An Analysis of Network Configuration Artifacts

LISA '09, November 5, 2009

David Plonka & Andres Jaan Tack
{plonka,tack}@cs.wisc.edu

Motivation and Goals

- Like software quality, network reliability is evolving:
  - Expectation of high availability, increasing reliance
  - Increasing numbers of skilled practitioners
  - Increasing level of automation

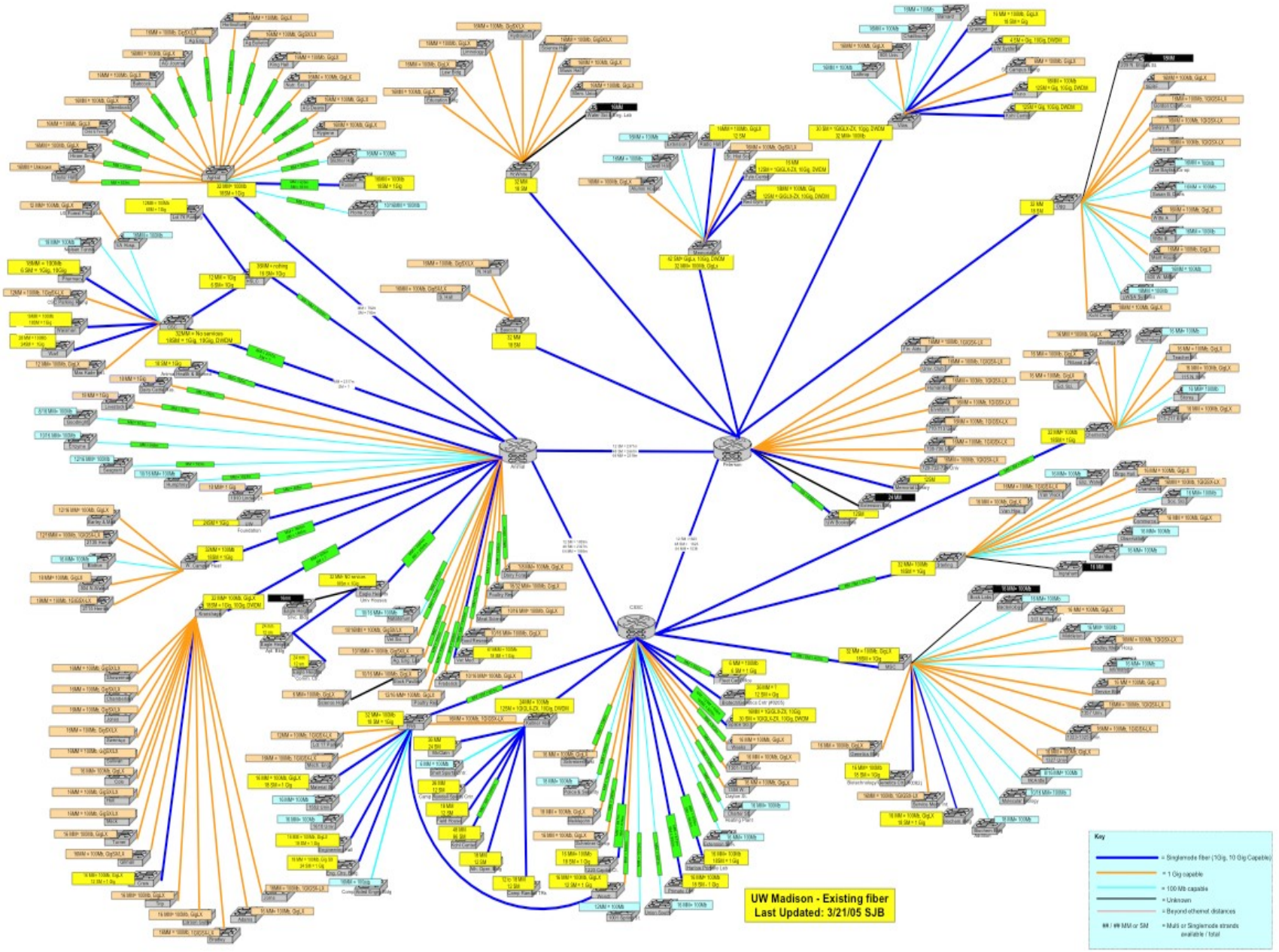
Motivation and Goals

- Like software quality, network reliability is evolving:
  - Expectation of high availability, increasing reliance
  - Increasing numbers of skilled practitioners
  - Increasing level of automation
- However, the management of networks and the Internet has not received similar attention to the development of software.

Motivation and Goals

- Like software quality, network reliability is evolving:
  - Expectation of high availability, increasing reliance
  - Increasing numbers of skilled practitioners
  - Increasing level of automation
- However, the management of networks and the Internet has not received similar attention to the development of software.
- We propose an ***analogy-based analysis***, and that these elements are akin to each other:
  - Networks : Software Systems
  - Network Engineering : Software Engineering
  - Network Operators : Programmers

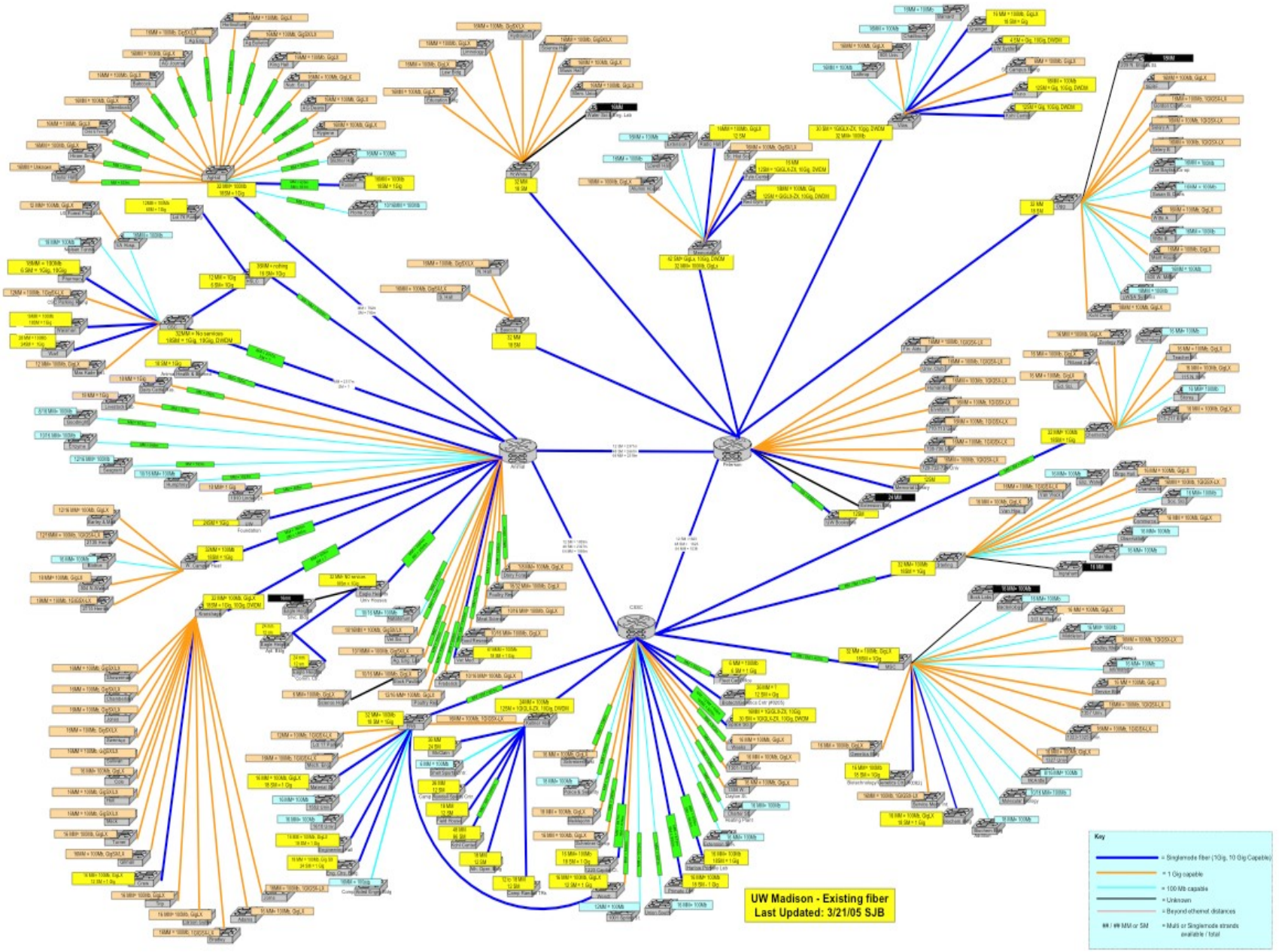
Campus Network



Network Artifacts

- *artifact* - an object created by humans, especially one remaining from a particular period
- Network Performance Measurements
- Network Management Systems' Topology
- Trouble Tickets
- ***Network Device Configurations***
  - Routers, switches, firewalls
  - Network practitioners use Source Code Management (SCM) of device configurations for:
    - Configuration backups
    - Communicating changes

Network Configuration Repositories



Networks Studied

Network	*Period in Years*	*Operators (super-users)*	*Devices / Configuration Files*	*Revisions*	*Lines of Code*
Campus	5+	343 (64)	3,839	128,394	2,898,362
Service Provider	10+	31 (31)	519	41,787	163,882

Mining SCM Repositories - Why?

