RRDtool as a Communication

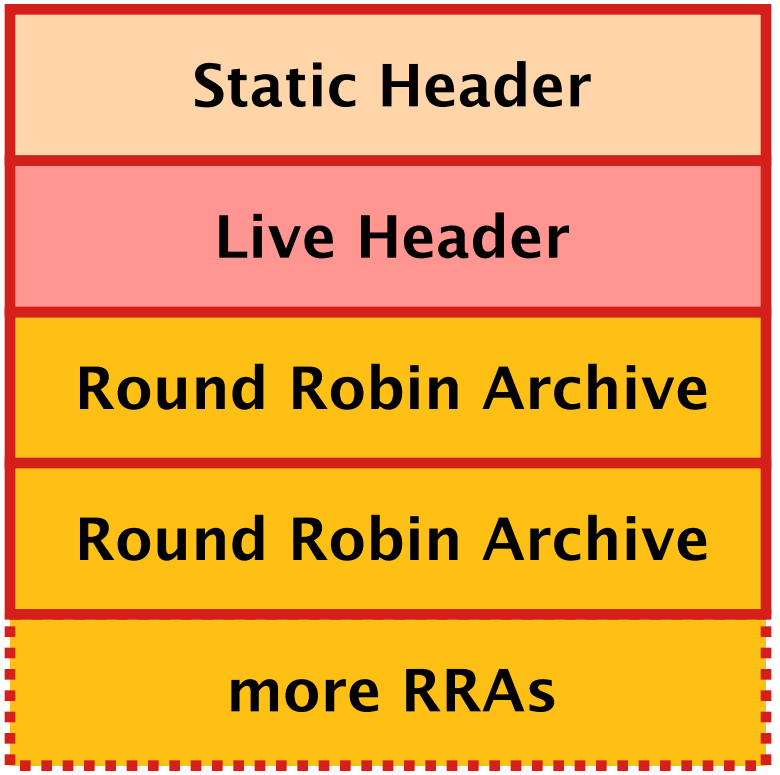
Tobi Oetiker <tobi@oetiker.ch>

How does that RRDtool Database work?

