CSE 265: System and Network Administration

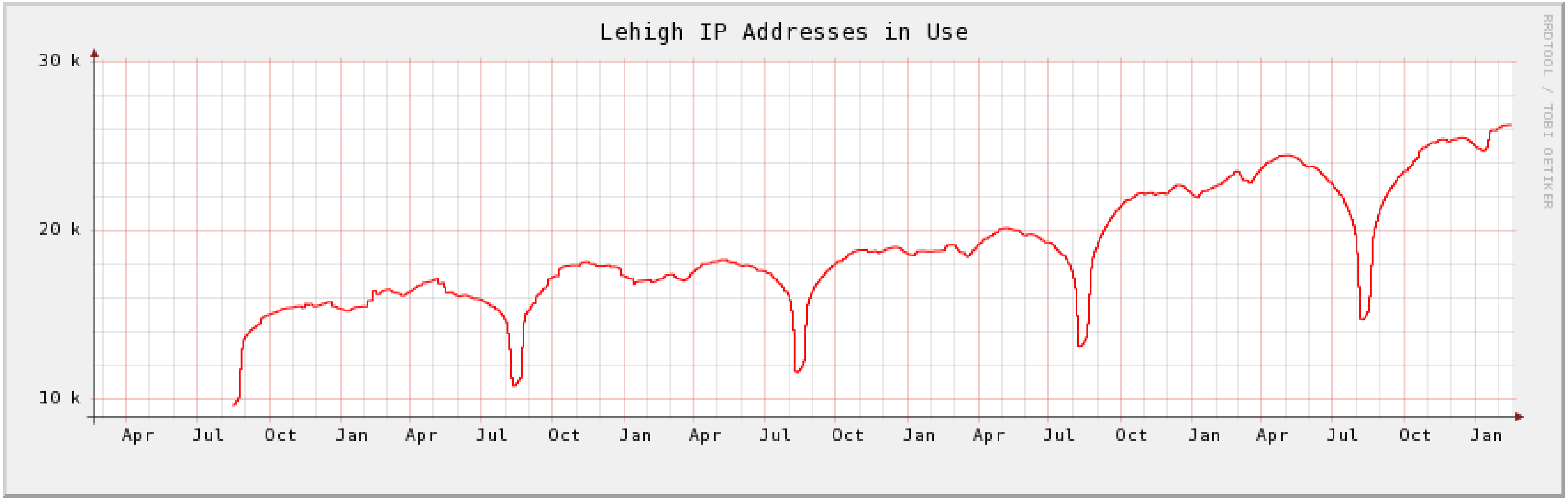
If you aren't measuring it, you aren't managing it.

- Service Monitoring
  - Historical data
  - Real-time monitoring
    - Alerting
    - Active monitoring systems
  - End-to-end tests
  - Application response time monitoring
- Syslog and Log files
  - Logging policies – what do you do with logs?
  - Linux log files
  - Logrotate: Manage log files
  - Syslog: system event logger
  - Condensing log files

Motivation for monitoring

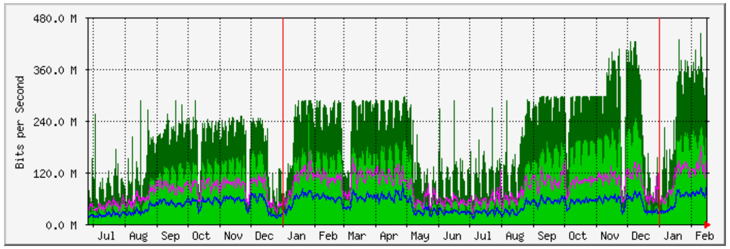
Motivation for monitoring

- Rapidly detect and fix problems
- Identify the source of problems
- Predict and avoid future problems
- Provide data on SA's achievements



Historical data

- Historical availability
  - record long-term uptime statistics
  - show improvements (99.99% uptime vs 99.9%)
- Utilization data useful for capacity planning
- Process
  - Poll systems at regular intervals
  - Collected, often graphed
  - Example: network status
    - https://ss.cc.lehigh.edu/public/mrtg/internett.html