- While successful in the PL community, this hasn't been leveraged in the context of network configuration and management.
- To visualize and elucidate network operation with the goal of understanding and improving the practice.

Mining SCM Repositories - How?

- Convert existing custom network version control system repositories to common CVS repositories.
- Use existing tools from the Programming Language (PL) and open source developer communities, e.g.:
  - StatCVS-XML
  - cvs2cl (CVS to ChangeLog)
- Perform additional static file analyses, e.g.:
  - Syntax-aware statistics (i.e. config stanzas)
  - Revision lifetimes

Configuration Files / Code Sample

```
version 12.2
no service pad
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
!
hostname s-bldg-5-2-access
!
spanning-tree mode rapid-pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
```

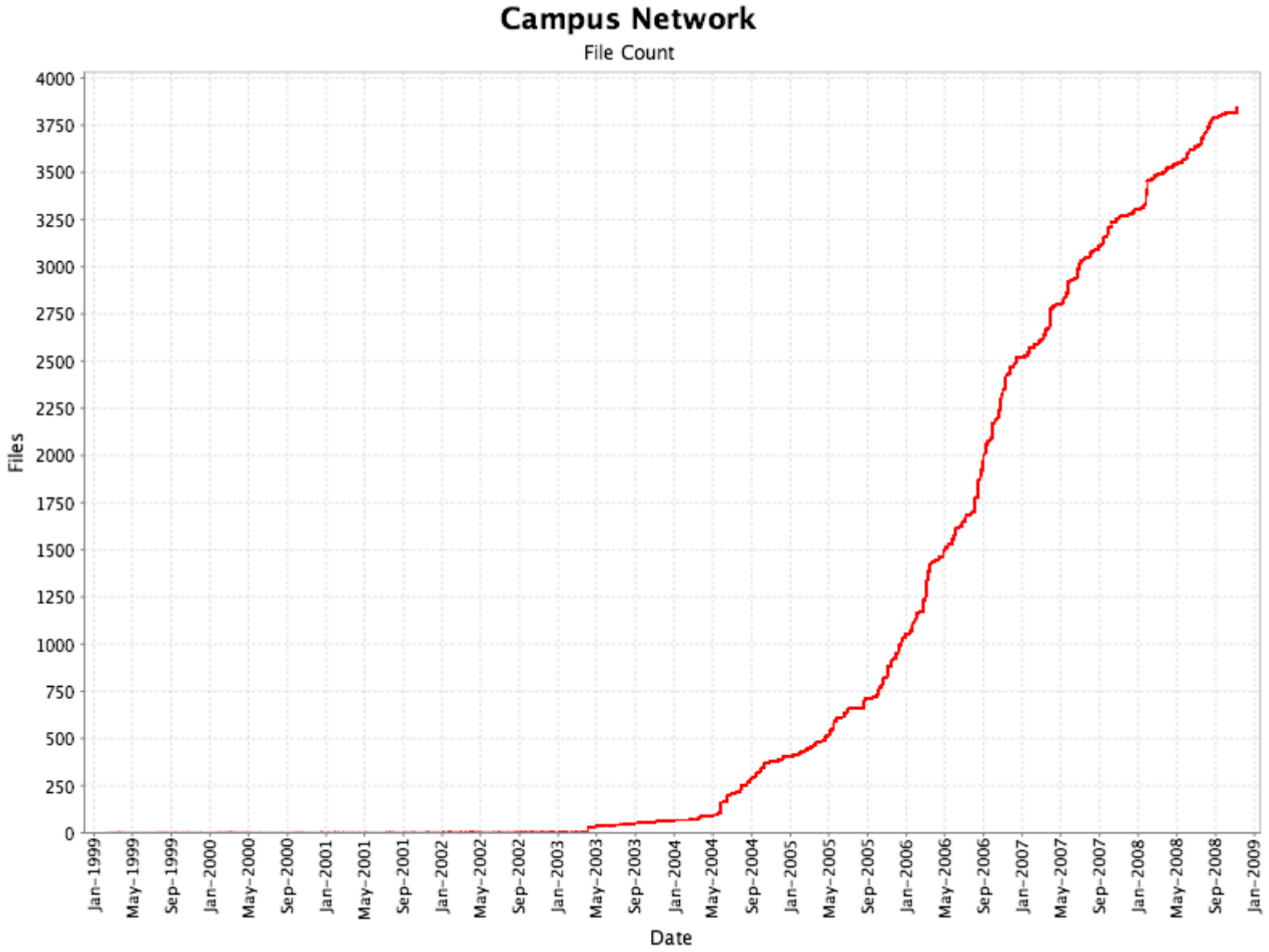
Code Sample (2)

```
interface FastEthernet1/0/1
 description sample 100Mbps ethernet interface
 switchport access vlan 42
 switchport mode access
 ip access-group nodhcpserver in
 snmp trap mac-notification change added
 snmp trap mac-notification change removed
 no snmp trap link-status
 no mdix auto
 spanning-tree portfast
 spanning-tree bpduguard enable
 spanning-tree guard root
```

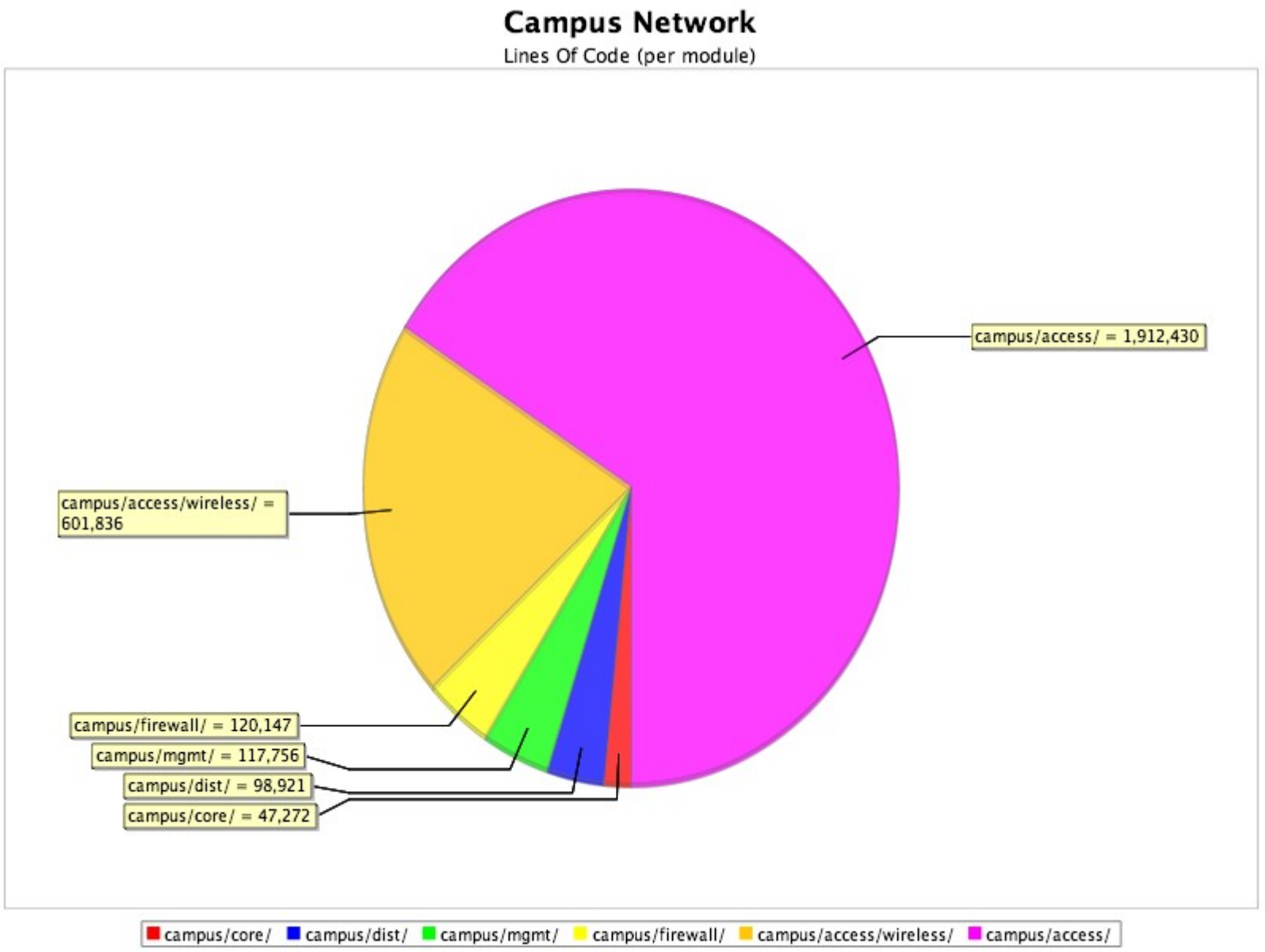
Code Sample (3)

```
ip access-list extended nodhcpserver
   remark Id: ndhcp.acl,v 1.2 2005-05-20 11:26:03 ashley Exp
   deny   udp any eq bootps any
   permit ip any any
!
access-list 5 permit 192.2.0.1
access-list 5 remark Allow foo, bar, and baz servers
access-list 5 permit 192.2.0.10
access-list 5 permit 192.2.0.11
!
...
!
```

