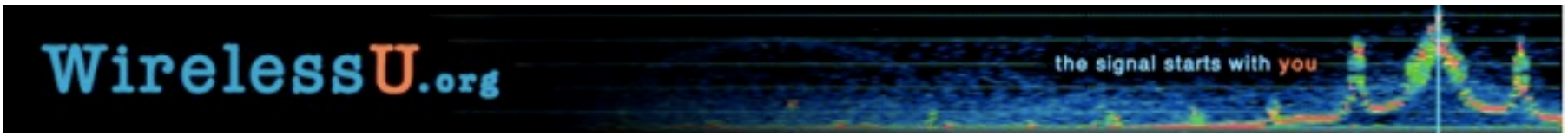
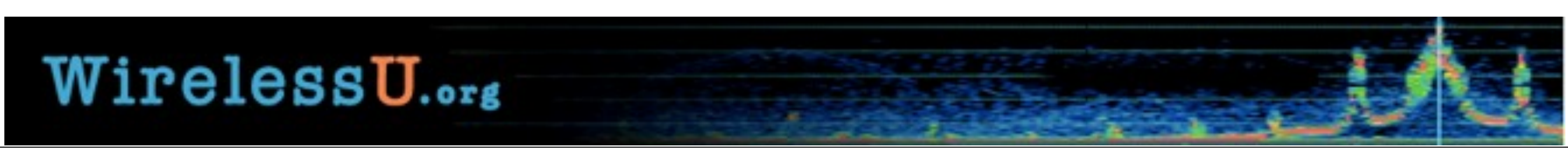


Monitoring tools and techniques for ICT4D systems

Stephen Okay

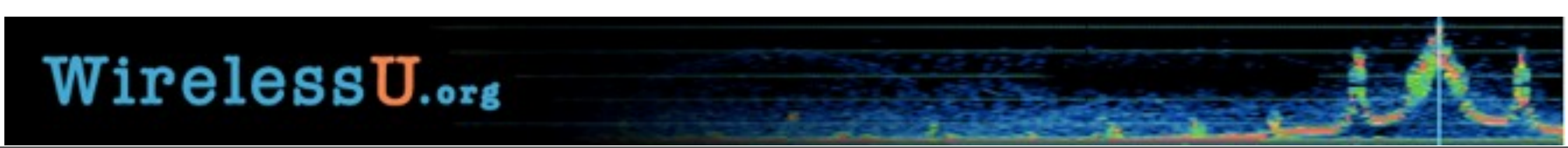
Effective Monitoring

- Why do monitoring ?
- Monitoring tools and Applications
- Monitoring:What,Where, Why,How, etc.
- Alerting
- Off-the-shelf vs. Custom scripts
- Monitoring protocols and methods
  - Ports
  - SNMP
  - SSH
- Common problems with monitoring



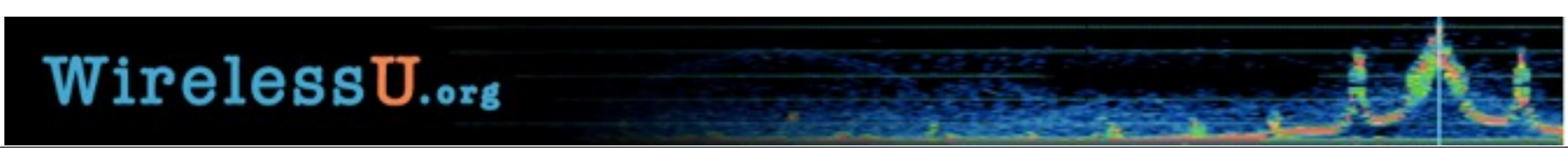
Why do monitoring ?

- Monitoring systems are the “radar” of system and network management.
- To track the performance or degradation of systems, devices and application
- To be notified when things go wrong
- To catch problems before they grow to be serious
- As a way to show system and network usage and aid in capacity planning.



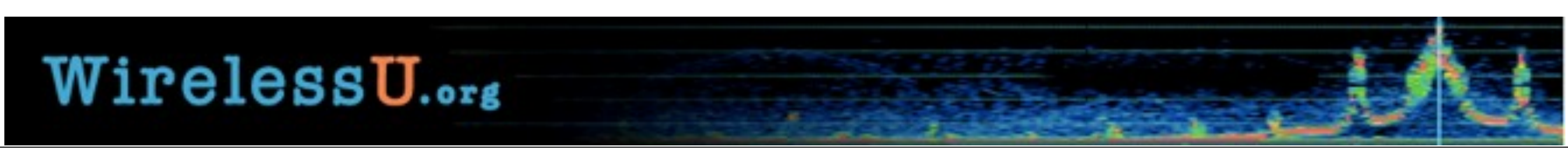
The RRDtool suite of monitoring applications

- RRDtool - Round Robin Database tool
  - http://oss.oetiker.ch/rrdtool/
- Nagios - Programmable monitoring and alert system
  - http://www.nagios.org
- Cacti - Extensive, customized graphs from RRDs
  - http://www.cacti.net
- MRTG - Multi-Router Traffic Grapher - Monitor network traffic from multiple routers/sources
  - http://oss.oetiker.ch/mrtg
- SmokePing - More specialized towards measuring network latency and congestion
  - http://oss.oetiker.ch/smokeping/