Tobi Oetiker <tobi@oetiker.ch>

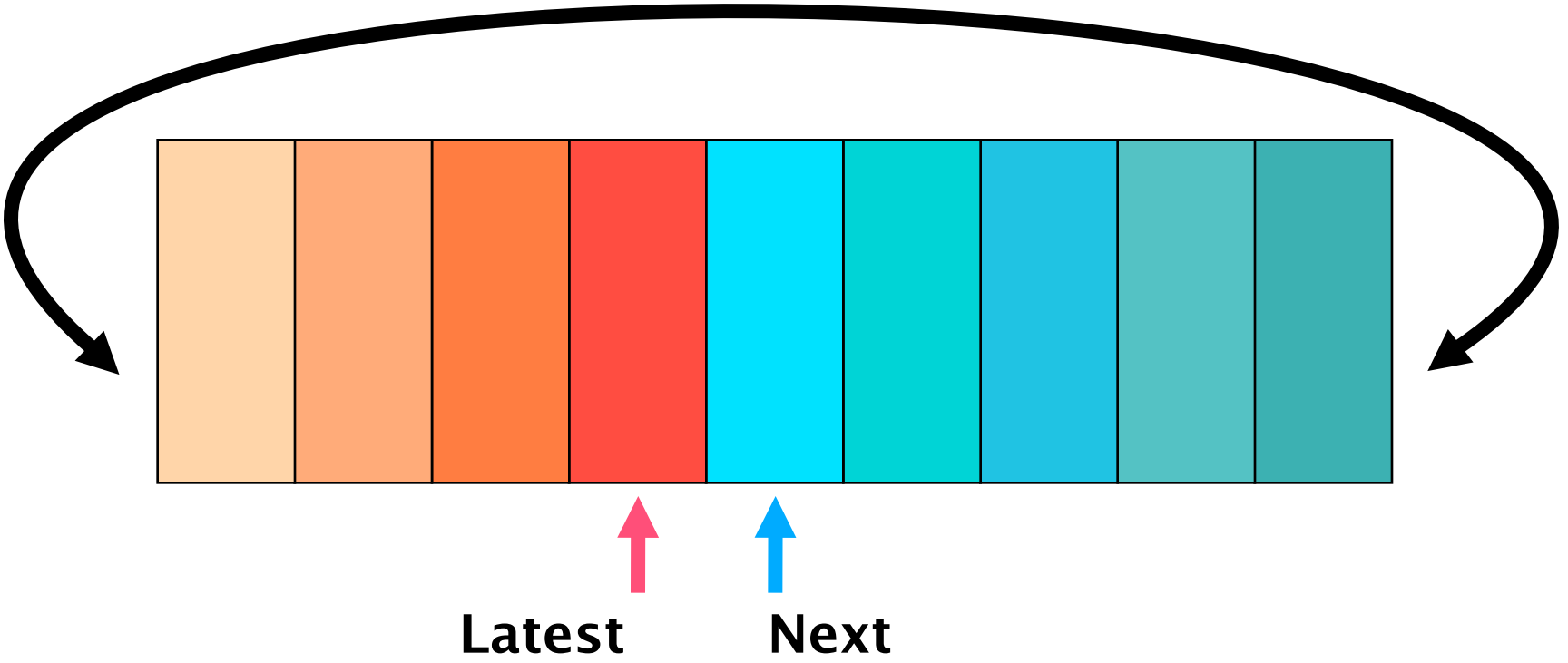
Components of an RRD file

- Our own Binary Data Format
- Data sources – DS
- Round Robin Archives – RRA



The Round Robin Principle

- fixed number of storage slots



What is a Data Source

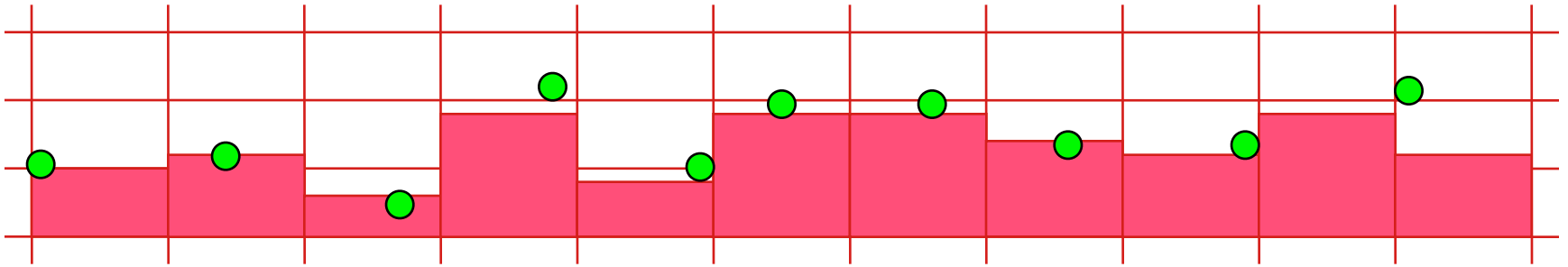
- “Anything with numbers”
- Log files
- SNMP counters
- /proc entries
- Output from an external program

Handling UNKNOWN Data

- Unknown is not 0!
- The Unknown is contagious
  1+Unknown=Unknown
- RRDtool handles the Unknown
- Configurable: How much Unknown shall be ignored.
- By default 50% is fine.

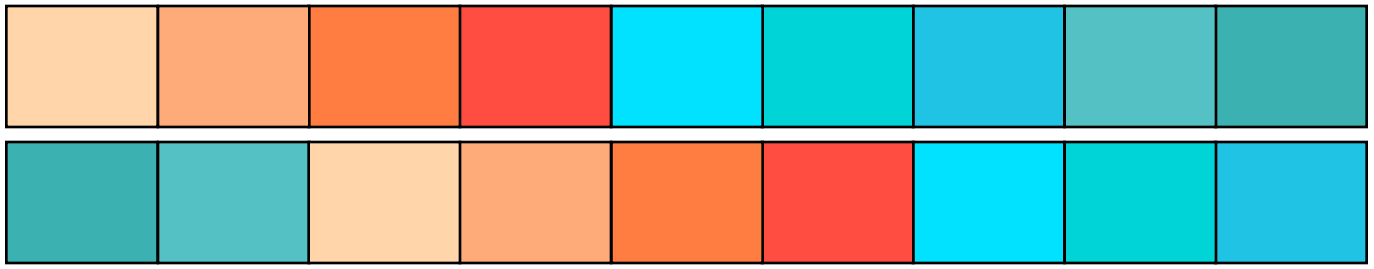
Remember this on DS

- RRDtool handles the UNKNOWN
- Everything is a rate.
  “real data” is not kept!
- Pick the right sampling interval.
  (Double the frequency)



Multiple Archives

- Keep data ready at the right resolutions
  - 5 Minute AVERAGE for 1 day
  - 1 hour MAX for a month



- Fit it to the questions you expect.

Working with Round Robin Database

Tobi Oetiker <tobi@oetiker.ch>

RRDtool create [1/3]

- man rrdcreate

```
rrdtool create \
  first.rrd \
  --step=300 \
  DS:speed:GAUGE:500:0:300 \
  RRA:AVERAGE:0.5:1:120 \
  RRA:AVERAGE:0.5:12:96
```

RRDtool create [2/3]

- **DS:***ds-name*:*GAUGE | COUNTER | DERIVE | ABSOLUTE*:*heartbeat*:*min*:*max*
- **RRA:***AVERAGE | MIN | MAX | LAST*:*xff*:*steps*:*rows*

- It is given as the ratio of allowed **UNKNOWN** PDPs to the number of PDPs in the interval. Thus, it ranges from 0 to 1 (exclusive). Ex. if set to 0.5, it means that if more than 50% points in the interval are unknown, the value is set to unknown.
- *steps* defines how many of these *primary data points* are used to build a *consolidated data point* which then goes into the archive.
- rows defines how many generations of data values are kept in an **RRA.**

RRDtool create [3/3]

```
rrdtool create temperature.rrd --step 300 \
DS:temp:GAUGE:600:-273:5000 \
RRA:AVERAGE:0.5:1:1200 \
RRA:MIN:0.5:12:2400 \
RRA:MAX:0.5:12:2400 \
RRA:AVERAGE:0.5:12:2400
```

- The first stores the temperatures supplied for 100 hours (1200 * 1 * 300 seconds = 100 hours).
- The second RRA stores the minimum temperature recorded over every hour (12 * 300 seconds = 1 hour), for 100 days (2400 hours).
- The third and the fourth RRA's do the same for the maximum and average temperature, respectively.

RRDtool update

- man rrdupdate

```
rrdtool update \
  first.rrd \
  --template=speed:distance \
  N:30:100
```