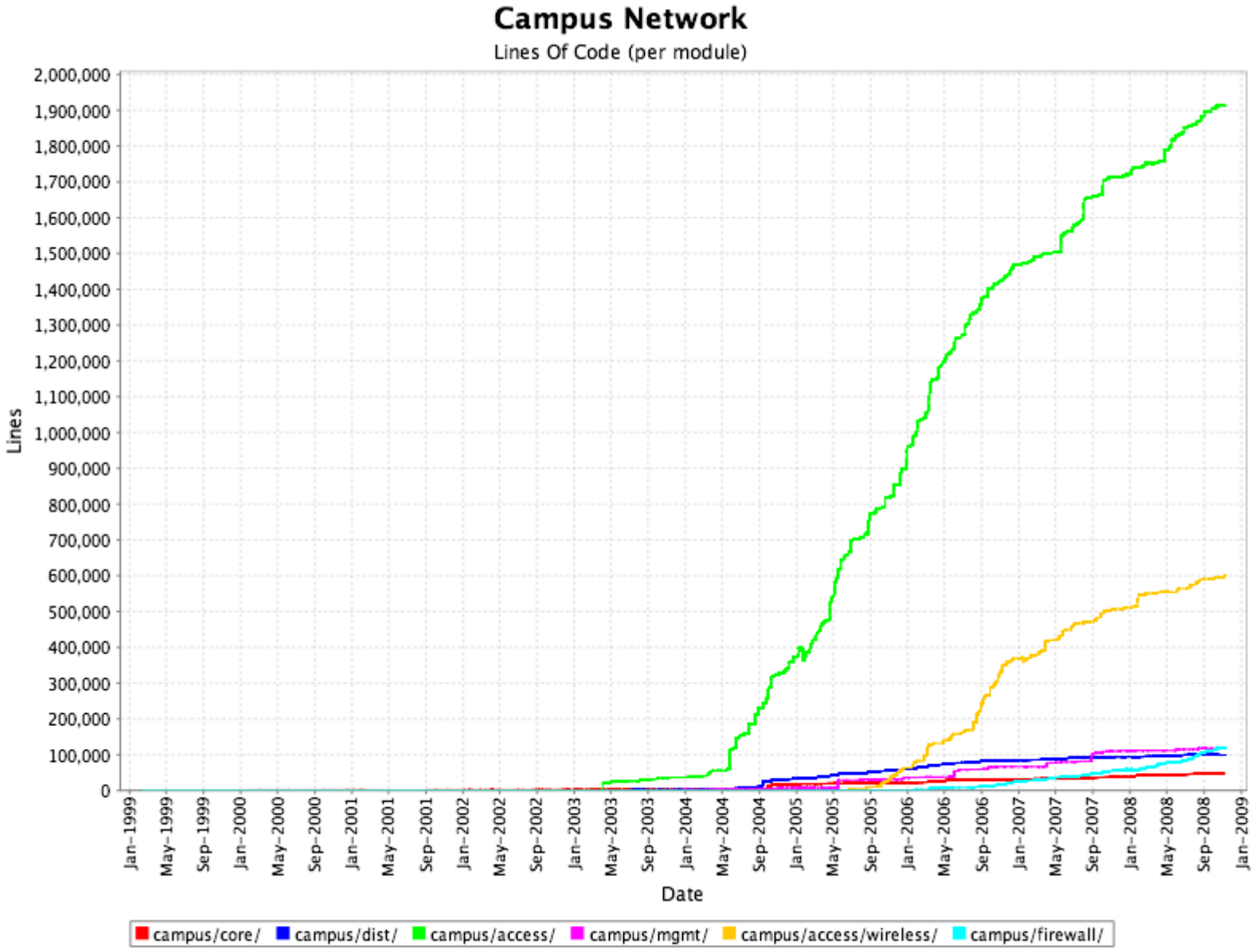
Campus File / Device Count



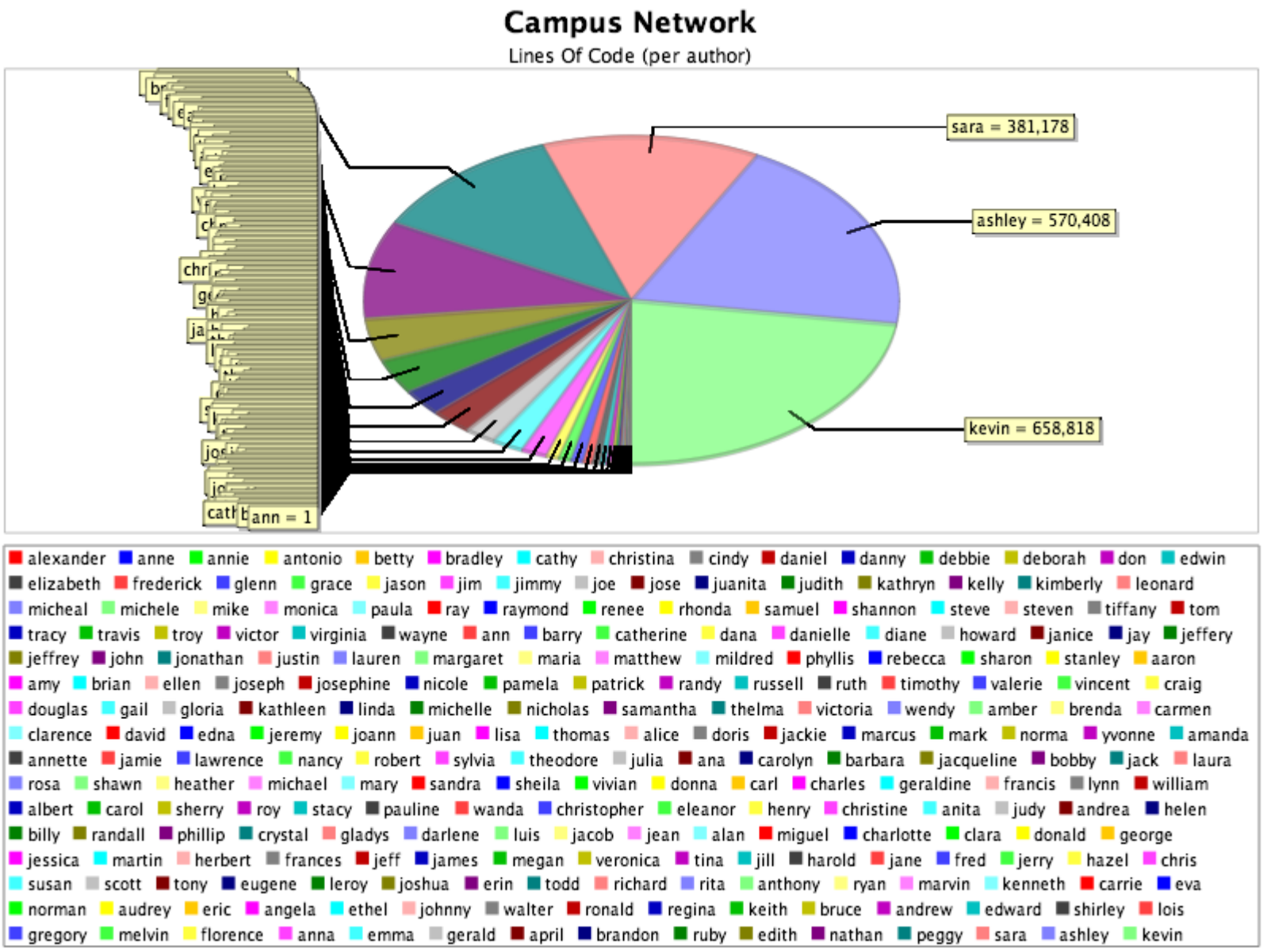
Campus LOC by Topology



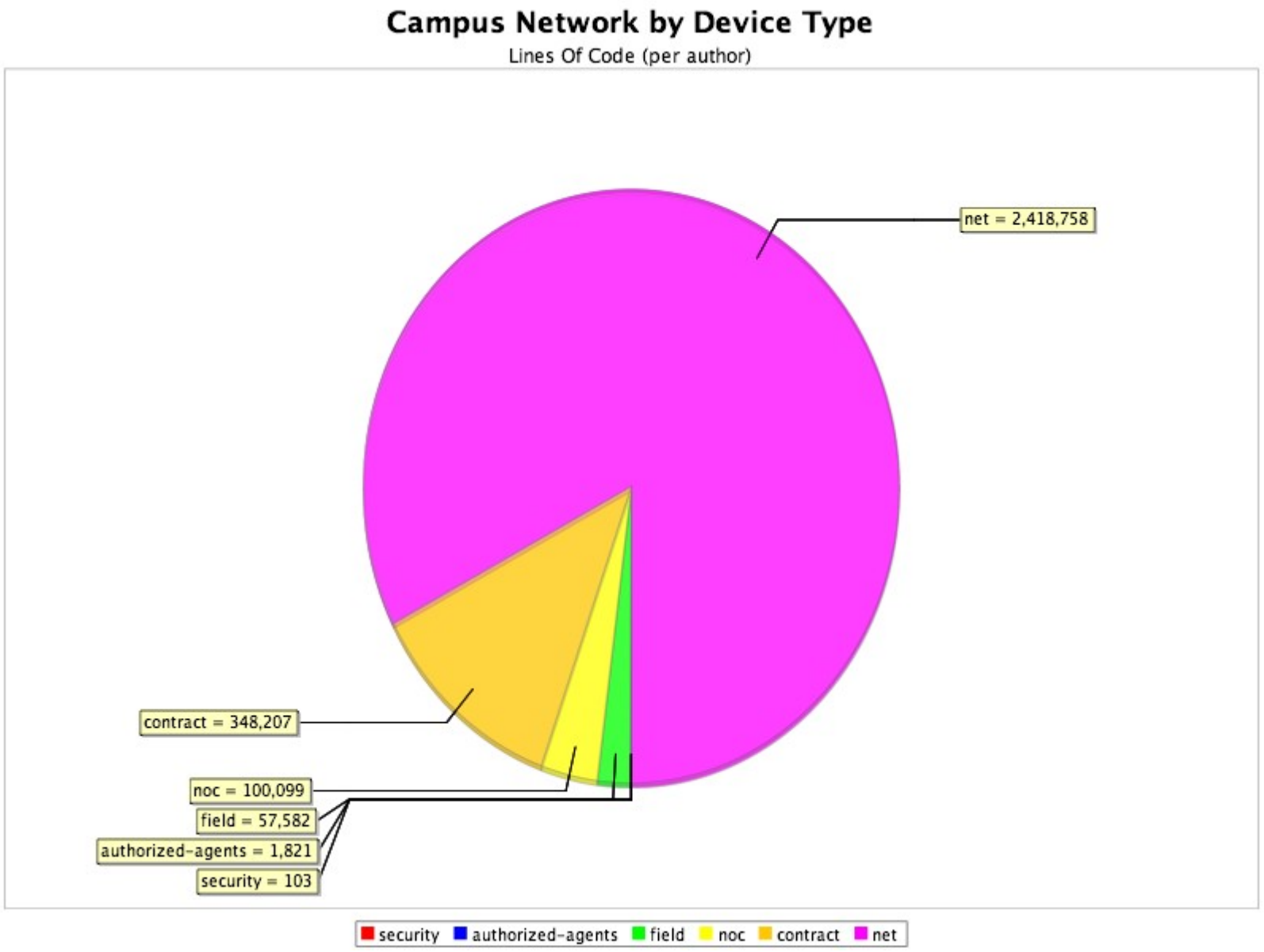
Campus LOC per Module



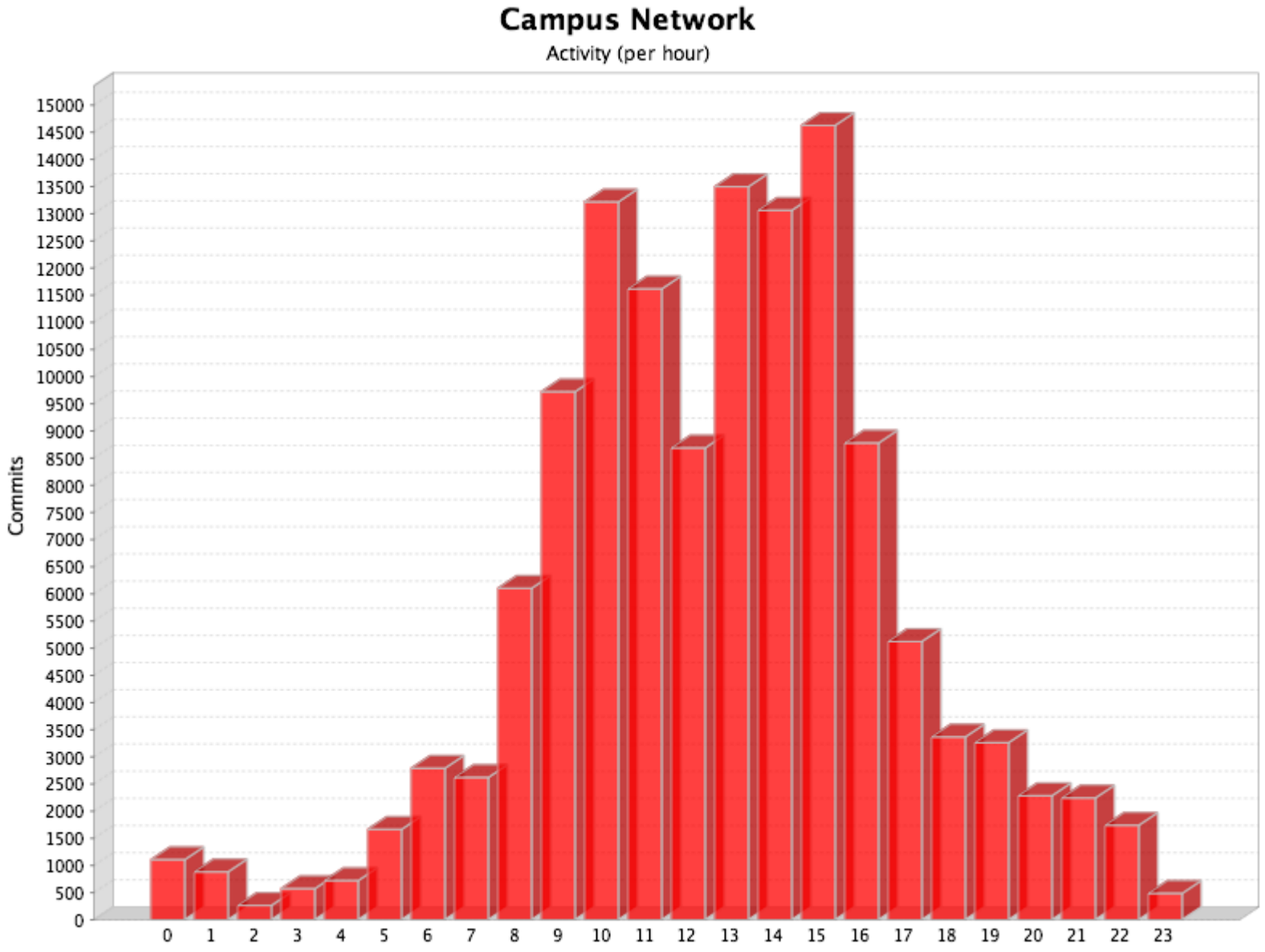
Campus Size Per Author



Campus Size Per Group



Campus Commits by Hour

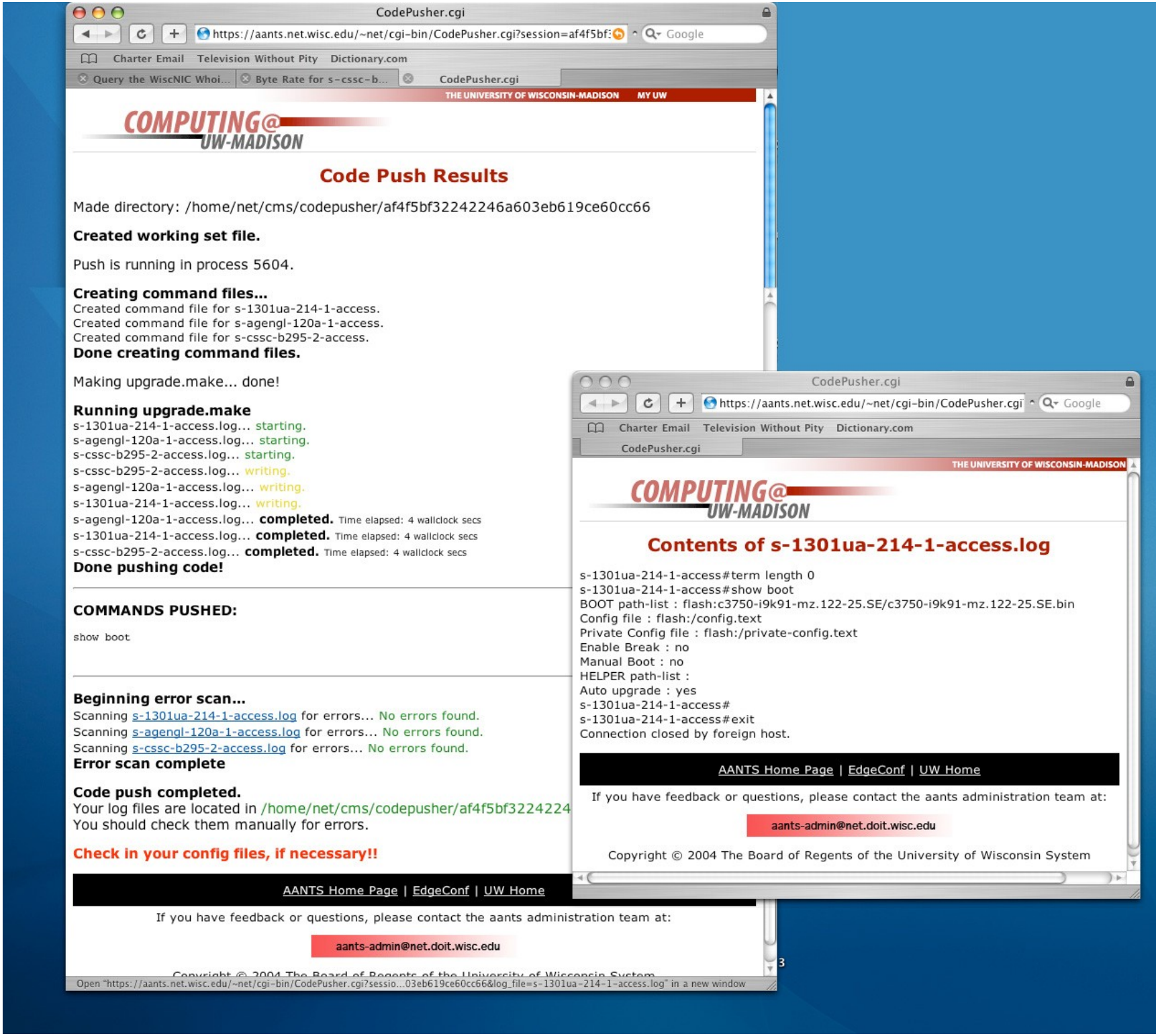


Common Commit Comments

Comment	Frequency
Initial revision	1487 (2.7%)
test	812 (1.5%)
asdf	593 (1.1%)
'newer bulk checkin'	411 (0.7%)
change vlan	316 (0.6%)

An Anomaly

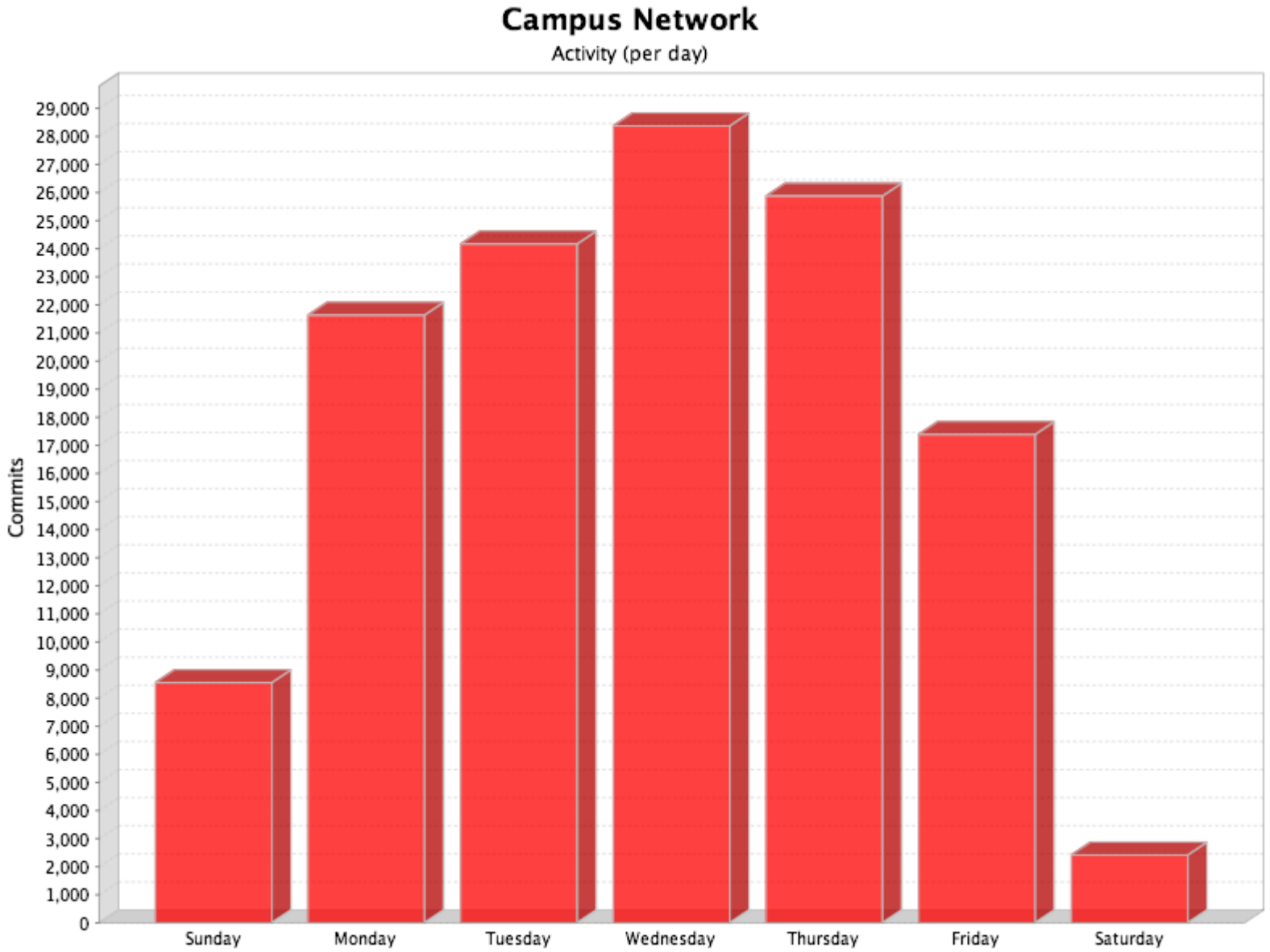
Author	Revisions	Lines of Code	Added Lines of Code	Lines of Code per Change
net	63468 (47.2%)	2418758 (82.9%)	3313853 (74.1%)	38.11
authorized-agents	38625 (28.8%)	1821 (0.1%)	208956 (4.7%)	0.05
system	11218 (8.4%)	-8795 (-0.3%)	125618 (2.8%)	-0.78
noc	10715 (8.0%)	100099 (3.4%)	303481 (6.8%)	9.34
field	6122 (4.6%)	57582 (2.0%)	152498 (3.4%)	9.41
contract	3959 (2.9%)	348207 (11.9%)	368518 (8.2%)	87.95
security	230 (0.2%)	103 (0.0%)	1898 (0.0%)	0.45


CodePusher.cgi
https://aants.net.wisc.edu/~net/cgi-bin/CodePusher.cgi?session=af4f5bf
Google
Charter Email
Television Without Pity
Dictionary.com
Query the WiscNIC Whoi...
Byte Rate for s-cssc-b...
CodePusher.cgi
THE UNIVERSITY OF WISCONSIN-MADISON
