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)

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

- 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

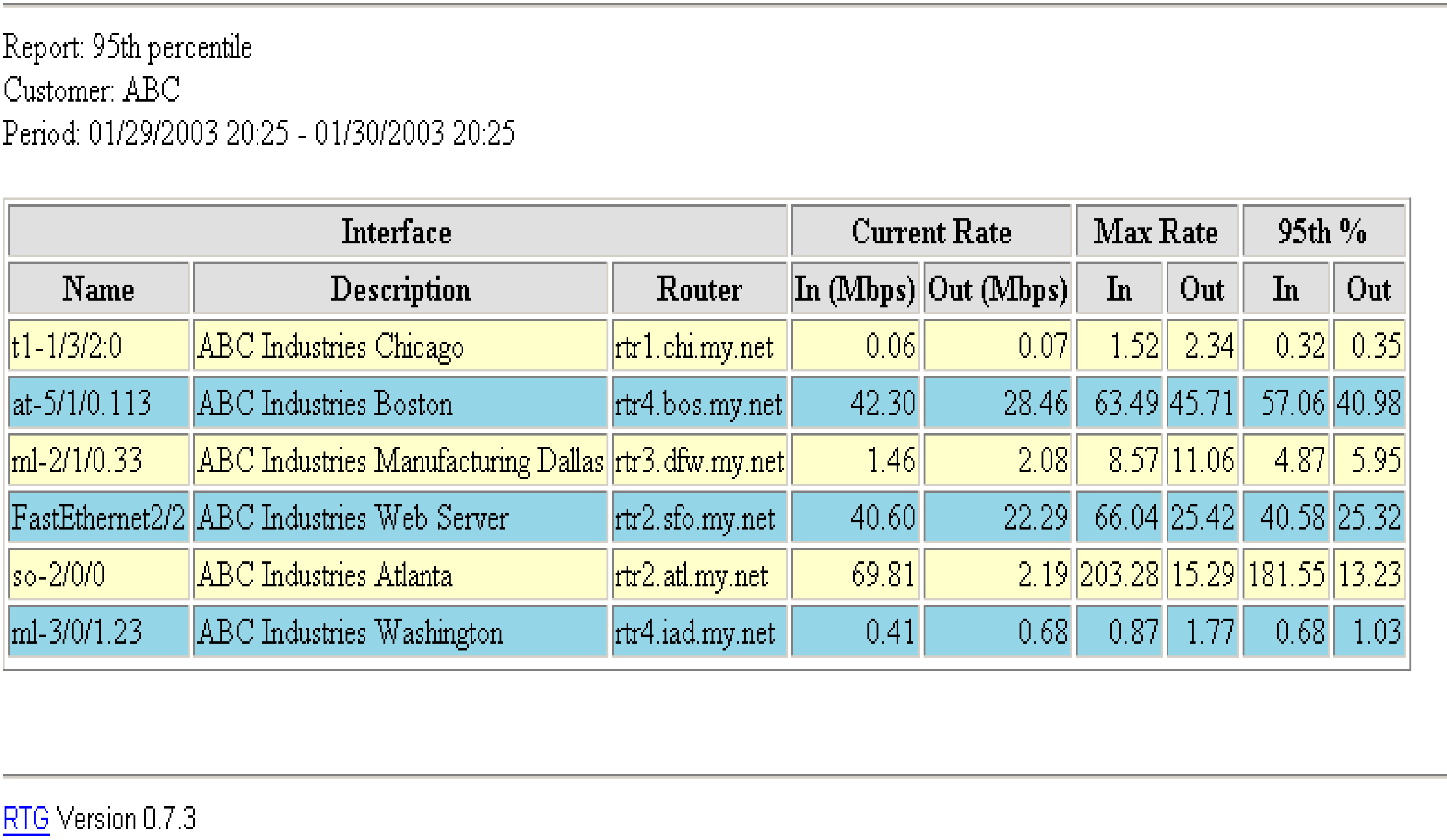
- Perl DBI scripts included
- Automate reporting, etc.
- Scripts to monitor thresholds, etc.