RRDtool info

- man rrdinfo

rrdtool info first.rrd

RRDtool tune

- man rrdtune

```
rrdtool tune first.rrd \
 --heartbeat=speed:600
```

RRDtool fetch

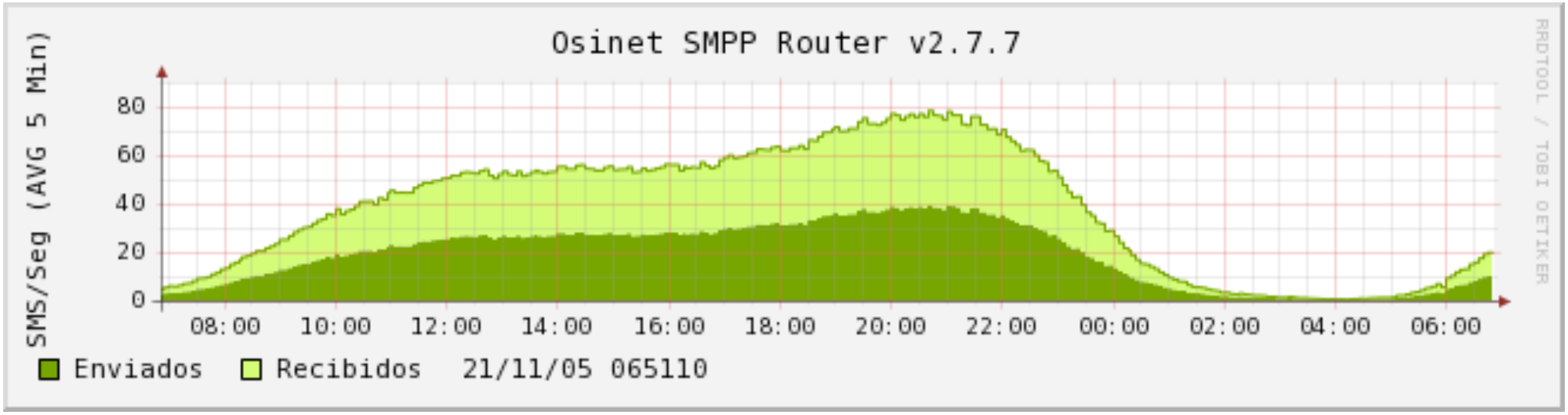
- man rrdfetch

rrdtool fetch first.rrd AVERAGE

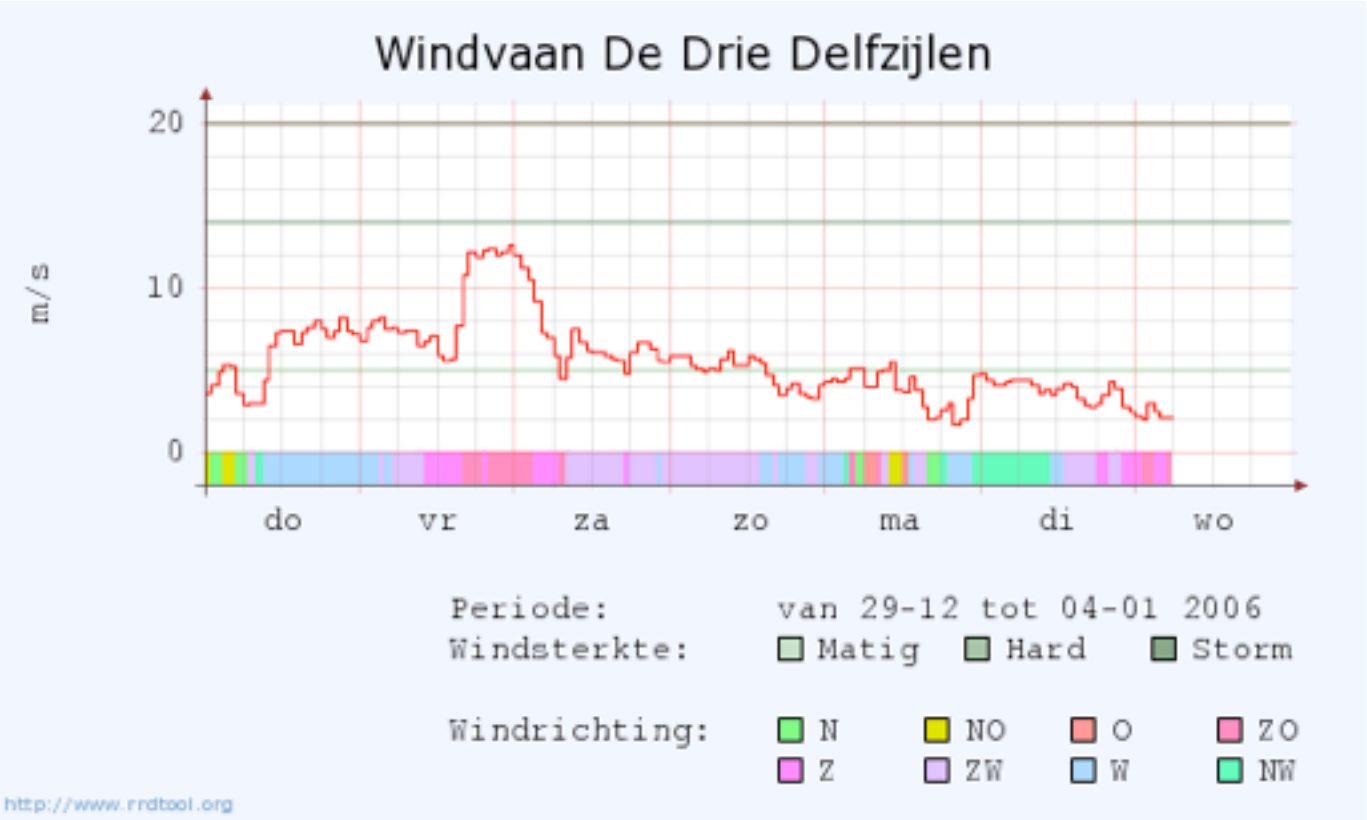
Graphing Basics

Tobi Oetiker <tobi@oetiker.ch>

Presenting the Results I



Presenting the Results II



RRDtool Graph work flow

- Get the data
- Run RPN calculations
- Put it on the graph

```
rrdtool graph my.png \
  DEF:in=first.rrd:speed:AVERAGE \
  CDEF:in8=in,8,* \
  LINE:in8#ff0000:'Bit Speed'
```

Getting the Data

- man rrdgraph_data

DEF:x=file.rrd:speed:MAX...

