL database
• Reports and Graphs generated via APIs to MySQL (Perl DBI, PHP, C)
Database Schema

• Non-trivial
  – Better schemas for different environments
  – RTG poller is indifferent to schema
• Need to retain long-term historical data (ideally indefinitely):
  – Legal/Billing
  – Disks are cheap, but keep as little data as possible
• Query execution time should be independent of time period requested:
  – Generating a report for a day one year ago should be as fast as generating today’s report
• Schema described in USENIX LISA 2002 paper
RTG Speed

- What makes RTG fast?
  - Daemon – No cron overhead
  - Written in C – No interpreter overhead
  - Multi-threaded:
    - Keep a constant number of “queries-in-flight”
    - Exploit Natural Parallelism in Slow I/O
    - Use multiple processors
  - Randomized targets:
    - An unresponsive device does not block all threads
### RTG Speed (Some Numbers)

<table>
<thead>
<tr>
<th>App</th>
<th>Targets (#OIDs)</th>
<th>Run Time (sec)</th>
<th>Targs/sec</th>
<th>Max Targs (in 5min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRTG</td>
<td>1618</td>
<td>365.4</td>
<td>4.4</td>
<td>1328</td>
</tr>
<tr>
<td>Cricket</td>
<td>2010</td>
<td>87.8</td>
<td>22.9</td>
<td>6868</td>
</tr>
<tr>
<td>RTG (v0.7.3)</td>
<td>2010</td>
<td>7.7</td>
<td>261.4</td>
<td>78423</td>
</tr>
</tbody>
</table>

- See USENIX LISA 2002 Paper for Full Comparison and Analysis (see link from http://rtg.sf.net)
RTG Output

• RTG includes various tools and scripts to generate useful output:
  – PHP scripts to generate MRTG-style output and 95th percentile reports
  – PHP script to interactively generate plots for arbitrary time periods
  – Perl scripts to summarize traffic, provide 95th percentile

• Can insert RTG plot easily in any web page with appropriate <IMG=rtgplot.cgi?> tag

• Specify any arbitrary time period
• Easy to build custom web pages
# RTG Reports

- Perl DBI scripts included
- Automate reporting, etc.
- Scripts to monitor thresholds, etc.

## Traffic Daily Summary
**Period:** [01/01/1979 00:00 to 01/01/1979 23:59]

<table>
<thead>
<tr>
<th>Site</th>
<th>GBytes In</th>
<th>GBytes Out</th>
<th>MaxIn (Mbps)</th>
<th>MaxOut</th>
<th>AvgIn</th>
<th>AvgOut</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtr1.someplace:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>so-5/0/0</td>
<td>384.734</td>
<td>360.857</td>
<td>49.013</td>
<td>43.420</td>
<td>35.630</td>
<td>33.426</td>
</tr>
<tr>
<td>so-6/0/0</td>
<td>357.781</td>
<td>421.736</td>
<td>42.923</td>
<td>50.861</td>
<td>33.137</td>
<td>39.053</td>
</tr>
<tr>
<td>t1-1/0/0</td>
<td>0.054</td>
<td>0.058</td>
<td>0.005</td>
<td>0.006</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>rtr3.someplace:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>so-6/0/0</td>
<td>1,115.258</td>
<td>1,246.163</td>
<td>168.776</td>
<td>172.690</td>
<td>103.173</td>
<td>115.439</td>
</tr>
<tr>
<td>so-3/0/0</td>
<td>1,142.903</td>
<td>1,028.256</td>
<td>152.232</td>
<td>162.402</td>
<td>105.863</td>
<td>95.142</td>
</tr>
<tr>
<td>so-7/0/0</td>
<td>152.824</td>
<td>199.742</td>
<td>22.052</td>
<td>35.005</td>
<td>14.152</td>
<td>18.488</td>
</tr>
</tbody>
</table>
### RTG Reports (95th Percentile)