MY UW
COMPUTING@ UW-MADISON
Code Push Results
Made directory: /home/net/cms/codepusher/af4f5bf32242246a603eb619ce60cc66
Created working set file.
Push is running in process 5604.
Creating command files...
Created command file for s-1301ua-214-1-access.
Created command file for s-agengl-120a-1-access.
Created command file for s-cssc-b295-2-access.
Done creating command files.
Making upgrade.make... done!
Running upgrade.make
s-1301ua-214-1-access.log... starting.
s-agengl-120a-1-access.log... starting.
s-cssc-b295-2-access.log... starting.
s-cssc-b295-2-access.log... writing.
s-agengl-120a-1-access.log... writing.
s-1301ua-214-1-access.log... writing.
s-agengl-120a-1-access.log... completed. Time elapsed: 4 wallclock secs
s-1301ua-214-1-access.log... completed. Time elapsed: 4 wallclock secs
s-cssc-b295-2-access.log... completed. Time elapsed: 4 wallclock secs
Done pushing code!
COMMANDS PUSHED:
show boot
Beginning error scan...
Scanning s-1301ua-214-1-access.log for errors... No errors found.
Scanning s-agengl-120a-1-access.log for errors... No errors found.
Scanning s-cssc-b295-2-access.log for errors... No errors found.
Error scan complete
Code push completed.
Your log files are located in /home/net/cms/codepusher/af4f5bf3224224
You should check them manually for errors.