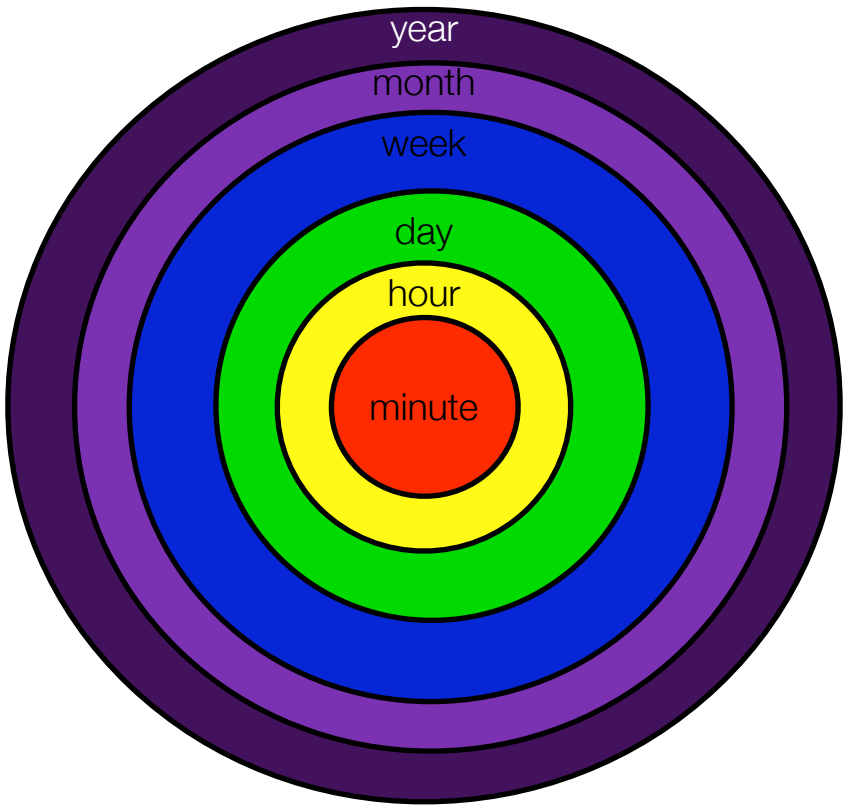


RRDtool

- Stores time-series data in a ring or “round-robin” database(RRD) that can store long histories of data like CPU, memory usage, network I/O, without growing in size.
- Each RRD consists of a series of datapoints recorded at user-defined intervals.
- When the RRD is full, the current data overwrites the first entry(s).
- Successive RRDs use a consolidation function to store longer term data.
  - Average of all data points in a RRD
  - Minimum value
  - Maximum value
- A group of RRDs is a “Round-Robin Archive”(RRA)



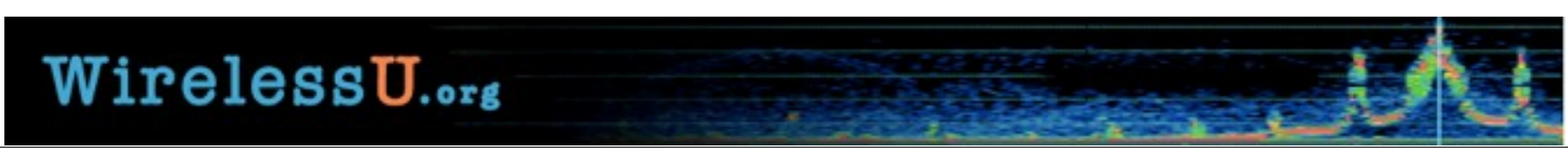
Example RRA: One year's worth of time-series data



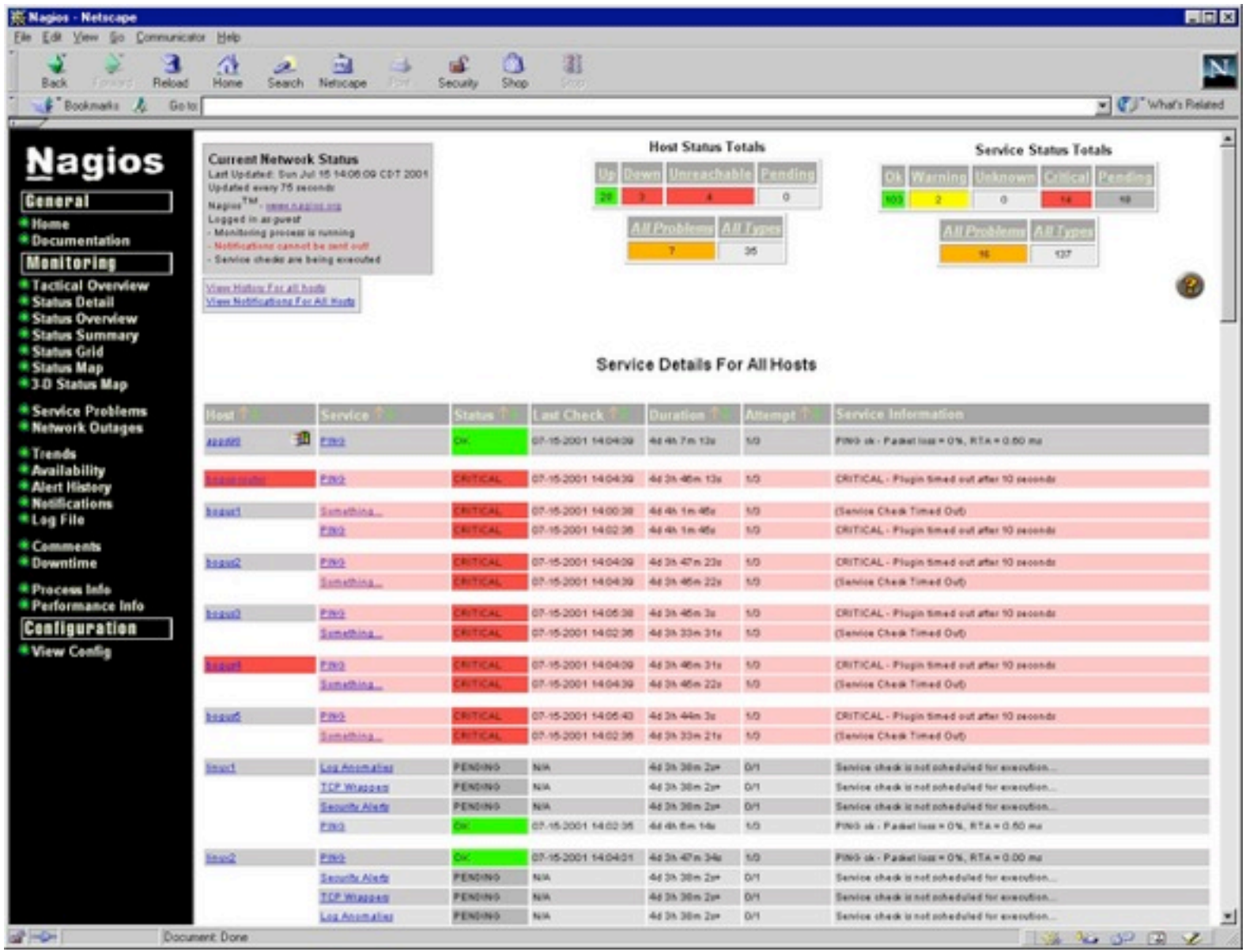
Outer ring data point=consolidation(next inner ring data set)