DEF...:step=3600

DEF...:start=end-1h

DEF...:end=11\:00

DEF...:reduce=AVERAGE

Calculating with RPN

- RPN = Reverse Polish Notation
- This is what HP Calculators used to do
- 1 [enter] 2 [enter] [+]
- no operator precedence
- simple programming language
- 1 [enter] 2 [enter] [keep larger]

RRDtool Graph RPN Basic

- man rrdgraph_rpn

- CDEF:bits=octets,8,*
- CDEF:avg=in,out,+,2,/
- CDEF:bigger=x,y,MAX
- CDEF:lim=a,0,100,LIMIT

RRDtool Graph RPN Adv

- man rrdgraph_rpn

- CDEF:bigger=x,y,LT,x,y,IF
- CDEF:unzero=x,UN,0,x,IF
- CDEF:avg2=v1,v2,v3,3,AVG
- CDEF:wind=x,1800,TREND
- CDEF:time=x,POP,TIME

CDEF Pitfalls

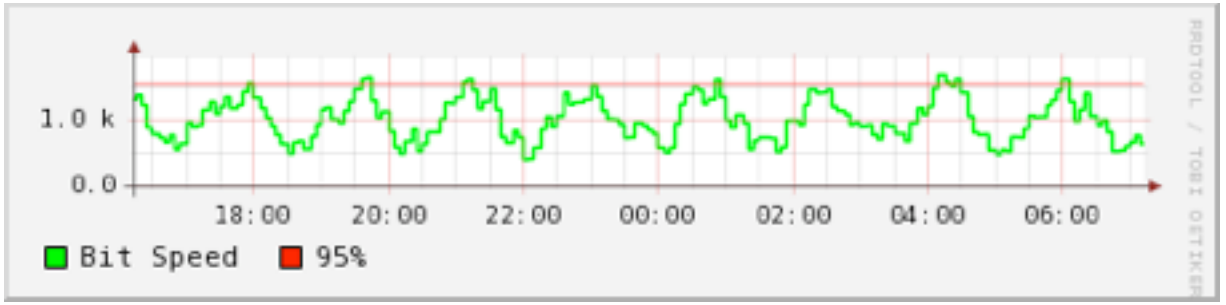
- there must be a DEF/CDEF variable in every CDEF the expression.
- CDEF:x=1,2,+ is INVALID!
- Trick:
  CDEF:x=y,POP,1,2,+

VDEF Expressions

- VDEF:var=data,95,PERCENT
- gives a single value!
- looks like CDEF but it's NOT
- result is usable in CDEF

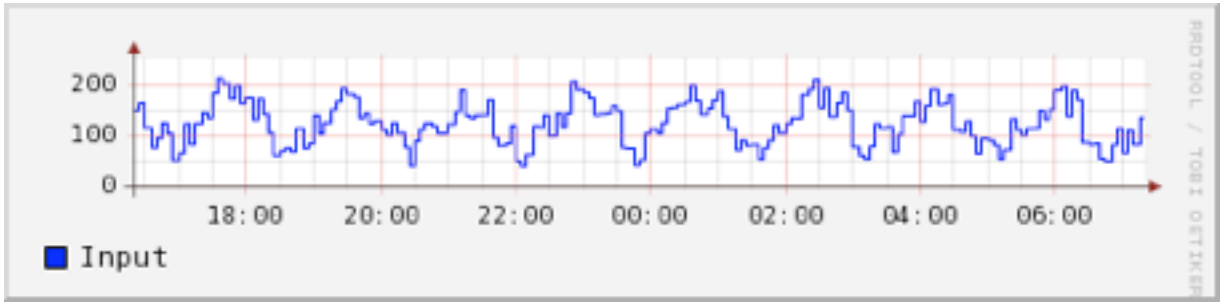
Drawing a “simple” graph

```
rrdtool graph my.png \
  DEF:in=first.rrd:speed:AVERAGE \
  CDEF:bits=in,8,* \
  VDEF:ninefive=bits,95,PERCENT \
  LINE2:in8#00ff00:’Bit Speed’ \
  LINE0.5:ninefive#ff0000:’95%’
```