Check in your config files, if necessary!!
AANTS Home Page | EdgeConf | UW Home
If you have feedback or questions, please contact the aants administration team at:
aants-admin@net.doit.wisc.edu
Copyright © 2004 The Board of Regents of the University of Wisconsin System
Open "https://aants.net.wisc.edu/~net/cgi-bin/CodePusher.cgi?sessio...03eb619ce60cc66&log_file=s-1301ua-214-1-access.log" in a new window
CodePusher.cgi
https://aants.net.wisc.edu/~net/cgi-bin/CodePusher.cgi
Google
Charter Email
Television Without Pity
Dictionary.com
CodePusher.cgi
THE UNIVERSITY OF WISCONSIN-MADISON
COMPUTING@ UW-MADISON
Contents of s-1301ua-214-1-access.log
s-1301ua-214-1-access#term length 0
s-1301ua-214-1-access#show boot
BOOT path-list : flash:c3750-i9k91-mz.122-25.SE/c3750-i9k91-mz.122-25.SE.bin
Config file : flash:/config.text
Private Config file : flash:/private-config.text
Enable Break : no
Manual Boot : no
HELPER path-list :
Auto upgrade : yes
s-1301ua-214-1-access#
s-1301ua-214-1-access#exit
Connection closed by foreign host.
AANTS Home Page | EdgeConf | UW Home
If you have feedback or questions, please contact the aants administration team at:
aants-admin@net.doit.wisc.edu
Copyright © 2004 The Board of Regents of the University of Wisconsin System