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

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

RTG Reports (95th Percentile)

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

                         RateIn   RateOut     MaxIn    MaxOut    95% In  95% Out
Connection                 Mbps      Mbps      Mbps      Mbps      Mbps     Mbps
--------------------------------------------------------------------------------
at-1/2/0.111 rtr-1.chi     0.09      0.07      0.65      0.22      0.22     0.13
at-1/2/0.113 rtr-1.dca     0.23      0.19      1.66      1.12      0.89     0.57
at-3/2/0.110 rtr-2.bos     0.11      0.16      0.34      0.56      0.26     0.40
```

RTG PHP 95th Percentile



Report: 95th percentile

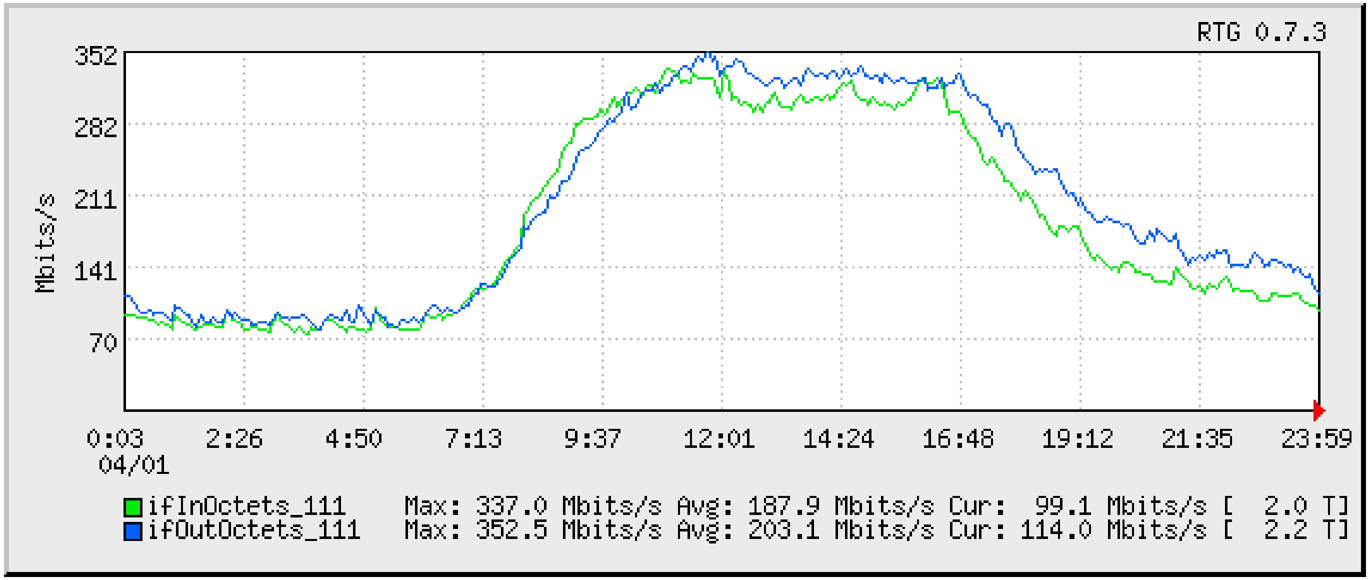
Customer: ABC

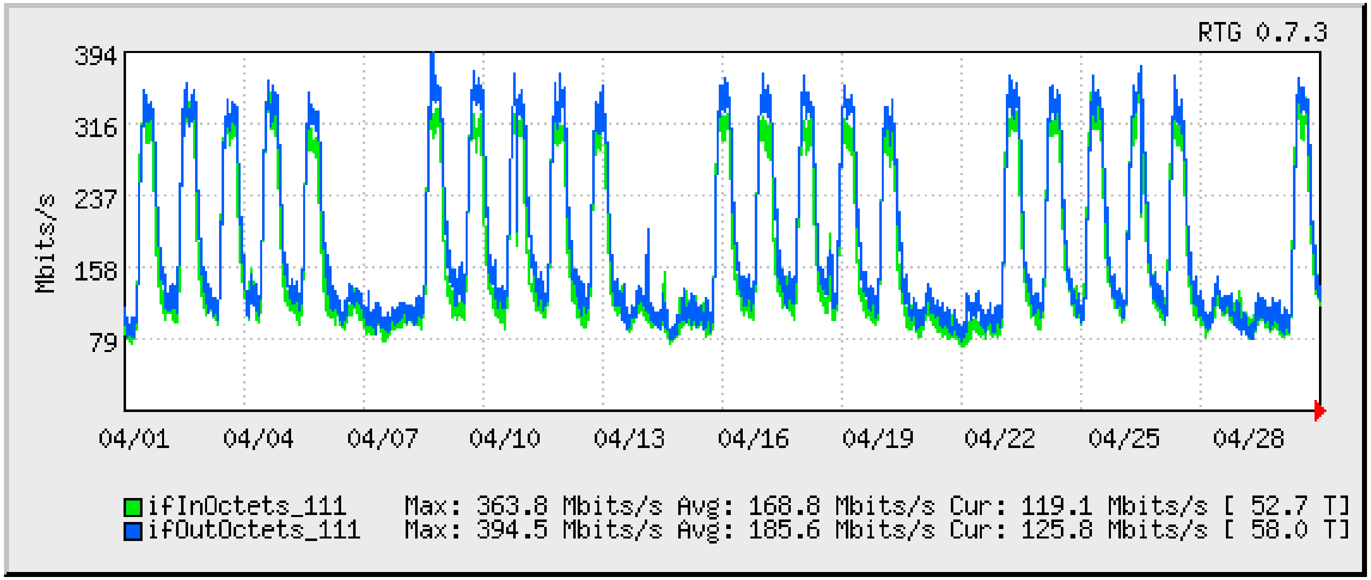
Period: 01/29/2003 20:25 - 01/30/2003 20:25

Interface			Current Rate		Max Rate		95th %	
Name	Description	Router	In (Mbps)	Out (Mbps)	In	Out	In	Out
t1-1/3/2:0	ABC Industries Chicago	rtr1.chi.my.net	0.06	0.07	1.52	2.34	0.32	0.35
at-5/1/0.113	ABC Industries Boston	rtr4.bos.my.net	42.30	28.46	63.49	45.71	57.06	40.98