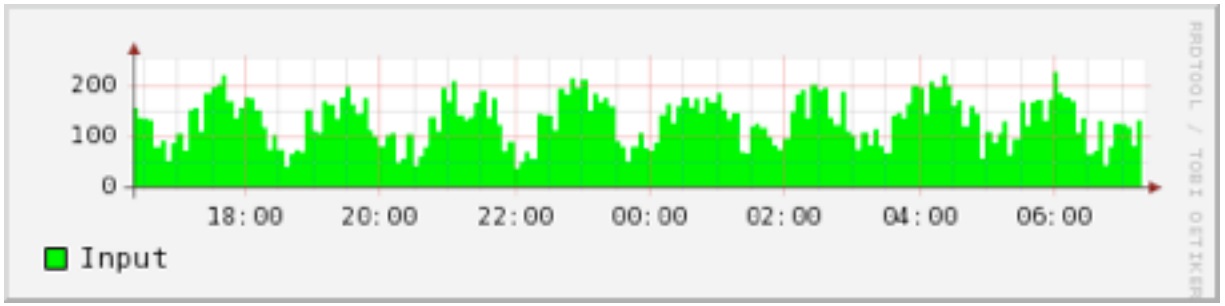


Drawing Elements

- LINE:input#0000ff:Input

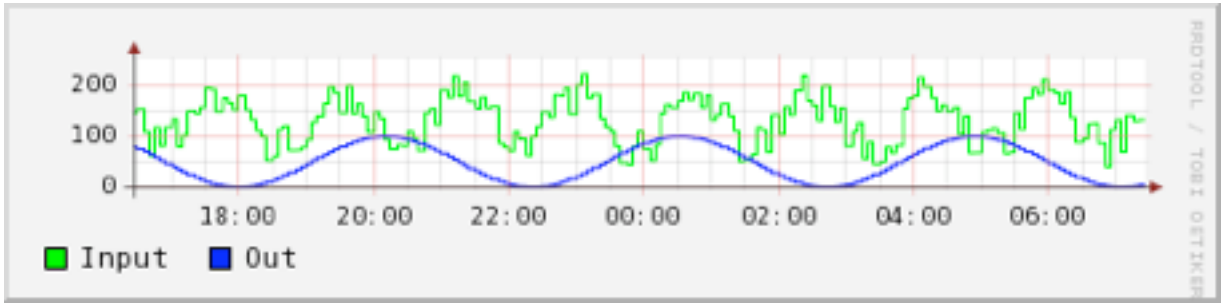


- AREA:input#00ff00:Input

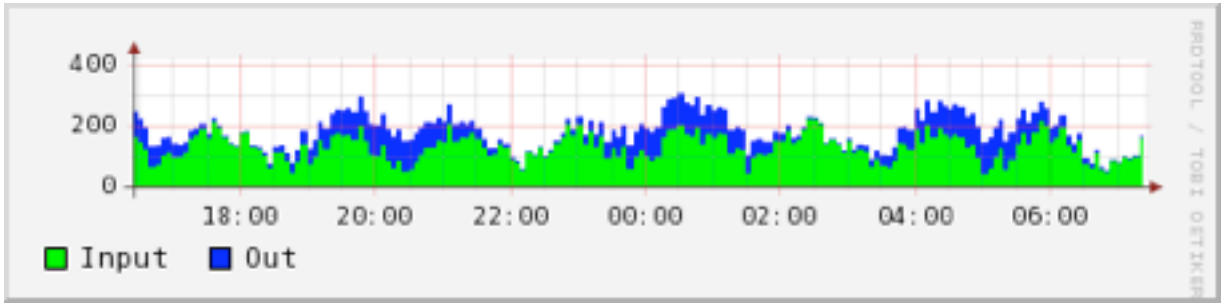


Graphing several Elements

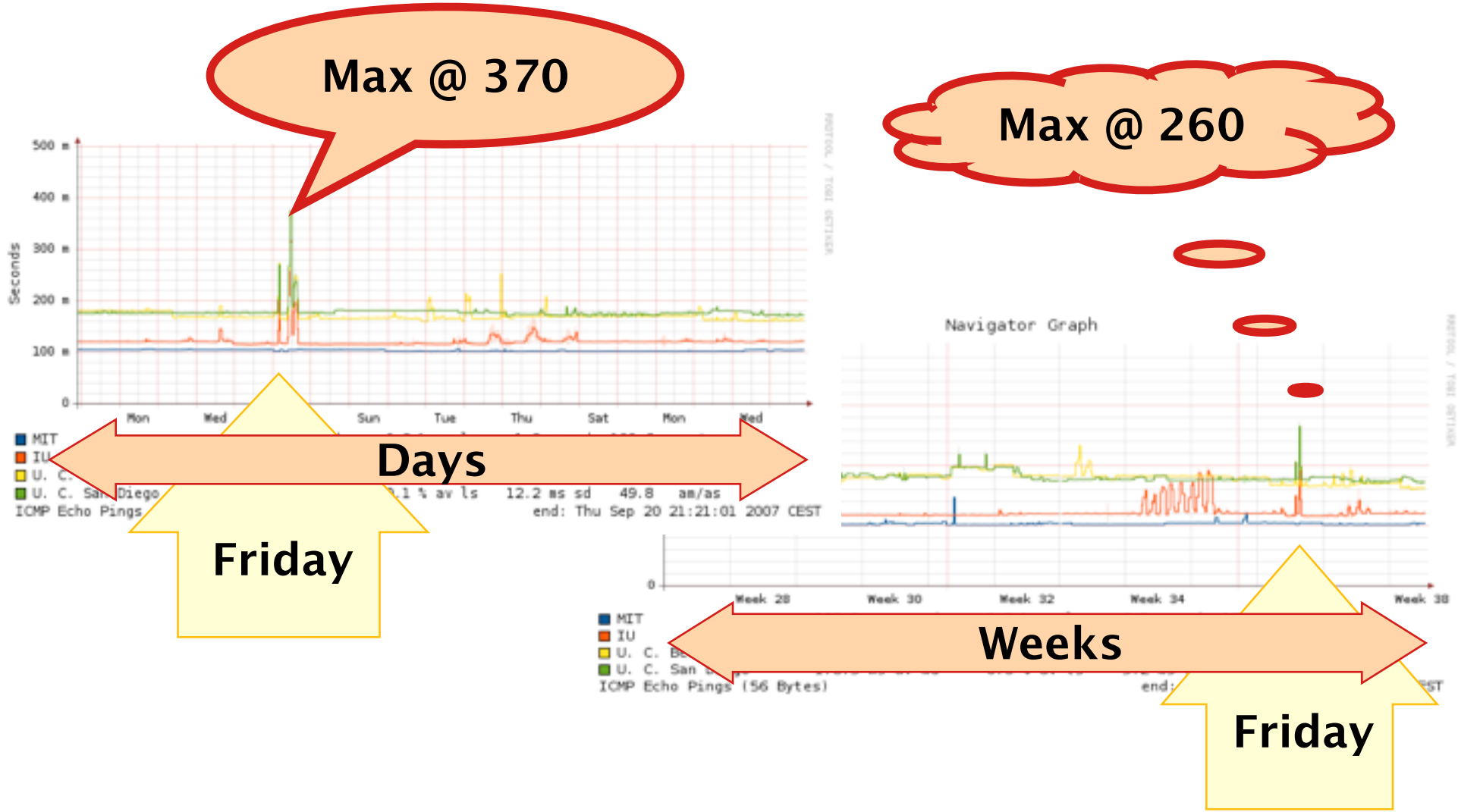
- LINE:... LINE:....



- AREA:... AREA:...:**STACK**



Where did Max go ?



Programming with RRDtool

Tobi Oetiker <tobi@oetiker.ch>

Programming with Shell

- Command line interface for shell scripting.
- Command line pipe interface
  - echo 'info my.rrd'| rrdtool -