Evaluating Practitioner Effort

- Measurements of practitioner effort
  - How often are “fixes” introduced?
  - How often do configurations change?
  - “Bad Days” (are Friday checkins more buggy?)
- Look toward improvements:
  - Syntax-aware revision analysis (stanzas)
  - How do we direct tool development?

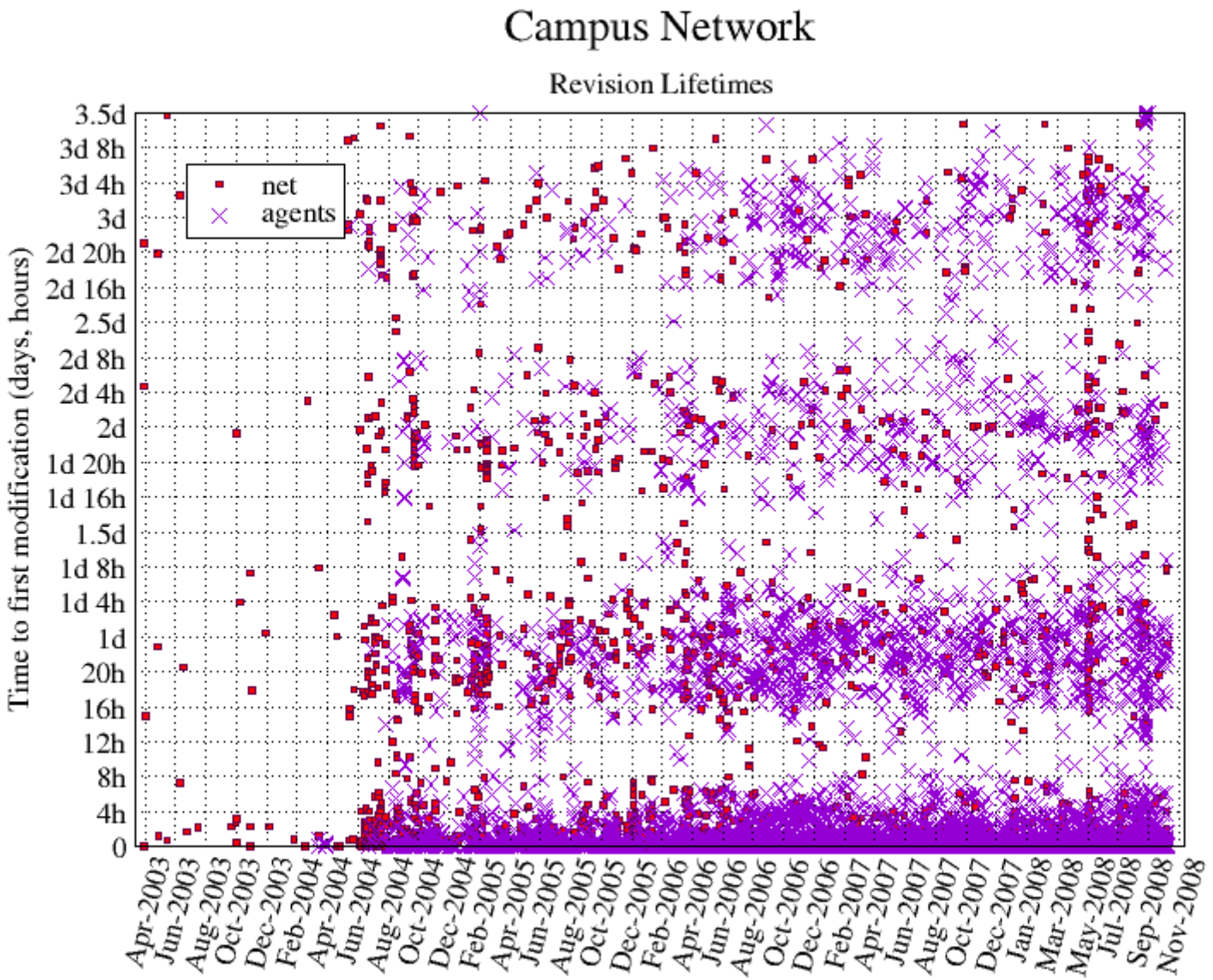
Campus Commits by Day



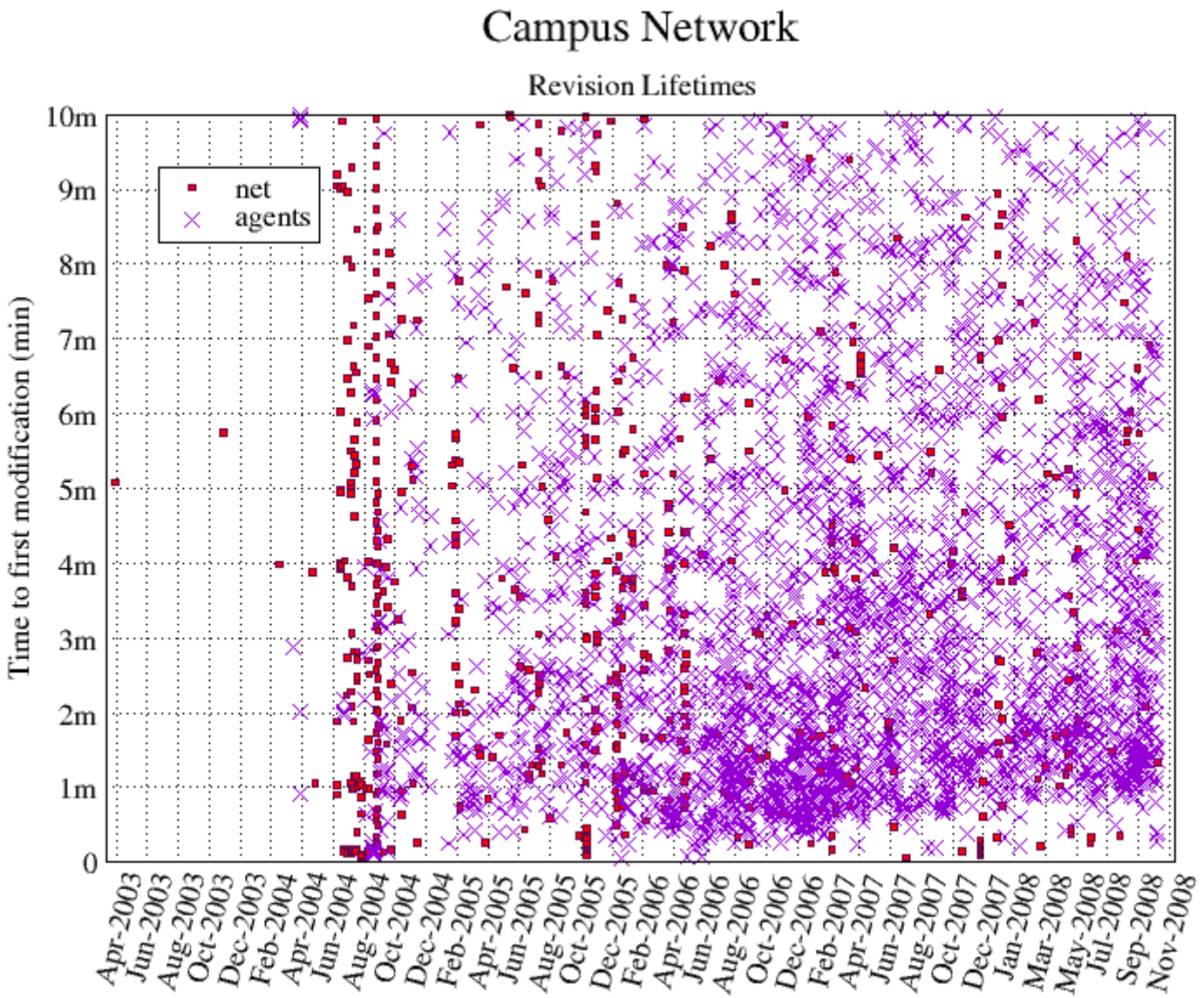
Revision Lifetimes

- How long does a revision last before it is next modified?
  - Suggests the modus operandi of practitioners
  - Suggests the value or the staying power of a revision
  - Might also suggest some measure of network volatility