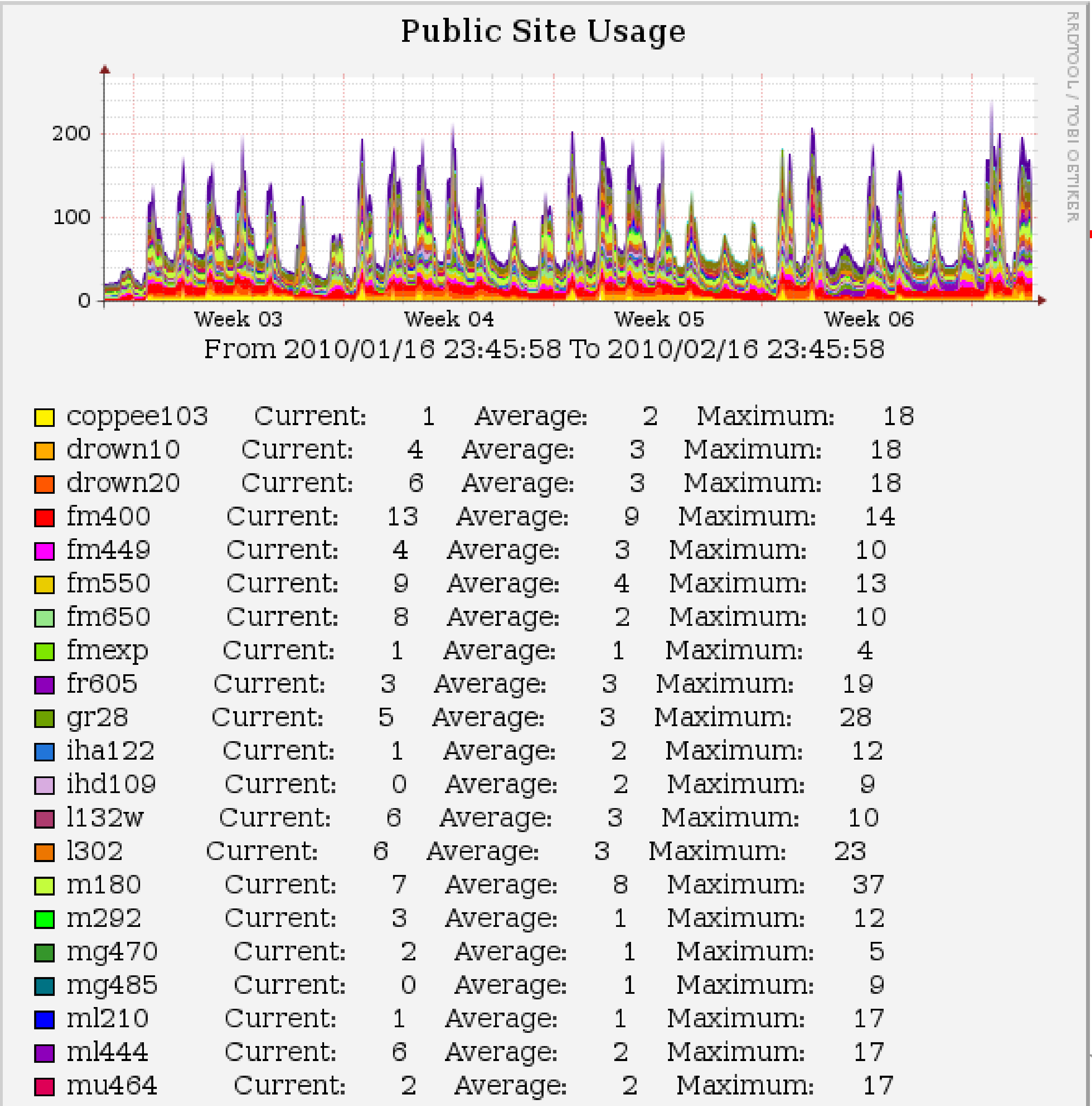


Real-time monitoring

- Alert SA immediately about a failure
- Want to notice outage before customer does
  - maintain reputation
  - minimize downtime
- Two components
  - Monitoring (polling) systems to check status, watching error messages, checking subsystems
  - Alerting – recognize problems and notify SAs