Example RRD

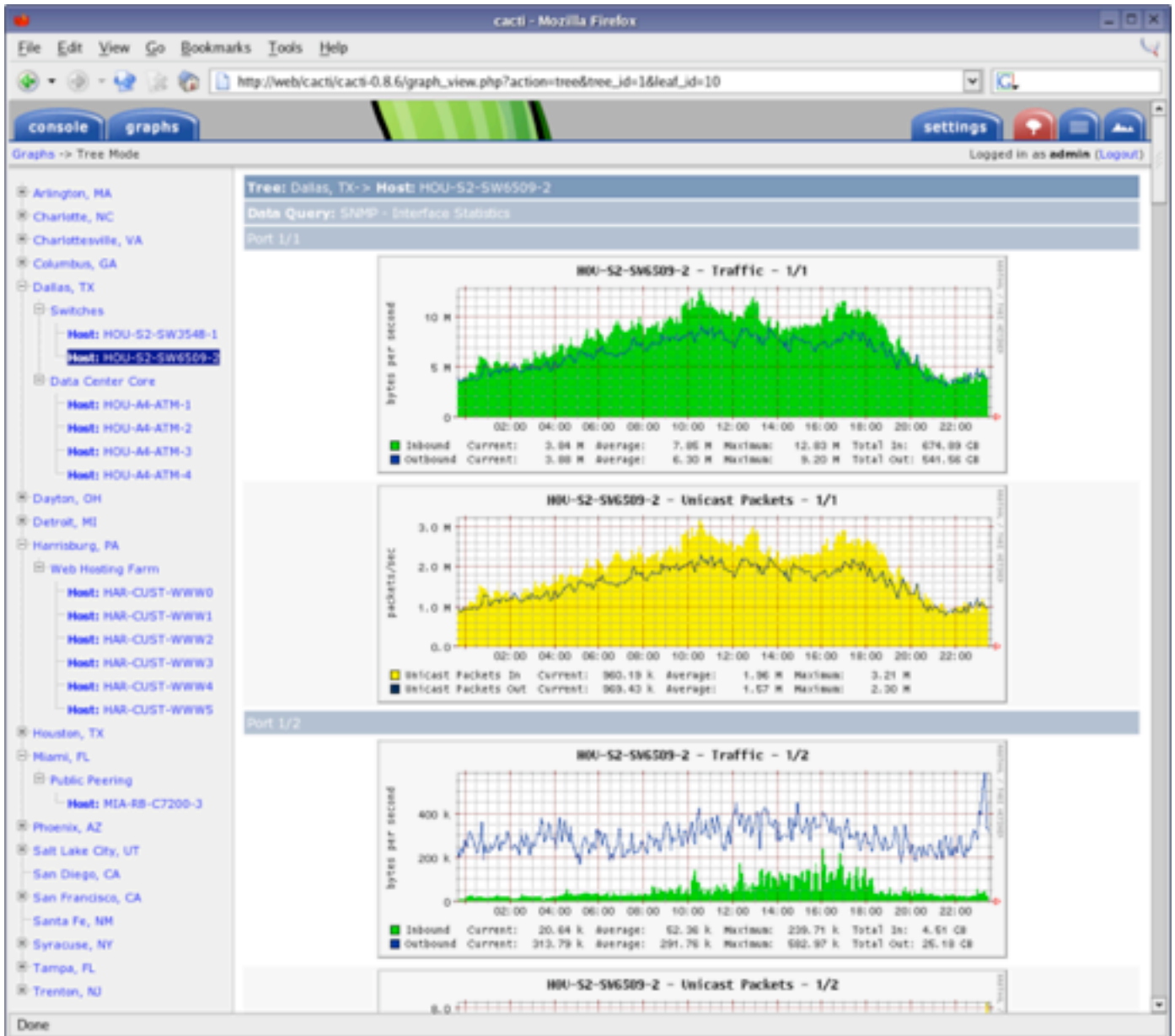
- There are 31,536,000 seconds in a calendar year.
- To track the temperature inside an access point every 5 minutes for a year, you could collect and store 6.3M records OR
- Create a RRD that stores the samples like:
  - One RRA that stores 1 temp. reading ever 5 minutes for 1200 readings(100 hours)
  - One RRA that downsamples 12 5-minute readings into a 1-hour average and stores 100 days of hourly average temperatures
  - One RRA that downsamples 24 1-hour averages into a daily average and stores 300 days of daily average temperatures
  - ...and so on
  - RRDs let you store a lot of data in not a whole lot of space!