**ABC Industries Traffic**  
**Period:** [01/01/1979 00:00 to 01/31/1979 23:59]

<table>
<thead>
<tr>
<th>Connection</th>
<th>RateIn Mapping</th>
<th>RateOut Mapping</th>
<th>MaxIn Mapping</th>
<th>MaxOut Mapping</th>
<th>95% In Mapping</th>
<th>95% Out Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>at-1/2/0.111 rtr-1.chi</td>
<td>0.09 Mbps</td>
<td>0.07 Mbps</td>
<td>0.65 Mbps</td>
<td>0.22 Mbps</td>
<td>0.22 Mbps</td>
<td>0.13 Mbps</td>
</tr>
<tr>
<td>at-1/2/0.113 rtr-1.dca</td>
<td>0.23 Mbps</td>
<td>0.19 Mbps</td>
<td>1.66 Mbps</td>
<td>1.12 Mbps</td>
<td>0.89 Mbps</td>
<td>0.57 Mbps</td>
</tr>
<tr>
<td>at-3/2/0.110 rtr-2.bos</td>
<td>0.11 Mbps</td>
<td>0.16 Mbps</td>
<td>0.34 Mbps</td>
<td>0.56 Mbps</td>
<td>0.26 Mbps</td>
<td>0.40 Mbps</td>
</tr>
</tbody>
</table>
## RTG PHP 95th Percentile

**Report**: 95th percentile  
**Customer**: ABC  
**Period**: 01/29/2003 20:25 - 01/30/2003 20:25

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Router</th>
<th>Current Rate</th>
<th>Max Rate</th>
<th>95th %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>In (Mbps)</td>
<td>Out (Mbps)</td>
<td>In</td>
</tr>
<tr>
<td>1-1/3/2:0</td>
<td>ABC Industries Chicago</td>
<td>rtr1.chi.my.net</td>
<td>0.06</td>
<td>0.07</td>
<td>1.52</td>
</tr>
<tr>
<td>at-5/1/0.113</td>
<td>ABC Industries Boston</td>
<td>rtr4.bos.my.net</td>
<td>42.30</td>
<td>28.46</td>
<td>63.49</td>
</tr>
<tr>
<td>ml-2/1/0.33</td>
<td>ABC Industries Manufacturing Dallas</td>
<td>rtr3.dfw.my.net</td>
<td>1.46</td>
<td>2.08</td>
<td>8.57</td>
</tr>
<tr>
<td>FastEthernet2/2</td>
<td>ABC Industries Web Server</td>
<td>rtr2.sfo.my.net</td>
<td>40.60</td>
<td>22.29</td>
<td>66.04</td>
</tr>
<tr>
<td>so-2/0/0</td>
<td>ABC Industries Atlanta</td>
<td>rtr2.atl.my.net</td>
<td>69.81</td>
<td>2.19</td>
<td>203.28</td>
</tr>
<tr>
<td>ml-3/0/1.23</td>
<td>ABC Industries Washington</td>
<td>rtr4.iad.my.net</td>
<td>0.41</td>
<td>0.68</td>
<td>0.87</td>
</tr>
</tbody>
</table>

**RTG** Version 0.7.3
RTG Traffic Graphs

Graph 1:
- **ifInOctets_111** Max: 337.0 Mbits/s Avg: 187.9 Mbits/s Cur: 99.1 Mbits/s [ 2.0 T]
- **ifOutOctets_111** Max: 352.5 Mbits/s Avg: 203.1 Mbits/s Cur: 114.0 Mbits/s [ 2.2 T]

Graph 2:
- **ifInOctets_111** Max: 363.8 Mbits/s Avg: 168.8 Mbits/s Cur: 119.1 Mbits/s [ 52.7 T]
- **ifOutOctets_111** Max: 394.5 Mbits/s Avg: 185.6 Mbits/s Cur: 125.8 Mbits/s [ 58.0 T]
Sub-Minute Polling

- RTG’s speed and architecture allow for sub-one minute polling
- Finer sampling granularity often reveals important traffic characteristics hidden in typical 5-minute aggregates
- Particularly useful as a diagnostic on slower links with bursty traffic
- Example (same interface and period, 30s vs 300s samples):
RTG Sub-Minute Polling

300s samples:

30s samples:
RTG Impulse Graph

- rtgplot can plot impulses (errors)
- rtgplot can plot gauges (temperature, CPU, etc)
- SONET Errors impulse graph showing errors leading to circuit failure (ES/SES):
Long-Term Trending

• Example of 3rd party scripts built on RTG foundation
• Perl scripts analyze data and produce CSV output that is easily imported into spreadsheets
• Ideal for management reports, trending, etc.
Thanks!

- Off-line Questions: Please See Me!
- Questions?

RTG Home: http://rtg.sf.net