Monitoring

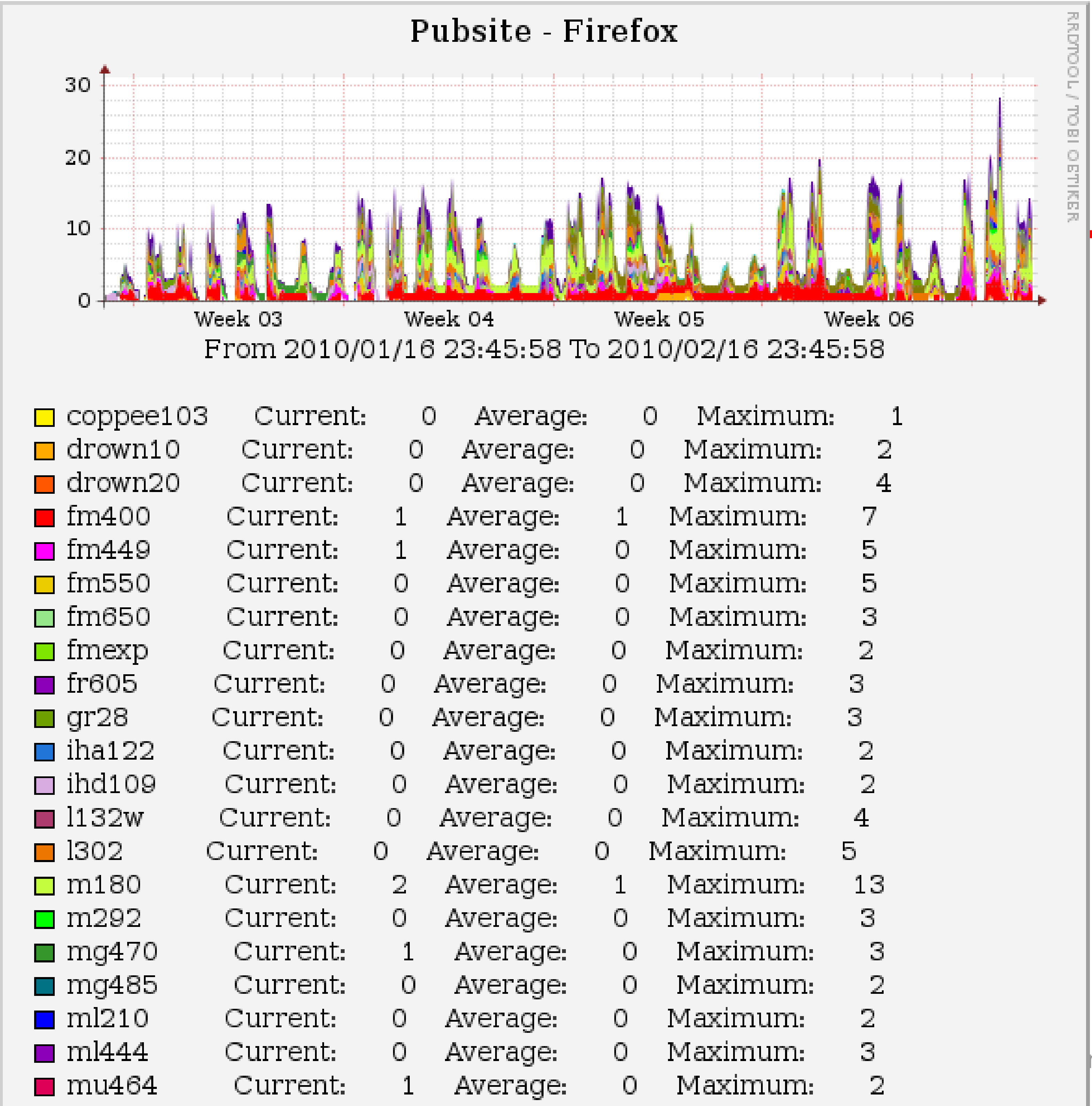
- Want to monitor everything that can indicate a problem
- Availability monitoring
  - Host/network/application failures
- Capacity monitoring
  - Approaching or past overload