Nagios

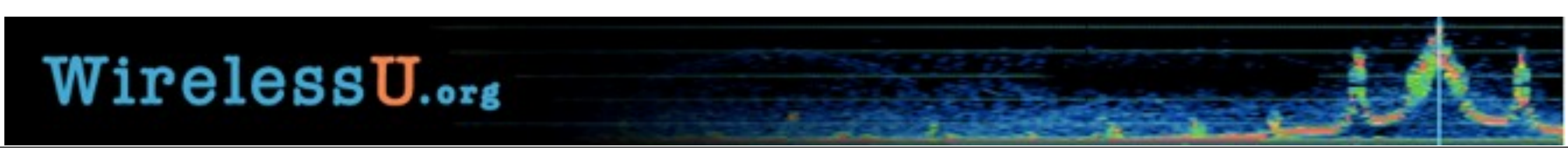


Cacti


cacti - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://web/cacti/cacti-0.8.6j/graph_view.php?action=tree&tree_id=1&leaf_id=10
console
graphs
settings
Graphs -> Tree Mode
Logged in as admin (Logout)
Arlington, MA
Charlotte, NC
Charlottesville, VA
Columbus, GA
Dallas, TX
Switches
Host: HOU-S2-SW3548-1
Host: HOU-S2-SW6509-2
Data Center Core
Host: HOU-A4-ATM-1
Host: HOU-A4-ATM-2
Host: HOU-A4-ATM-3
Host: HOU-A4-ATM-4
Dayton, OH
Detroit, MI
Harrisburg, PA
Web Hosting Farm
Host: HAR-CUST-WWW0
Host: HAR-CUST-WWW1
Host: HAR-CUST-WWW2
Host: HAR-CUST-WWW3
Host: HAR-CUST-WWW4
Host: HAR-CUST-WWW5
Houston, TX
Miami, FL
Public Peering
Host: MIA-RB-C7200-3
Phoenix, AZ
Salt Lake City, UT
San Diego, CA
San Francisco, CA
Santa Fe, NM
Syracuse, NY
Tampa, FL
Trenton, NJ
Tree: Dallas, TX-> Host: HOU-S2-SW6509-2
Data Query: SNMP - Interface Statistics
Port 1/1
HOU-S2-SW6509-2 - Traffic - 1/1
bytes per second
Inbound Current: 3.04 M Average: 7.05 M Maximum: 12.03 M Total In: 674.89 GB
Outbound Current: 3.08 M Average: 6.30 M Maximum: 9.20 M Total Out: 541.56 GB
HOU-S2-SW6509-2 - Unicast Packets - 1/1
packets/sec
Unicast Packets In Current: 960.19 k Average: 1.96 M Maximum: 3.21 M
Unicast Packets Out Current: 969.43 k Average: 1.57 M Maximum: 2.30 M
Port 1/2
HOU-S2-SW6509-2 - Traffic - 1/2
bytes per second
Inbound Current: 20.64 k Average: 52.36 k Maximum: 239.71 k Total In: 4.51 GB
Outbound Current: 313.79 k Average: 291.76 k Maximum: 502.97 k Total Out: 25.10 GB
HOU-S2-SW6509-2 - Unicast Packets - 1/2
Done

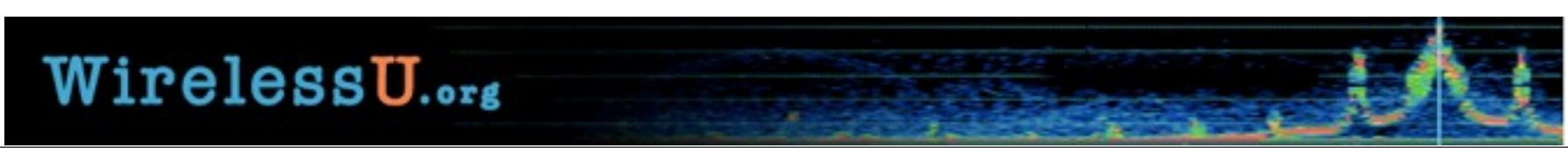
A Good Monitoring Plan Answers

- **What** are you going to monitor ?
- **Where** is this going to happen from ?
- **When**/How often do you need to do this ?
- **Why** is this monitor needed ?
- **Who** is going to answer alerts ?
- **How** will people know there's a problem ?



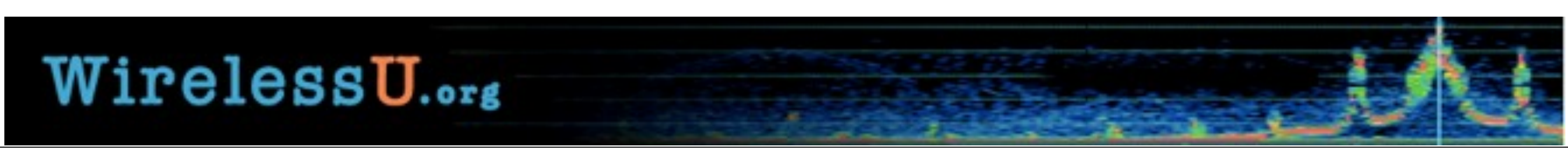
What to monitor

- Systems, devices and applications
  - that are critical to the function of your network
  - that other systems or people depend on.
  - that can give you clear useful information on their status
  - that you have some control or influence over
- Examples
  - HTTP check against a well-known site (Google, CNN, etc.) for external connectivity
  - Ping against far-side router/AP for a WLAN(for connectivity and throughput)
  - Temperature/Humidity check
  - Battery Level/Power status check
  - Send/Receive Usage checks



Alerting and Escalating

- Alerts
  - Should be sent for some critical problem that need immediate attention
    - Disk filling up
    - Temperature/Humidity, etc.
    - Physical Intrusions on-site
- Alert on what you can respond to
  - Nobody likes to be woken up @ 3AM for a system they don't have access to or can't control
- Keep monitors/services that actually send alerts to a minimum
  - Ask what would happen if the situation persisted for
    - 5 minutes, 30 minutes, 60 minutes, a day
    - If the answer is not something like "we would go out of business" or "the experiment would be ruined", you probably don't need to send an alert about it, just warn about it on the GUI or send an email message.