ml-2/1/0.33	ABC Industries Manufacturing Dallas	rtr3.dfw.my.net	1.46	2.08	8.57	11.06	4.87	5.95
FastEthernet2/2	ABC Industries Web Server	rtr2.sfo.my.net	40.60	22.29	66.04	25.42	40.58	25.32
so-2/0/0	ABC Industries Atlanta	rtr2.atl.my.net	69.81	2.19	203.28	15.29	181.55	13.23
ml-3/0/1.23	ABC Industries Washington	rtr4.iad.my.net	0.41	0.68	0.87	1.77	0.68	1.03

RTG Version 0.7.3

RTG Traffic Graphs



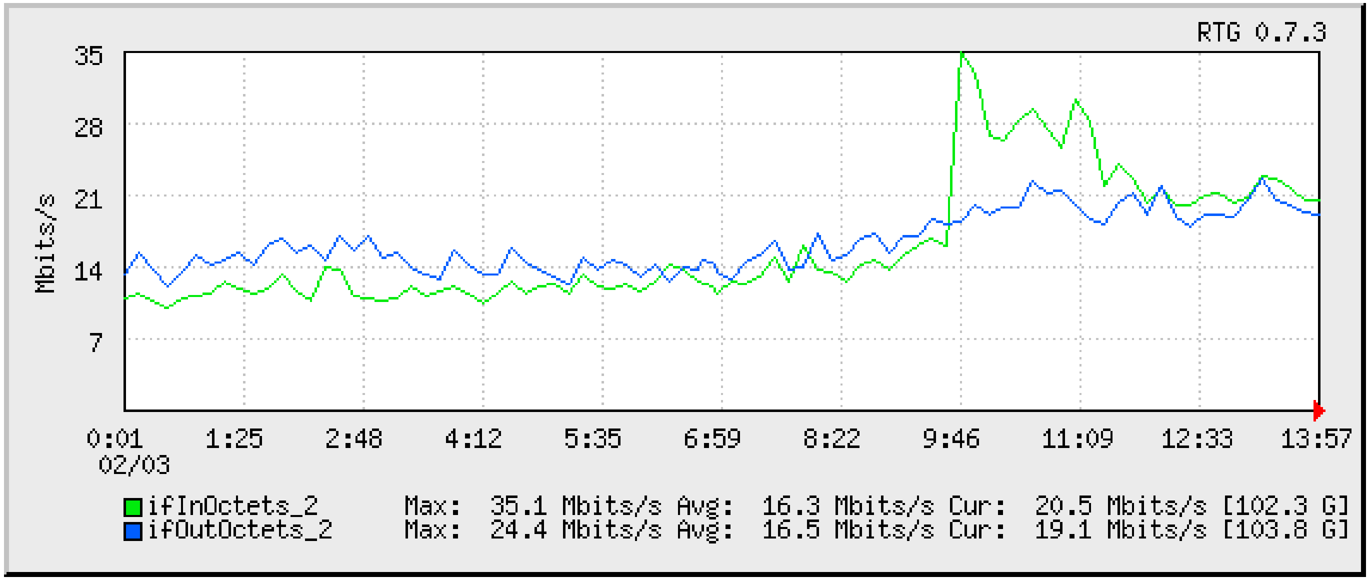


Sub-Minute Polling

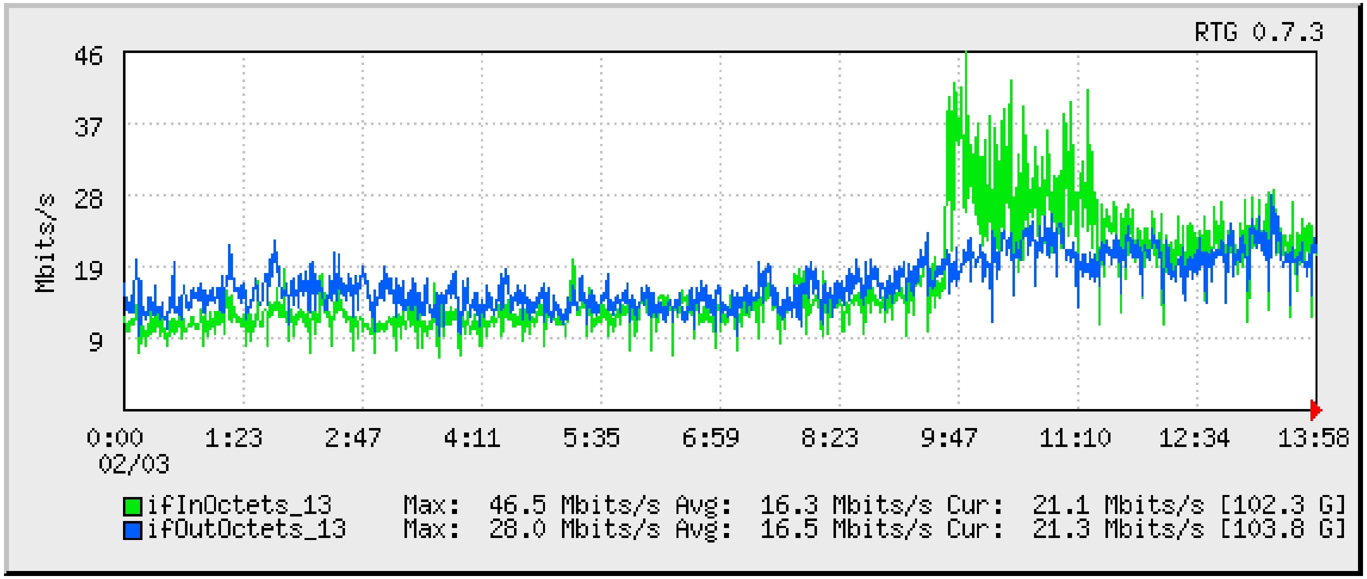