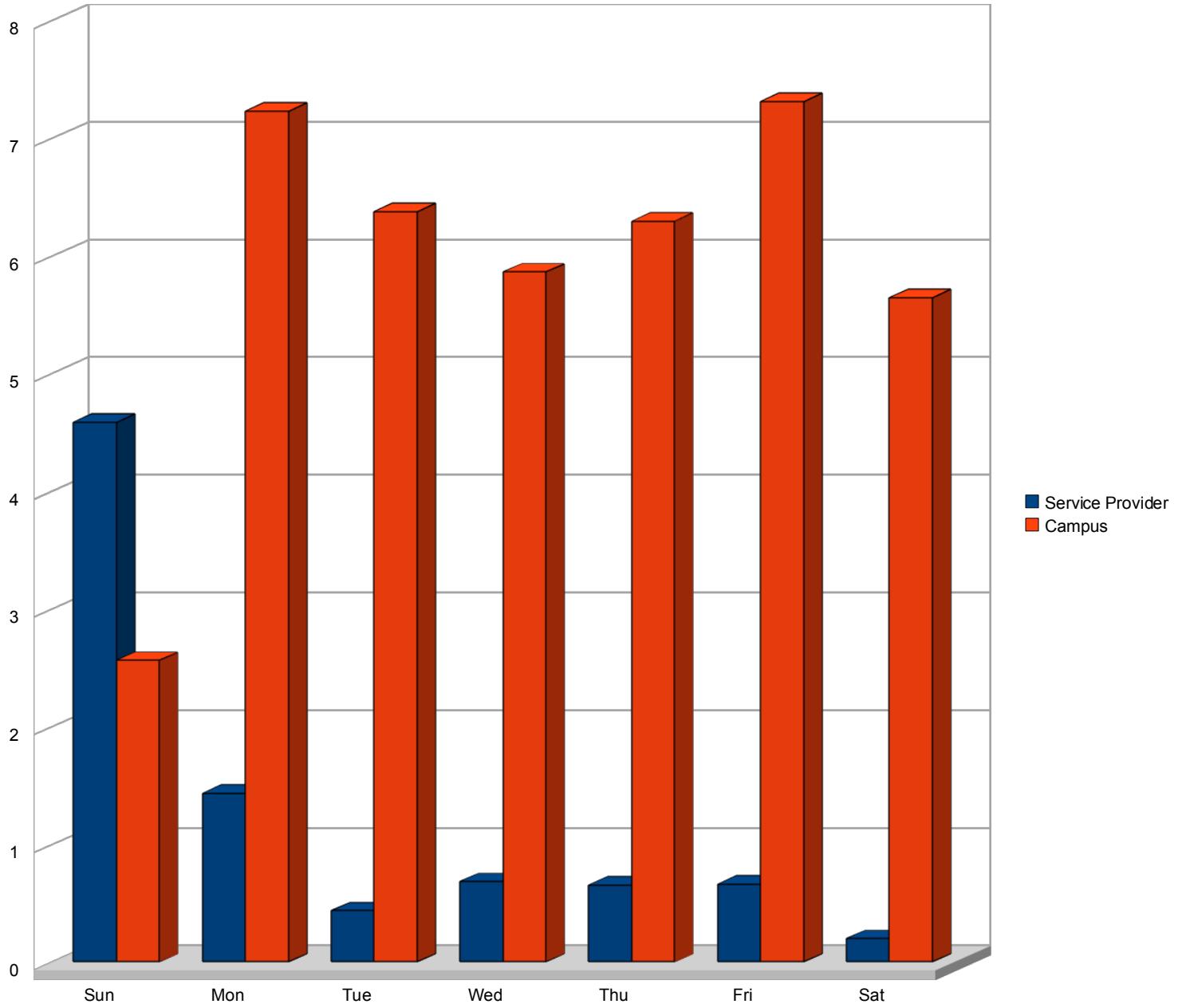
Campus Revision Lifetimes (<3.5 days)



Campus Revision Lifetimes (<10 min)