COTS vs. Custom monitors in Nagios

- COTS(Common Off-the-Shelf)
  - Lots of them out there, for many common devices and situations
    - System: CPU activity, memory/ disk usage, network I/O, network errors
    - Service:IP Port monitors, ping time, HTTP, ssh, etc.
    - Health:CPU temp,Case temp, battery voltage, humidity, etc.
  - About 70% of most monitoring tasks can be accomplished with an existing plug-in.
- Custom Monitors
  - When you need to monitor output from a device/ application that you can't get any other way.
  - Or as a learning tool...

Guidelines for writing custom Nagios plugins

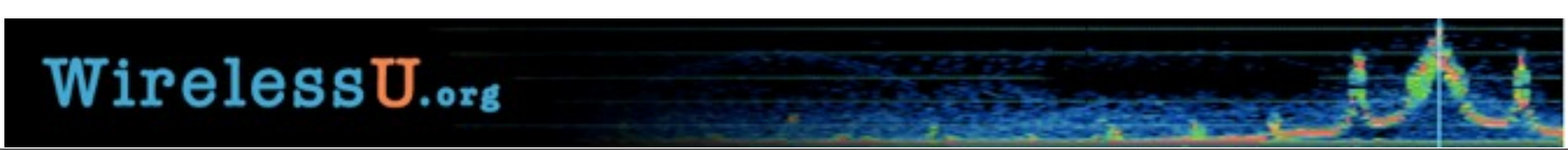
- Use a low/mid-level language that you're familiar with
  - BASH, Perl, Python, C
- Avoid things like Java which can be very heavy or have significant startup times
- Keep runtime to seconds, not minutes
  - plugins need to run and respond quickly or they may hold up other monitors or give erroneous results
- Return data
  - Max 1 line of output
  - return only data or data followed by an exit code:
    - 0-OK, 1-WARN, 2-CRITICAL, 3-UNKNOWN
    - script errors/failures should return UNKNOWN
    - Use CRITICAL only for data that crosses the critical threshold
    - Use WARNING or UNKNOWN for everything else

Writing custom plug-ins (cont'd)

- test, Test, TEST!

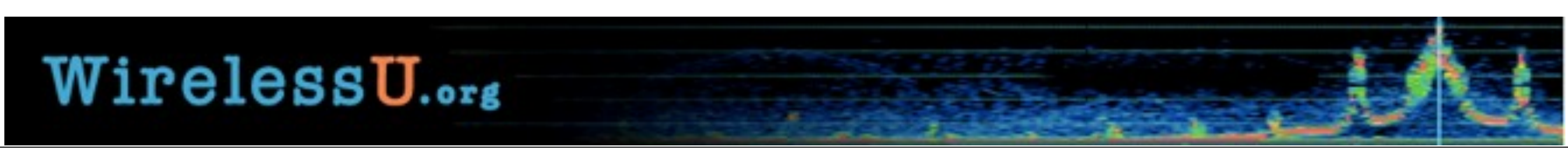
Protocols/Methods for running monitors

- Basic “Is it open ?” checks
  - TCP/UDP port probes
    - Uses a standard TCP/UDP socket connection to establish that a port is open or not.
    - Assumption:
      - Open - Working
      - Closed - Down/blocked
  - Service Status checks - “Anybody home ?”
    - HTTP -get known page, look at return code
    - SIP server - can you start a session with the server ?
    - Mail
    - MySQL - try check or set a simple counter entry
  - Firewall rules can affect these checks and give false negatives



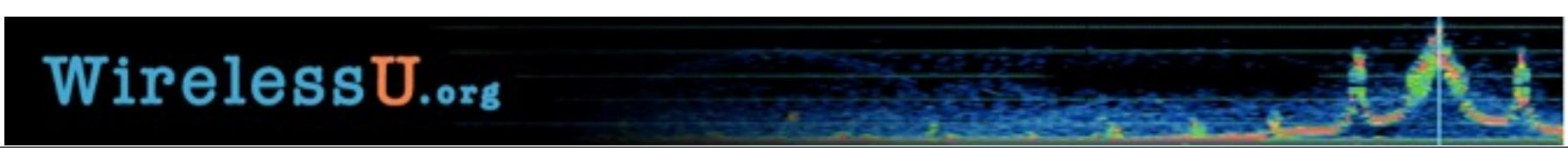
Protocols/methods for running monitors

- SNMP
  - Simple Network Management Protocol
  - Described in IETF RFC(s) 1157(original) 3410-3418(current)
  - Current release is v3, although many devices in the field still using v2c
  - Used mostly for monitoring devices:routers,switches,printers, UPSes, environmental sensors, but also vending machines, small children, etc.
  - low-resource usage
  - Industry standard
  - Managers/Agents communicate data between host system and devices
  - Insecure, v1 & v2 send passwords in cleartext
  - Not the best choice when you need to the agent to return complex data
    - Can run scripts, but parameter passing can be tricky/impossible



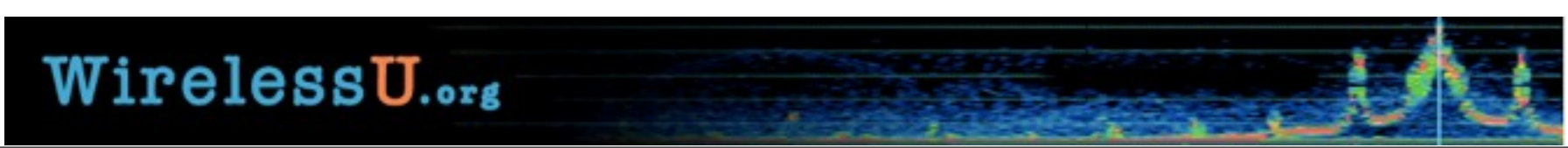
Protocols/Methods for doing monitoring

- SSH
  - Secure SHell
  - Normally used for encrypted remote interactive access
  - Public keys can execute remote commands non-interactively.
  - Can run scripts that require or return rich data sets
  - Encryption can use significant resources on monitoring server, esp. as monitoring activity grows.