- RTG’s speed and architecture allow for sub-one minute polling
- Finer sampling granularity often reveals important traffic characteristics hidden in typical 5-minute aggregates
- Particularly useful as a diagnostic on slower links with bursty traffic
- Example (same interface and period, 30s vs 300s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

RTG Sub-Minute Polling

300s samples:

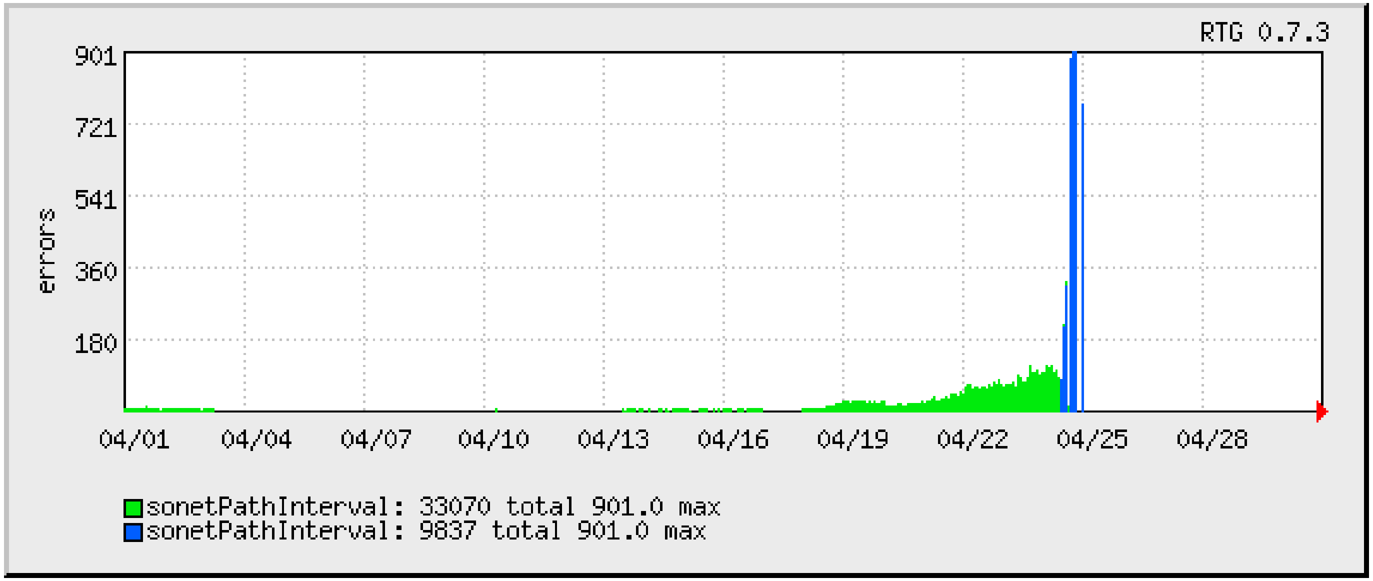


30s samples:



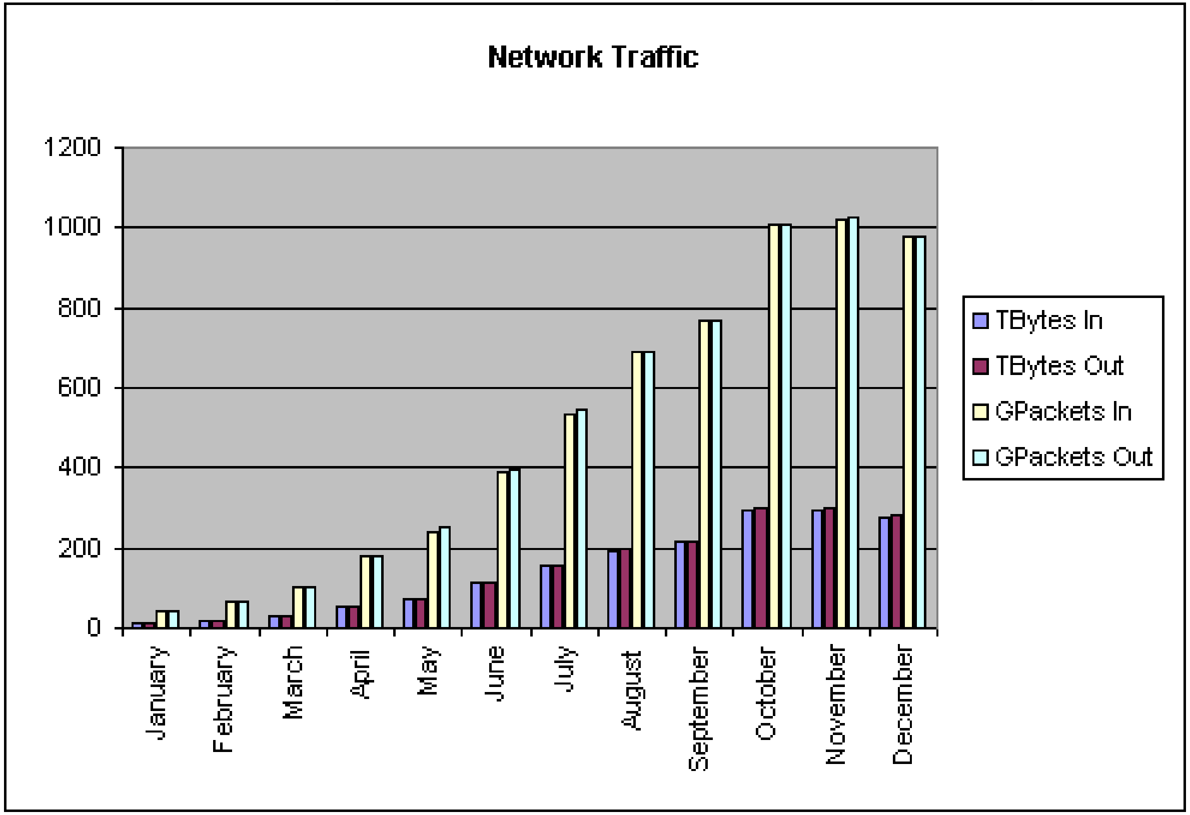
RTG Impulse Graph

- 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`