Public Site Usage

From 2010/01/16 23:45:58 To 2010/02/16 23:45:58

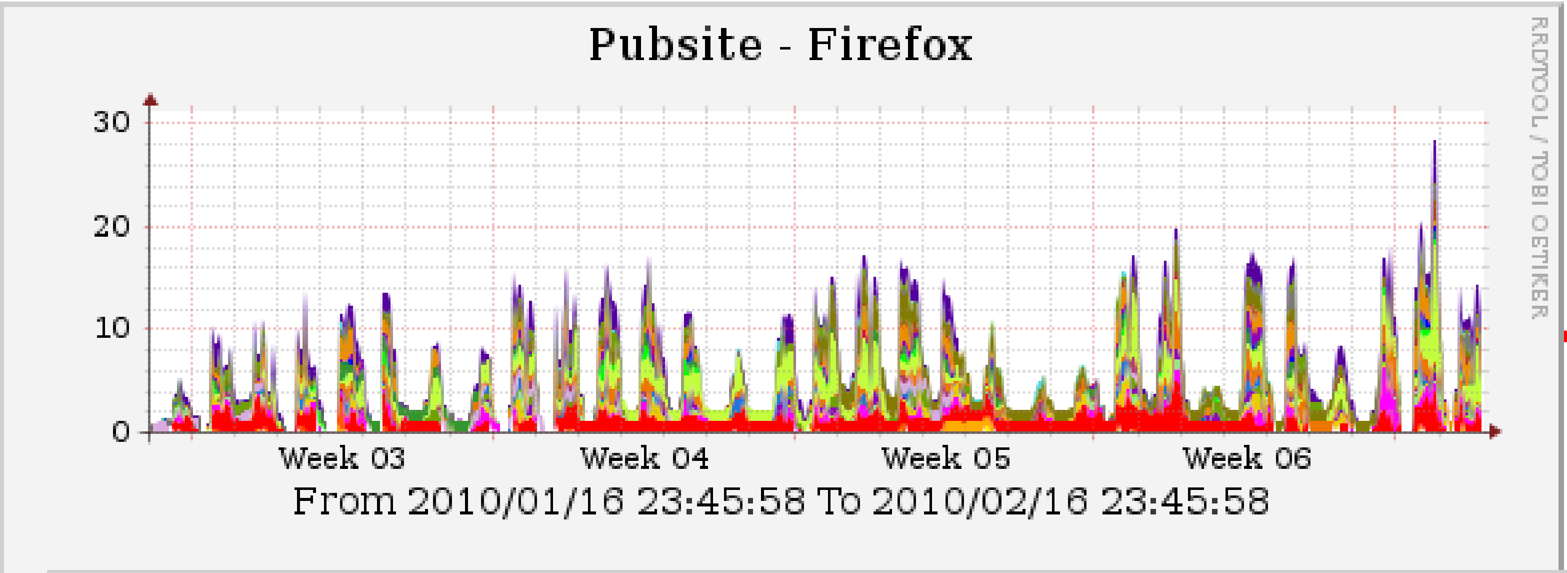
🟨	coppee103	Current:	1	Average:	2	Maximum:	18
🟧	drown10	Current:	4	Average:	3	Maximum:	18
🟧	drown20	Current:	6	Average:	3	Maximum:	18
🟥	fm400	Current:	13	Average:	9	Maximum:	14
🟪	fm449	Current:	4	Average:	3	Maximum:	10
🟨	fm550	Current:	9	Average:	4	Maximum:	13
🟩	fm650	Current:	8	Average:	2	Maximum:	10
🟩	fmexp	Current:	1	Average:	1	Maximum:	4
🟪	fr605	Current:	3	Average:	3	Maximum:	19
🟩	gr28	Current:	5	Average:	3	Maximum:	28
🟦	iha122	Current:	1	Average:	2	Maximum:	12
🟪	ihd109	Current:	0	Average:	2	Maximum:	9
🟥	l132w	Current:	6	Average:	3	Maximum:	10
🟧	l302	Current:	6	Average:	3	Maximum:	23
🟩	m180	Current:	7	Average:	8	Maximum:	37
🟩	m292	Current:	3	Average:	1	Maximum:	12
🟩	mg470	Current:	2	Average:	1	Maximum:	5
🟦	mg485	Current:	0	Average:	1	Maximum:	9
🟦	ml210	Current:	1	Average:	1	Maximum:	17
🟪	ml444	Current:	6	Average:	2	Maximum:	17
🟥	mu464	Current:	2	Average:	2	Maximum:	17