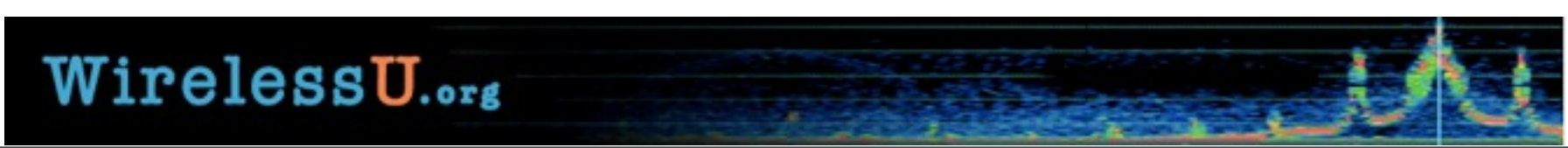
Common problems in system/network monitoring

- Flapping
  - Monitor constantly triggers as critical values bounce or “flap” over and under threshold values.
- Critical threshold too high
  - Monitor never triggers an alert
- Critical threshold too low
  - Monitor always alerts, even target device/service is fine
- Script errors
  - Script crashes due to a syntax or other error, so the check never actually completes so the actual state of the system is never known.
- Misapplied monitors
  - Example: A machine that is both a web and mail server becomes just a web server, but somebody forgets to turn off the mail service monitor for that machine.



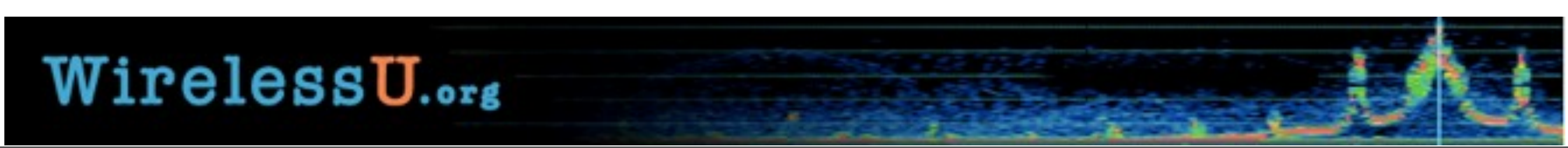
Solutions to system/network monitoring problems

- Flapping
  - “loosen” or adjust thresholds. For example, allow a longer amount of time at a critical state before alerting.
- Threshold too high/low.
  - Adjust in the appropriate direction
- Scripts
  - Test repeatedly with multiple options and values until you are certain its working properly and you understand how to run it by hand.
- Misapplication
  - If you are monitoring a system or service, including a monitoring review in any migration/change plan to make sure that old monitors are still needed.