Programming with Perl

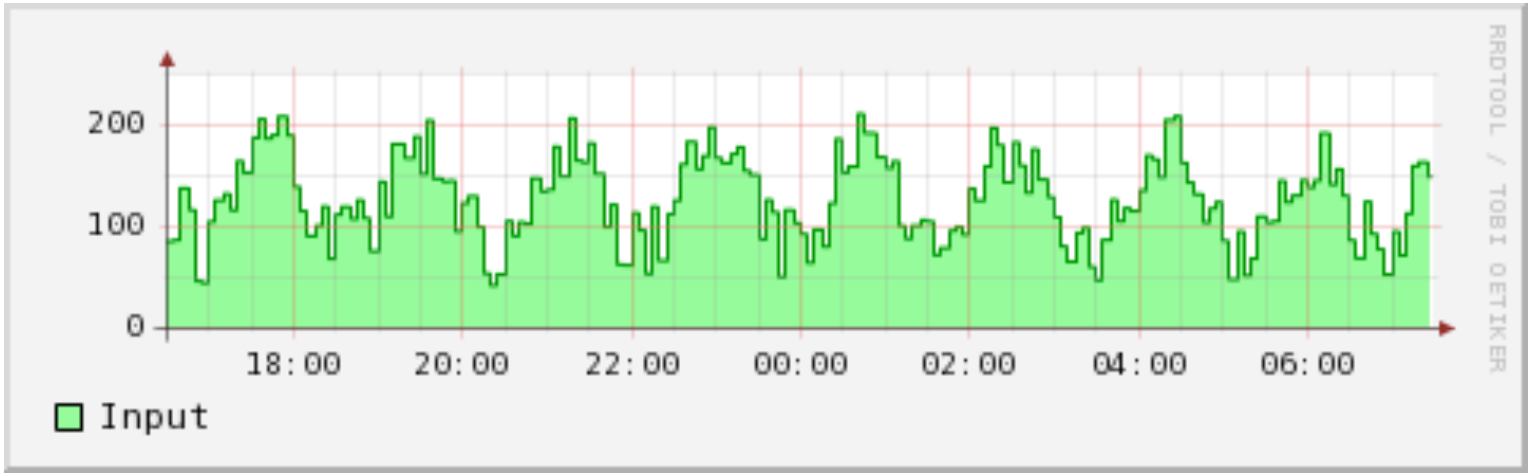
- By default bindings in rrdtool tree.
- use lib qw(/path/to/rrdtool/lib/perl);
  use RRDs;
- Similar for Python, TCL, Ruby

Programming to the C API

- Never intended
- Use rrd_tool.c as an example
- When multi-threading use thread safe functions if available.

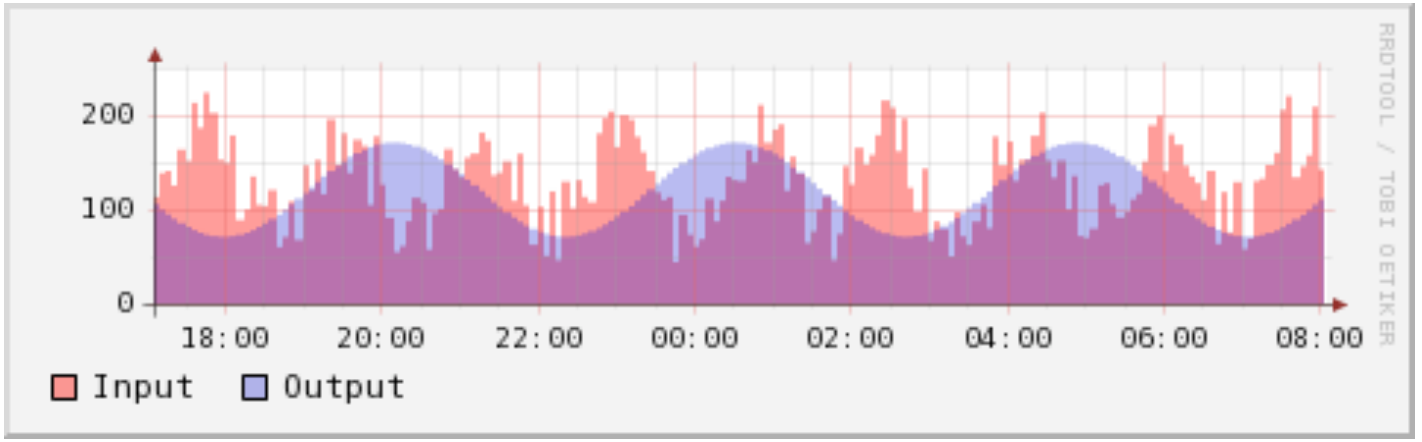
Effect: Outlining

```
rrdtool graph my.png \
  DEF:in=first.rrd:speed:AVERAGE \
  AREA:in#8f8:'Bit Speed' \
  LINE:in#080
```