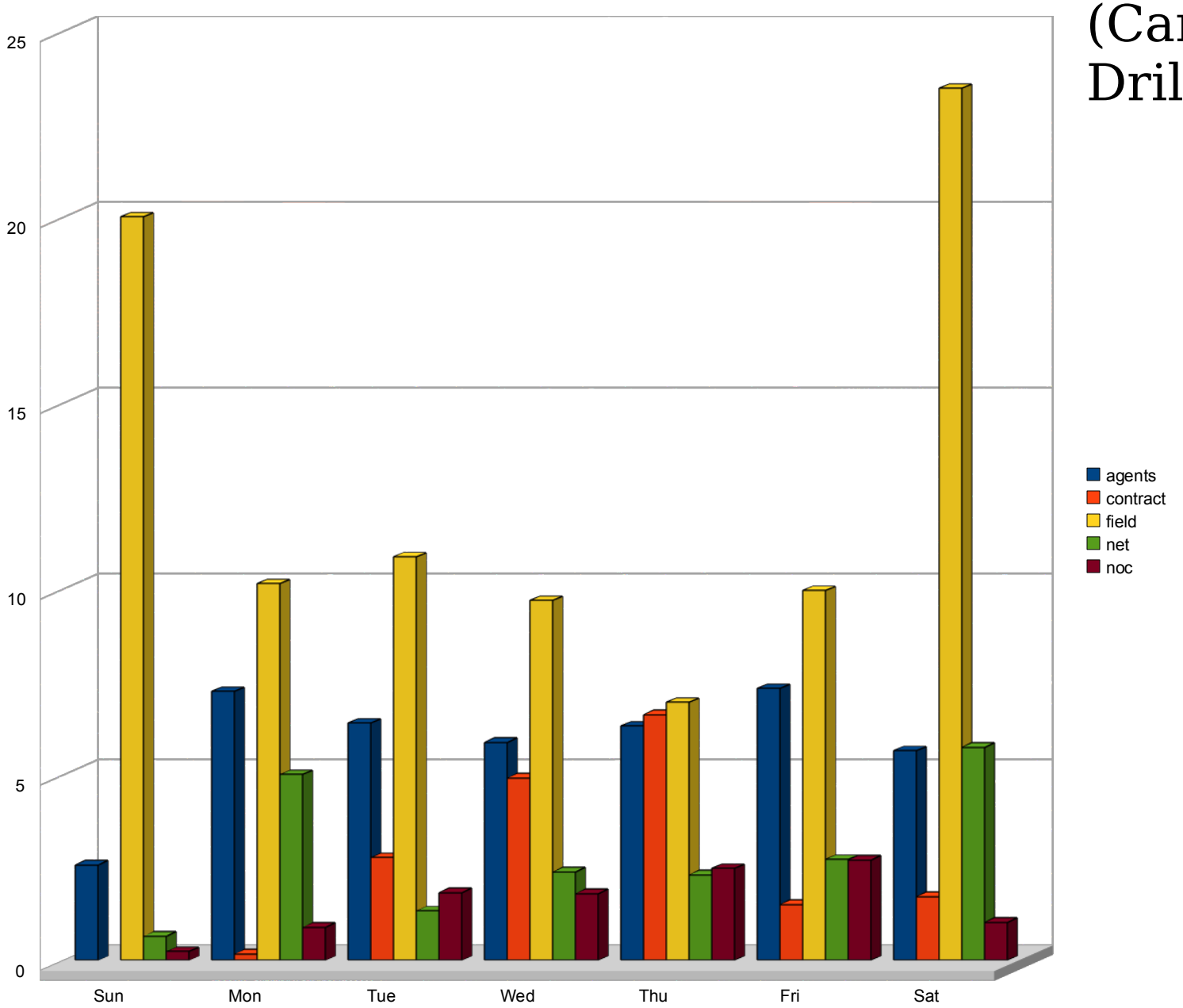


% Short-Lived Revisions by Day

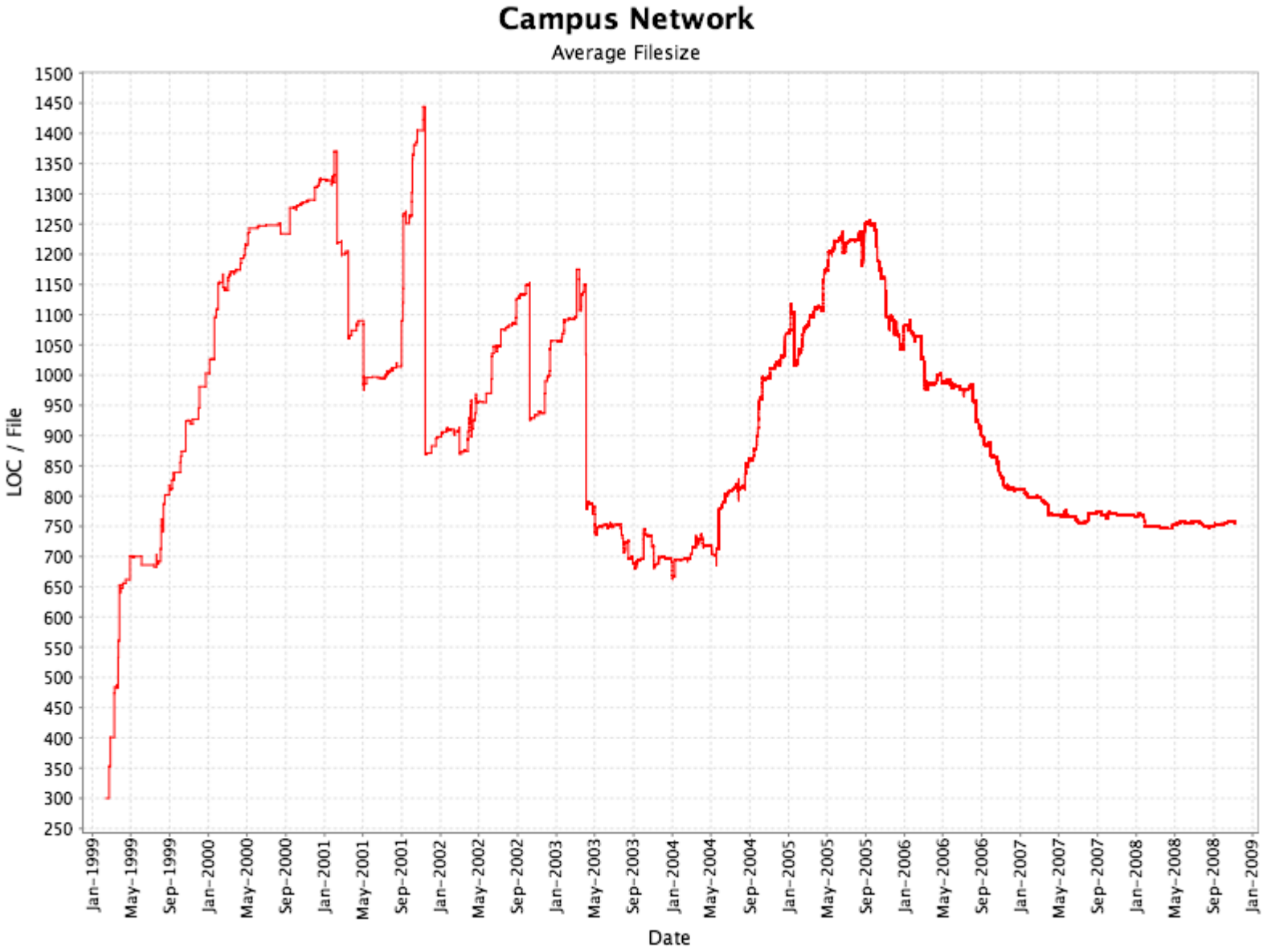


% Short-Lived Revisions by Day

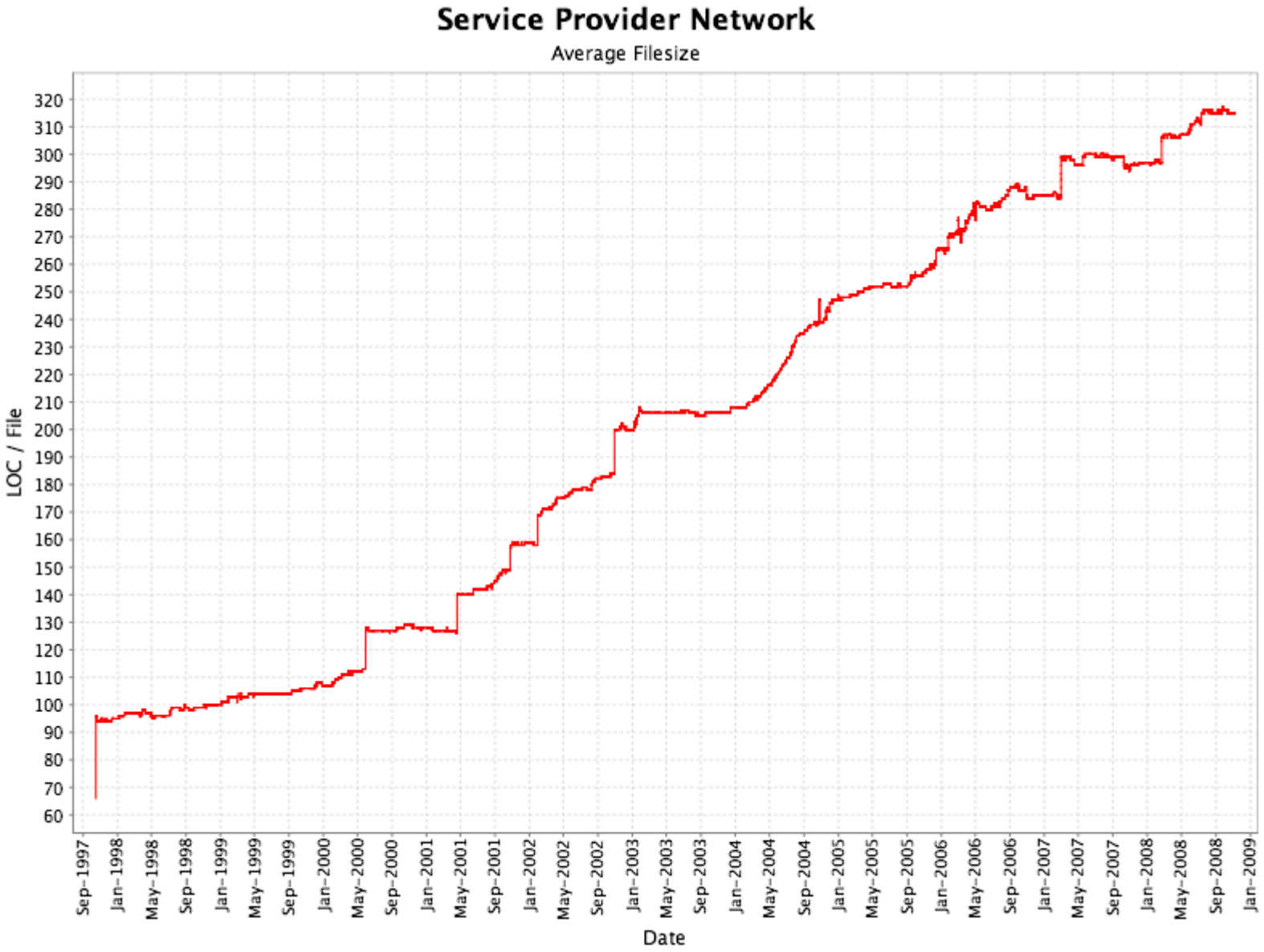
(Campus Drill-Down)



Campus Average File Size



Service Provider Average File Size



Campus Revisions by Stanza Type

Stanza Type	*Total Revision Count*	*Revisions per Instance*
interface	471,238	4
vlan	25,591	1
global	12,534	4
logging	12,390	9
ip	12,006	1
bridge	4,353	1
line	3,936	1
banner	3,810	1
dot11	3,324	1
control-plane	3,013	1

Some Conclusions

- With varying device types, LOC is an erratic metric for the stanza-based, declarative network configuration language, (such as Cisco IOS)
- Analysis of network configurations exposes pertinent network management details including:
  - Group behaviors
  - Outstanding practitioners
  - Change times
  - High level of user compliance, but some curiosities
  - Tool-based efficiencies both expected and invented

Contributions

- An initial application of software development analysis tools to network operations based on existing, freely-available tools
- Beginnings of a network operations-specific measurement of practitioner effort to guide tool development, such as SCM and IDE-like tools for network operators
- In our case studies, this analogy-based analysis approach shows promise based on feedback by expert interviews.

Discussion and Future Work

- As in software, can we identify and investigate code decay, refactorings, and code clones?
- Leverage other artifacts to measure practitioner compliance and network service reliability and performance.
- Develop a complexity metric based on stanzas and inter-stanza references. (see Benson, et al., NSDI 2009)

An Analysis of Network Configuration Artifacts

LISA '09, November 5, 2009

David Plonka & Andres Jaan Tack
{plonka,tack}@cs.wisc.edu