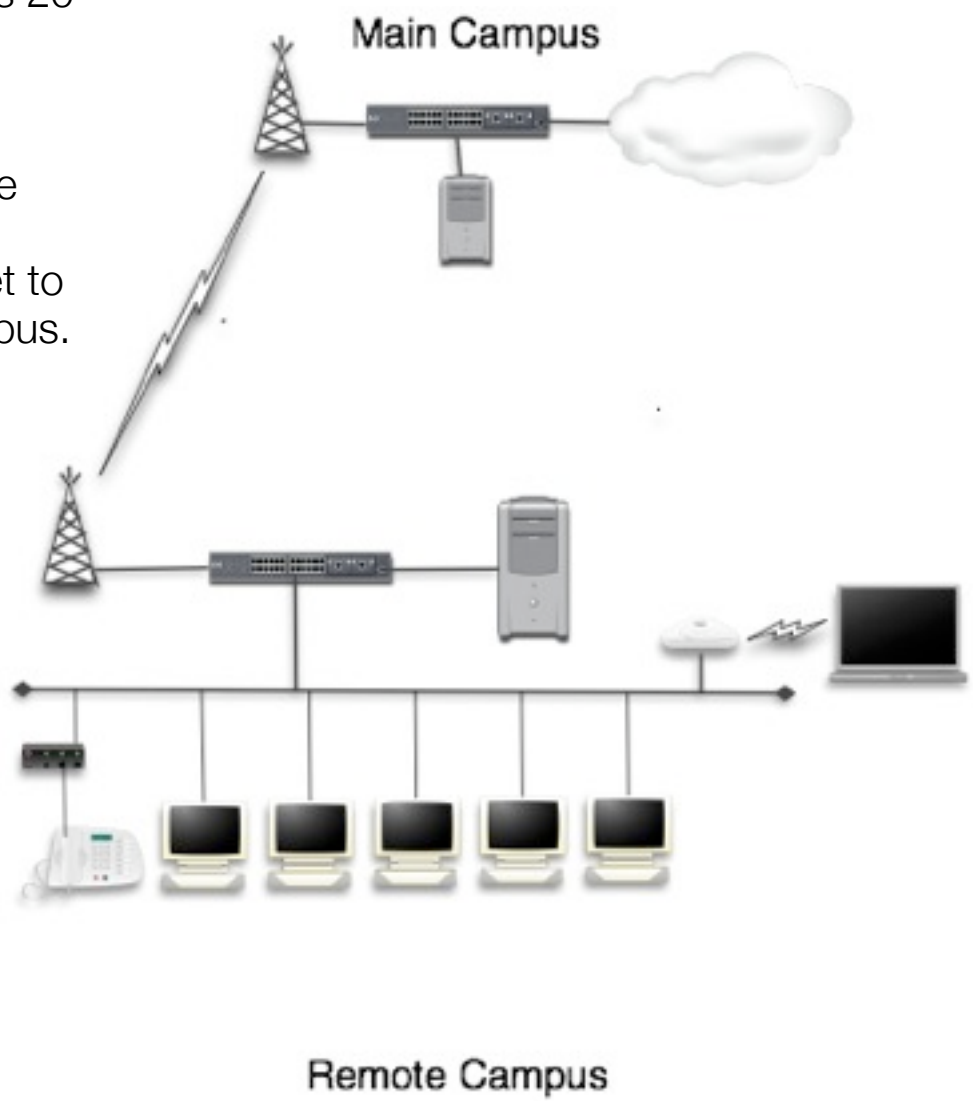
Some Closing thoughts

- Practice “Conservation of Monitoring”
  - Resist the temptation to monitor everything
  - Monitor from the edge of your control
- As the network grows...
  - Remember that each additional monitor adds load to the network
  - Stagger monitoring activities to avoid polling “storms”
  - Regularly examine your monitoring implementation to tune thresholds, add monitors where they would help and remove old ones



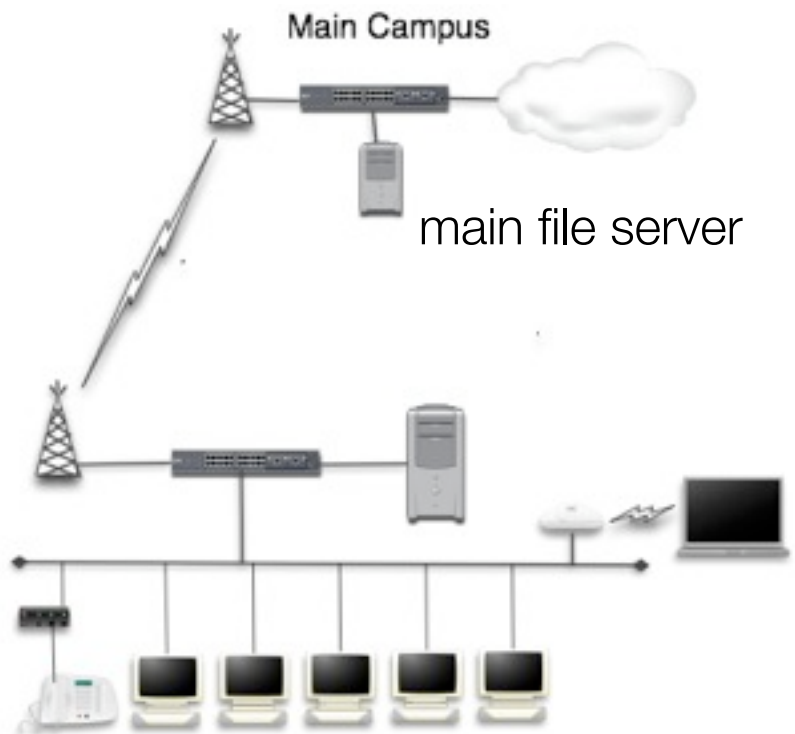
Exercise 1

You've just set up a wireless link between the main campus at your university and a remote campus 20 km away.
How would you check to see:
1. The wireless link is up
2. Both sites can connect to the Internet
2. Remote workstations can get to the file server on the main campus.