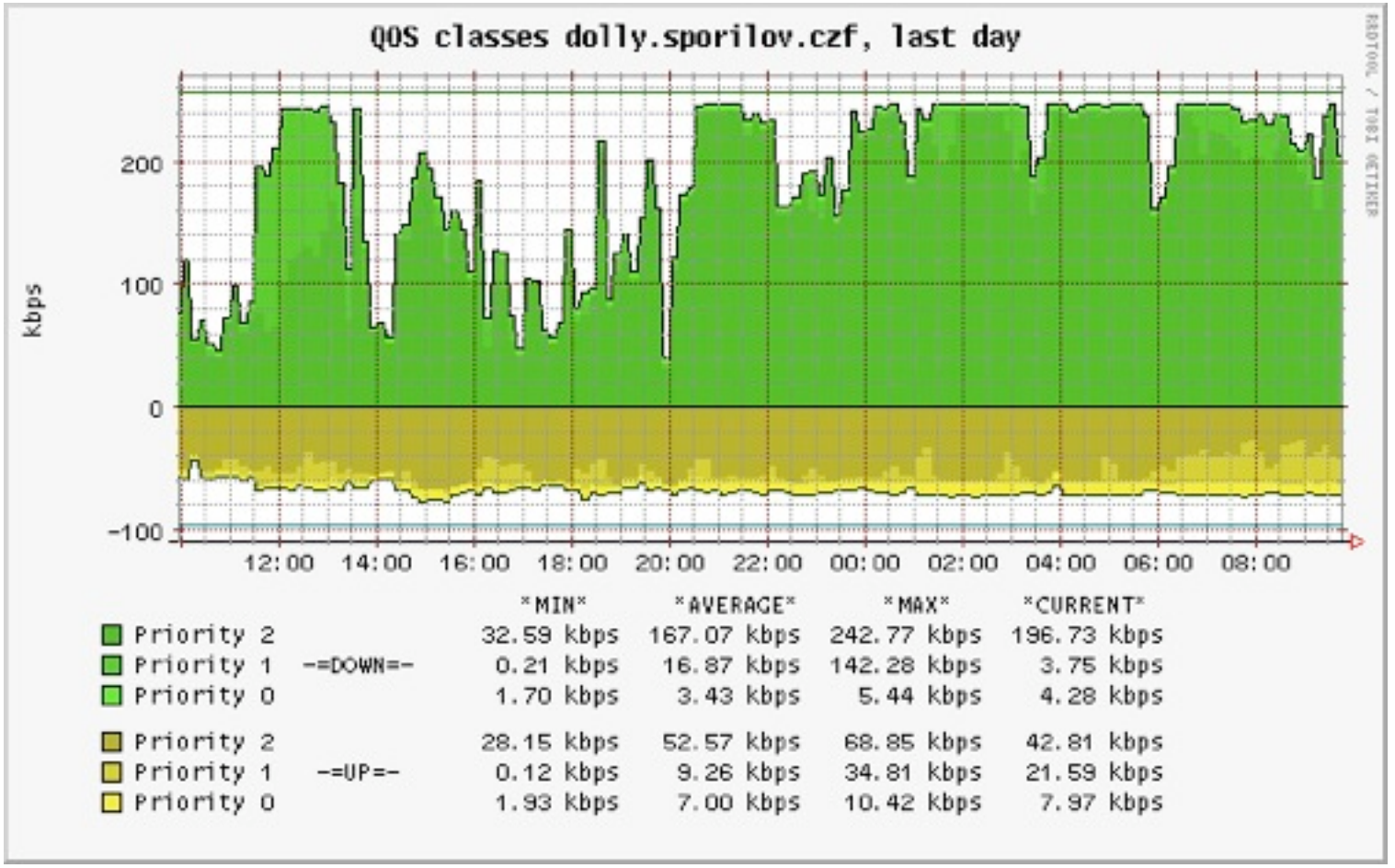


Effect: Transparency

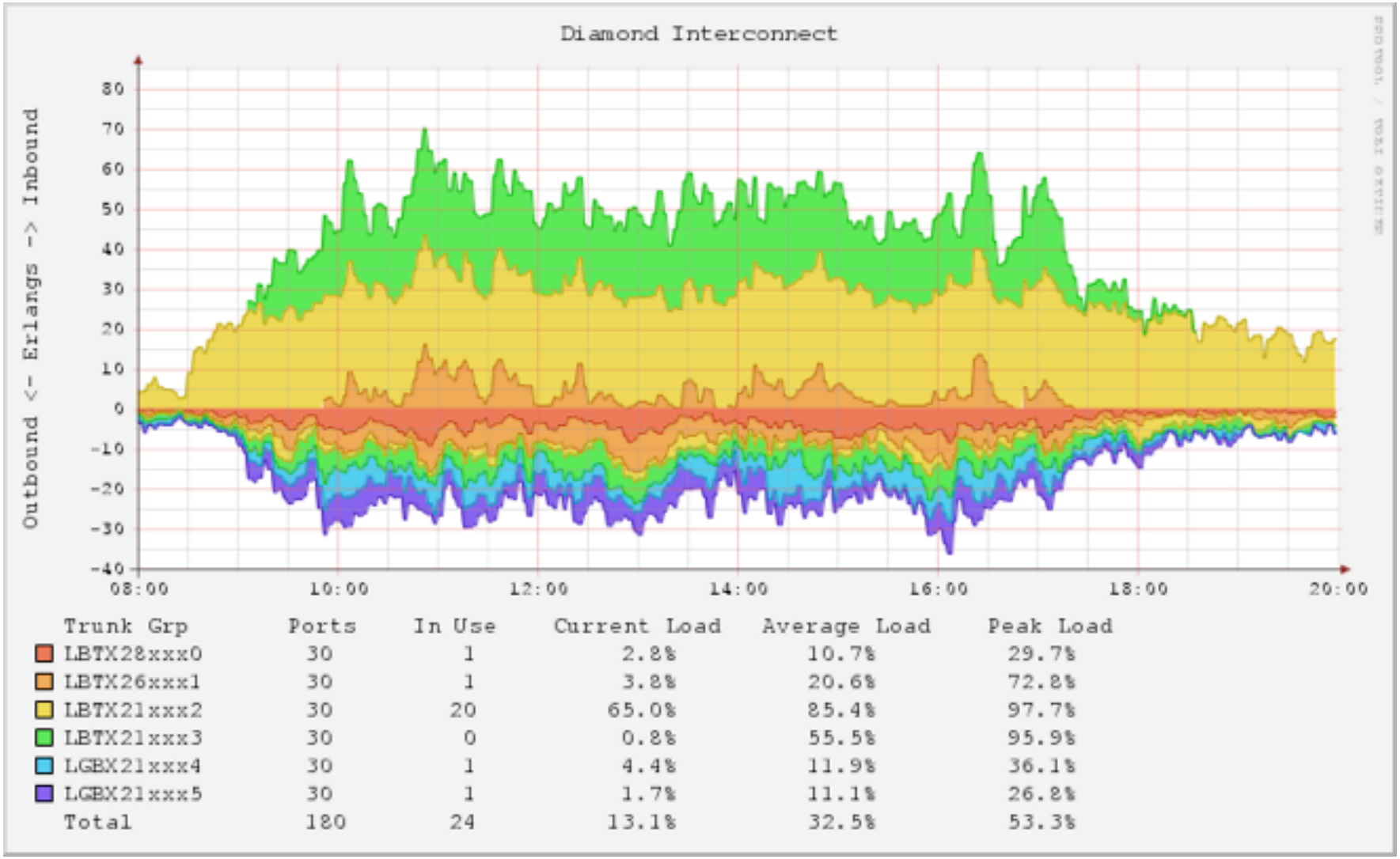
```
rrdtool graph my.png \
  DEF:in=first.rrd:in:AVERAGE \
  DEF:out=first.rrd:out:AVERAGE \
  AREA:in#f007:Input \
  AREA:out#0f05:Output
```