Pubsite - Firefox

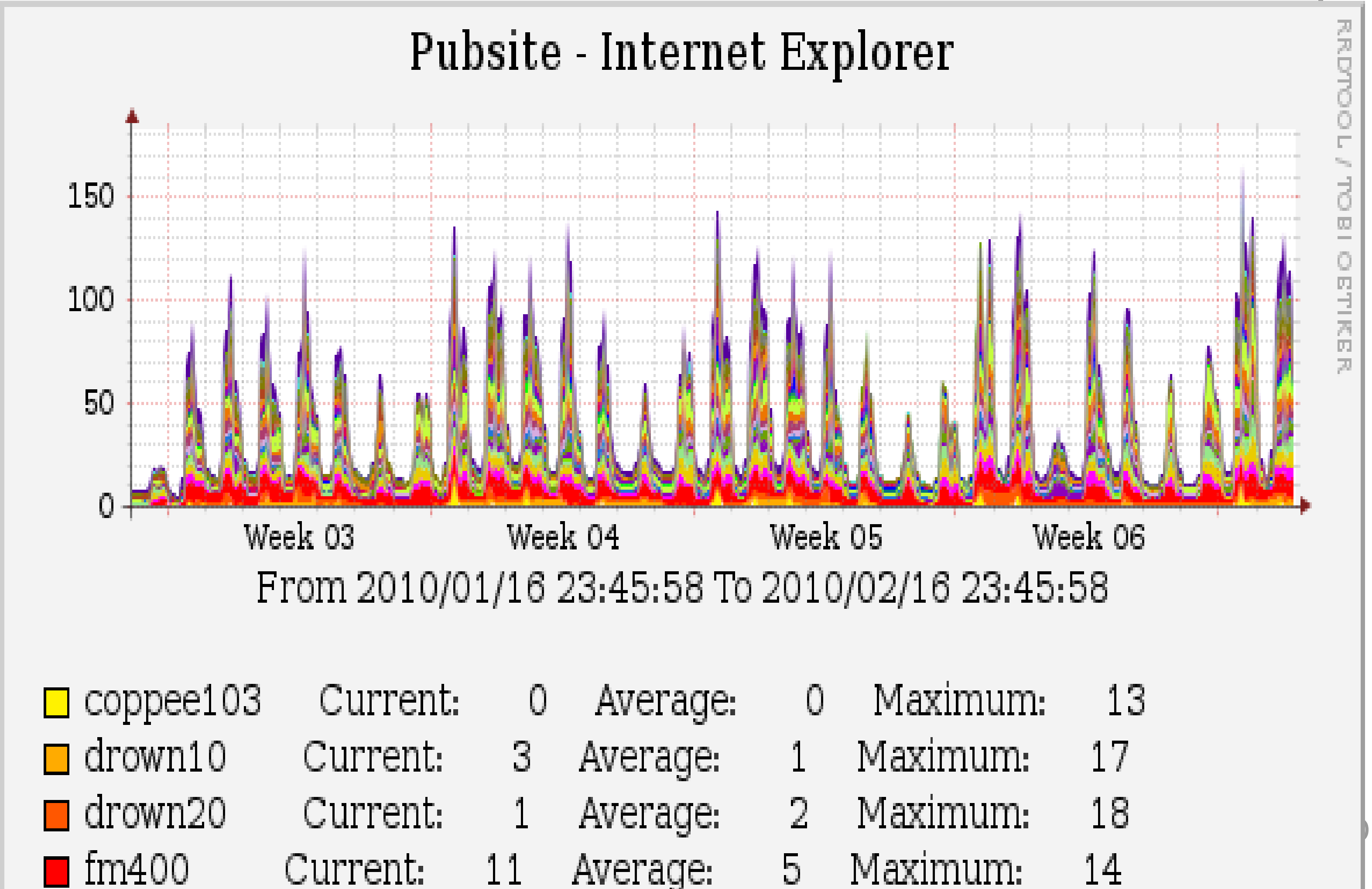
From 2010/01/16 23:45:58 To 2010/02/16 23:45:58