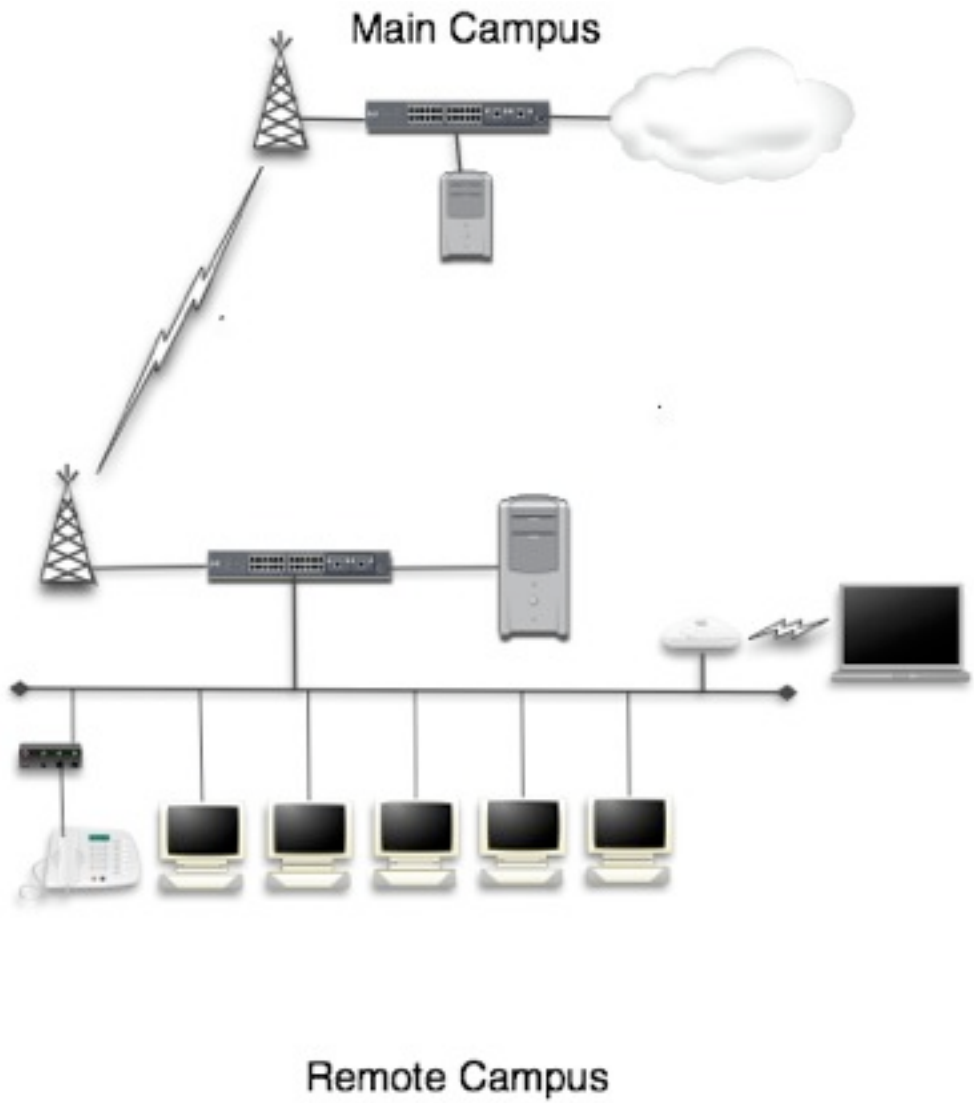
Exercise 2

Users working on a medical records project at the remote campus are connecting to a file server at the main campus. The connections to this server keep dropping though, disrupting work and wasting time. What would you monitor as part of diagnosing the problem ?

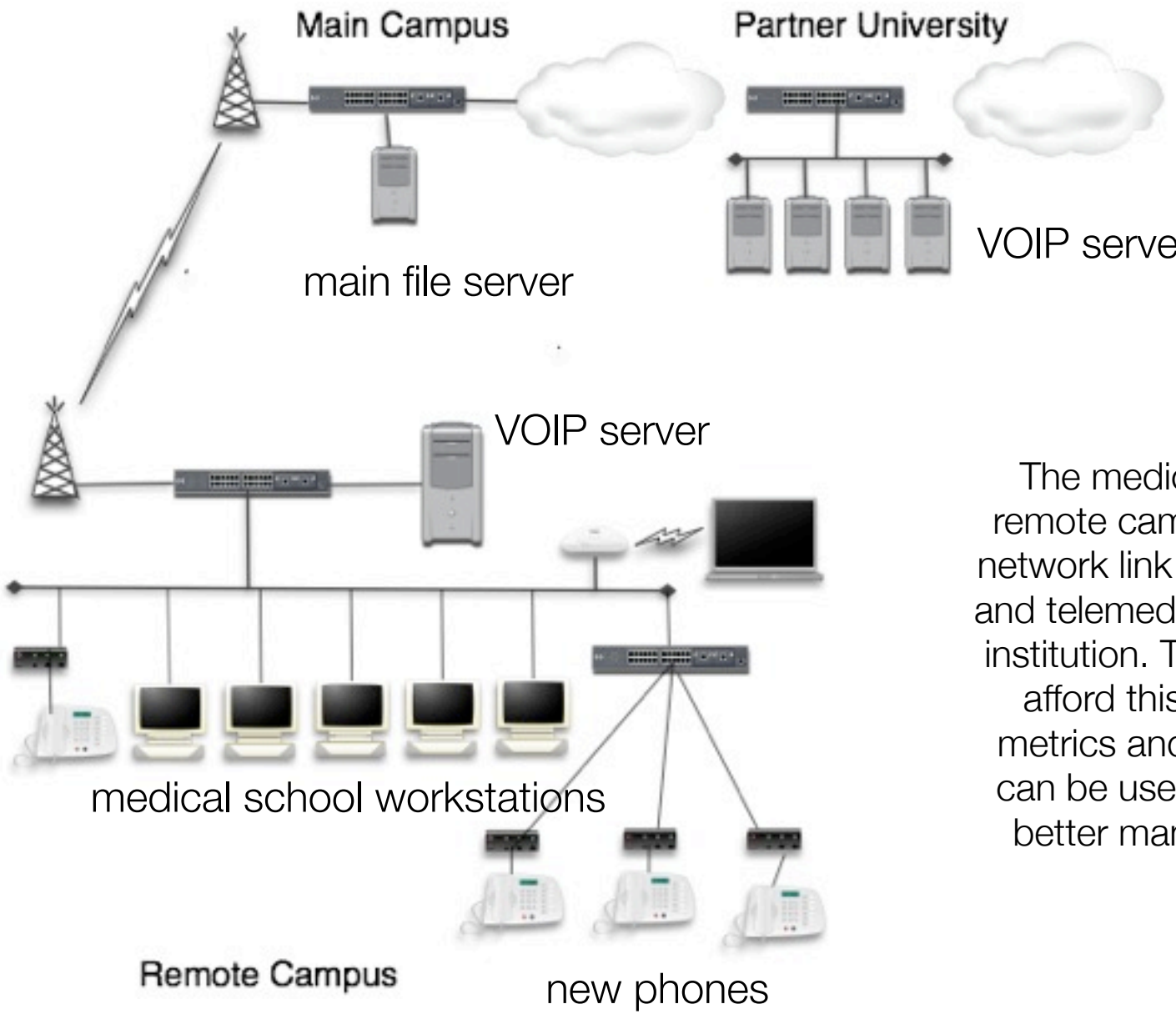


Exercise 2

The bill for last month's Internet usage at the main campus was very high. They suspect that someone at the remote campus is abusing the network.
How would you monitor the usage from the remote site to prove or disprove this ? What sort of tools other than Nagios would you need to determine this ?



Exercise 4



The medical school at the remote campus wants a new network link dedicated to VOIP and telemedicine with a partner institution. The university can't afford this however. What metrics and monitoring tools can be used to show how to better manage the existing link ?