Gallery I


QOS classes dolly.sporilov.czf, last day
kbps
200
100
0
-100
12:00
14:00
16:00
18:00
20:00
22:00
00:00
02:00
04:00
06:00
08:00
MIN *AVERAGE* *MAX* *CURRENT*
Priority 2 32.59 kbps 167.07 kbps 242.77 kbps 196.73 kbps
Priority 1 -=DOWN=- 0.21 kbps 16.87 kbps 142.28 kbps 3.75 kbps
Priority 0 1.70 kbps 3.43 kbps 5.44 kbps 4.28 kbps
Priority 2 28.15 kbps 52.57 kbps 68.85 kbps 42.81 kbps
Priority 1 -=UP=- 0.12 kbps 9.26 kbps 34.81 kbps 21.59 kbps
Priority 0 1.93 kbps 7.00 kbps 10.42 kbps 7.97 kbps

Gallery II



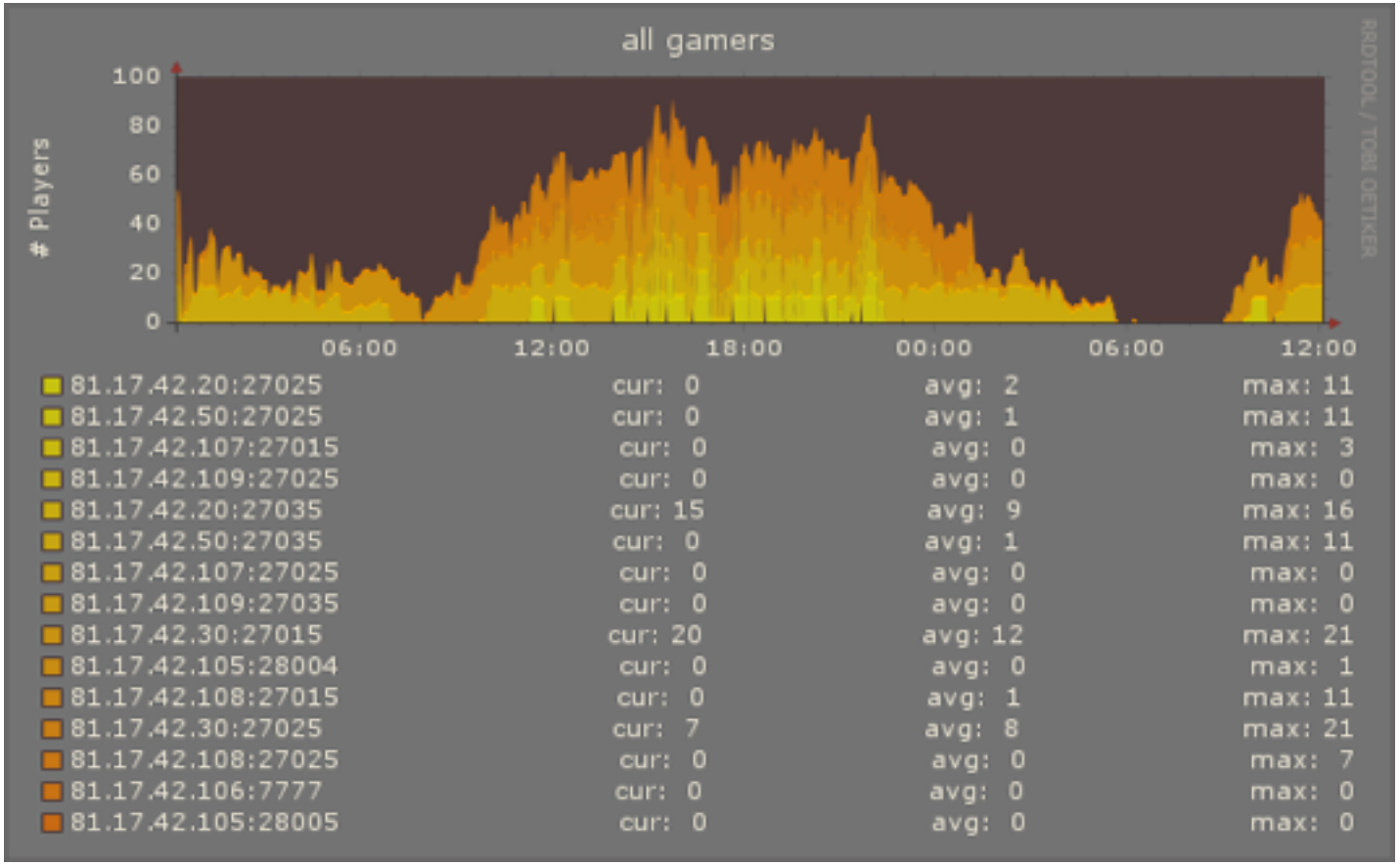
Trunk Grp	Ports	In Use	Current Load	Average Load	Peak Load
LBTX28xxx0	30	1	2.8%	10.7%	29.7%
LBTX26xxx1	30	1	3.8%	20.6%	72.8%
LBTX21xxx2	30	20	65.0%	85.4%	97.7%
LBTX21xxx3	30	0	0.8%	55.5%	95.9%
LGBX21xxx4	30	1	4.4%	11.9%	36.1%
LGBX21xxx5	30	1	1.7%	11.1%	26.8%
Total	180	24	13.1%	32.5%	53.3%

Graphing Pitfalls

- Design
- Content

Graphing Pitfalls

Design
Content



Adding Number to Graphs

- VDEF results can be printed
- Into the graph with GPRINT
- To the caller with PRINT
- Use sprintf formatting %lf

%lf	0.666666666667
%3.2lf	0.67

GPRINT and COMMENT

```
rrdtool graph my.png \
  DEF:in=first.rrd:speed:AVERAGE \
  LINE:in#f00 VDEF:avg=a,AVERAGE \
  GPRINT:avg:"Average %5.1lf" \
  COMMENT:"My Comment"
```