🟨	coppee103	Current:	0	Average:	0	Maximum:	1
🟧	drown10	Current:	0	Average:	0	Maximum:	2
🟧	drown20	Current:	0	Average:	0	Maximum:	4
🟥	fm400	Current:	1	Average:	1	Maximum:	7
🟪	fm449	Current:	1	Average:	0	Maximum:	5
🟨	fm550	Current:	0	Average:	0	Maximum:	5
🟩	fm650	Current:	0	Average:	0	Maximum:	3
🟩	fmexp	Current:	0	Average:	0	Maximum:	2
🟪	fr605	Current:	0	Average:	0	Maximum:	3
🟩	gr28	Current:	0	Average:	0	Maximum:	3
🟦	iha122	Current:	0	Average:	0	Maximum:	2
🟪	ihd109	Current:	0	Average:	0	Maximum:	2
🟥	l132w	Current:	0	Average:	0	Maximum:	4
🟧	l302	Current:	0	Average:	0	Maximum:	5
🟩	m180	Current:	2	Average:	1	Maximum:	13
🟩	m292	Current:	0	Average:	0	Maximum:	3
🟩	mg470	Current:	1	Average:	0	Maximum:	3
🟦	mg485	Current:	0	Average:	0	Maximum:	2
🟦	ml210	Current:	0	Average:	0	Maximum:	2
🟪	ml444	Current:	0	Average:	0	Maximum:	3
🟥	mu464	Current:	1	Average:	0	Maximum:	2



Pubsite - Firefox

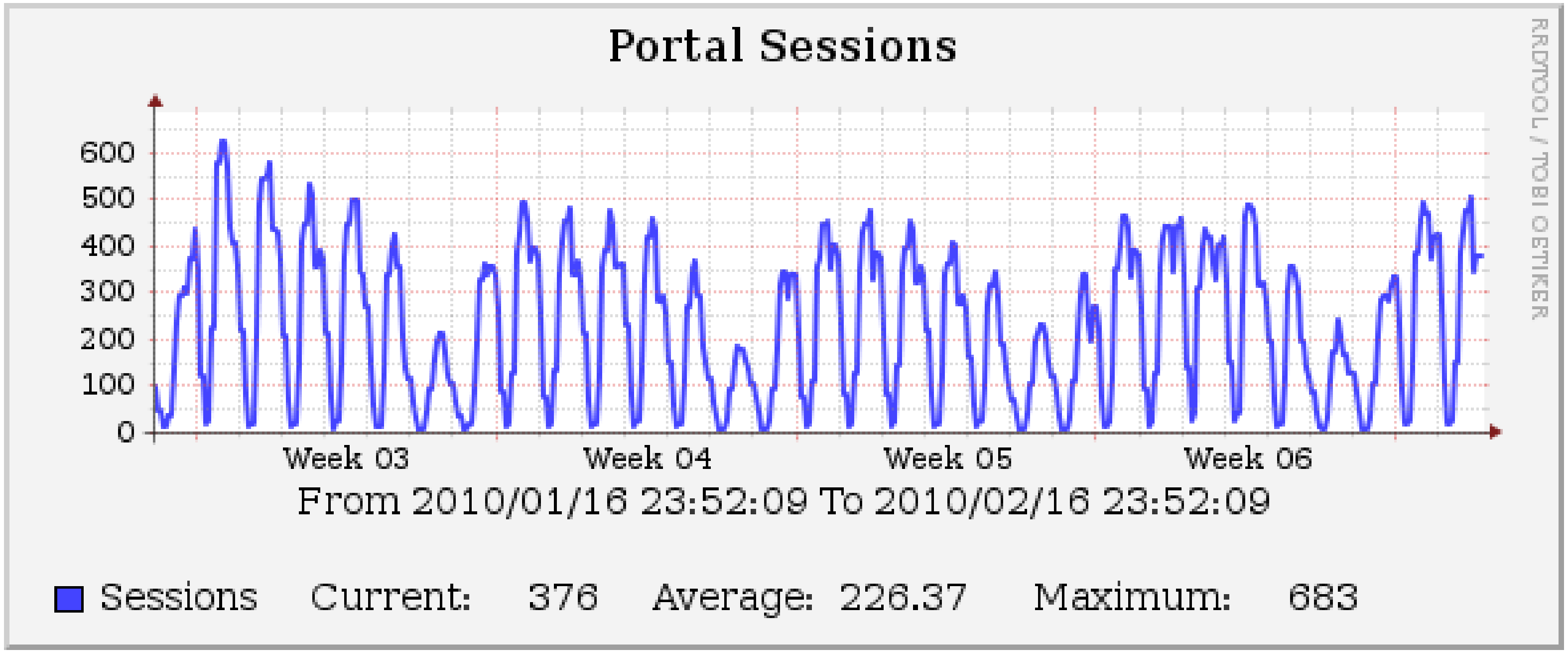
From 2010/01/16 23:45:58 To 2010/02/16 23:45:58



Pubsite - Internet Explorer

From 2010/01/16 23:45:58 To 2010/02/16 23:45:58

🟨 coppee103	Current:	0	Average:	0	Maximum:	13
🟧 drown10	Current:	3	Average:	1	Maximum:	17
🟧 drown20	Current:	1	Average:	2	Maximum:	18
🟥 fm400	Current:	11	Average:	5	Maximum:	14


Portal Sessions
600
500
400
300
200
100
0
Week 03
Week 04
Week 05
Week 06
From 2010/01/16 23:52:09 To 2010/02/16 23:52:09
■ Sessions Current: 376 Average: 226.37 Maximum: 683
RRDTOOL / TOBI OETIKER

Alerting

- Monitoring useless without alerting system
- Should not depend on system being monitored
  - e.g., don't depend on e-mail if network is down
- Who gets alerts? What if failure persists?
- Need to test alerting system
  - “I'm hot! I'm wet!”

Active monitoring

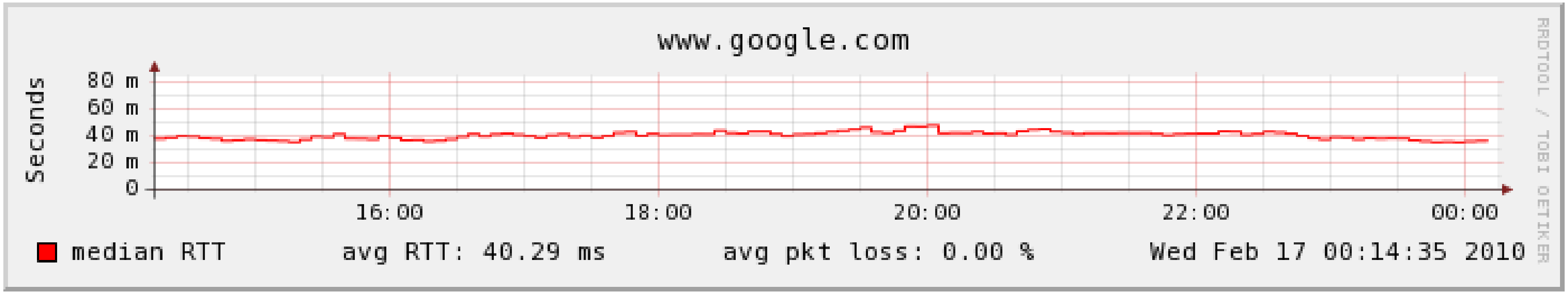
- Don't just monitor and alert, do something!
- Respond quickly/automatically
- Temporary solutions
  - Still need a permanent fix
- Can be a security risk (often requires privileges)

End-to-end tests

- Test entire transactions as a simulated customer
  - Send email through a server
  - Log in, select an item, check-out, get receipt
- Find problems before customers
- Find systemic problems, even when individual components are working

Application response time monitoring

- Even when everything works, if it is too slow, it is a failure
  - Loss of productivity
  - Loss of sales
  - Resentment
- Use historical monitoring, too



Summary

- Two types of monitoring:
  - Historical data gathering
    - Trends for capacity planning
    - Recognition of long-term improvements
  - Real-time monitoring and alerting
    - Detect problems faster
    - React before failure (e.g., before swap gets full)

Logging policies

- Log files grow and grow
- What do you do with log files? Some options:
  - Throw away all data immediately
  - Reset log files periodically
  - Rotate log files, keeping data for a fixed time
  - Compress and archive files to tape or other media
- Throwing away log files
  - Not recommended!
  - Need evidence of security problems
  - Alert for hardware and software problems
  - Ideally, keep for a month
    - may take that long to notice a problem!
  - Resetting when disk is full isn't good either

Rotating log files

- Keep a fixed set of previous log files
  - Rotate current file into set on a regular basis (daily, weekly, etc.)
  - Example:

```
#!/bin/sh
cd /var/log
mv logfile.2 logfile.3
mv logfile.1 logfile.2
mv logfile logfile.1
touch logfile
chmod 600 logfile
```

  - May want to add compression, reset server

Archiving log files

- May need to archive all accounting data and log files for policy, potential audits, etc.
- First rotate on disk
  - fast access to recent data
- Then write to tape or other media
- Log files should be part of backup sequence
  - Hackers tend to delete them!

Linux log files

- Most log files are recorded in /var/log
  - /var/adm may also contain some (distro dependent)
- Most programs send entries to syslog daemon
  - /etc/rsyslog.conf usually puts them in /var/log
- Sample log files:
  - messages – main system log file
  - maillog – record of sendmail activity
  - boot.log – output of system startup scripts

Other log files

- /var/log/wtmp
  - Record of users' logins and logouts
  - Binary format – use **last** to read
  - Still truncated and rotated
- /var/log/lastlog
  - Record of time of last log in
  - Binary format (is used to say when you last logged in)
  - Constant size – no need to rotate
- /var/log/dmesg
  - Dump of kernel message buffer at end of boot

Logrotate

- Excellent utility to manage log files
- Specifies groups of log files to be managed

```
# Example log rotation
rotate 5
weekly

/var/log/messages {
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid`
    endscript
}
/var/log/samba/*.log {
    notifempty
    copytruncate
    postrotate
        /bin/kill -HUP `cat /var/lock/samba/*.pid`
    endscript
}
```

Syslog

- Comprehensive logging system
  - Frees programmers from needing to write their own
  - Allows sysadmins to control logging
- Flexible
  - Can sort by source or severity level
  - Output to variety of destinations – files, terminals, other machines
- Can centralize logging to a well-controlled machine
- RHEL/CentOS 6 uses rsyslog (improved)

Syslog (continued)

- Three parts
  - syslogd – logging daemon (uses /etc/rsyslog.conf)
  - openlog – library routines
  - logger – shell command to submit log entries
- Apps use library to write to /dev/log
  - UNIX domain socket
- Syslogd reads messages from /dev/log
  - Outputs message depending on /etc/rsyslog.conf

Sample syslog.conf

```
# Emergencies: tell everyone who is logged in
*.emerg;user.none                            *

*.warning;daemon,auth.info,user.none         /var/log/messages
# Forward important messages to the central logger
*.warning;daemon,auth.info                   @netloghost

# printer errors
lpr.debug                                    /var/log/lpd-errs
```

Sample syslog output

```
Feb 22 04:04:21 wume1 named[2826]: lame server
  resolving '211.68.246.64.in-addr.arpa' (in
  '68.246.64.in-addr.arpa'?): 160.79.6.130#53
Feb 22 13:22:41 wume1 sshd(pam_unix)[16776]: session
  opened for user brian by (uid=0)
Feb 22 13:22:44 wume1 su(pam_unix)[16802]: session
  opened for user root by brian (uid=501)
Feb 25 20:31:57 wume1 sshd(pam_unix)[28375]: check
  pass; user unknown
Feb 25 20:32:00 wume1 sshd(pam_unix)[28375]: 1 more
  authentication failure; logname= uid=0 euid=0
  tty=NODEVssh ruser= rhost=dyn032098.cc.lehigh.edu
```

Condensing log files

- Syslog (as well as any other monitoring and logging facility) generates lots of log files
- Need utilities to scan log files and find important entries
  - security-related entries
  - messages about disks full
  - messages repeated many times

Summary

- It is imperative to monitor systems and generate logs
  - For warnings, job performance, trends, etc.
- Logs cannot be permitted to impact proper system operation