Networks 16: Other Applications

i.g.batten@bham.ac.uk

Applications

- TCP and below are complex, so that it’s easy to write applications
- IETF slogan “rough consensus and running code” means that applications tend to be proposed, implemented rapidly and then developed incrementally: grand architectures don’t achieve consensus.
- “Simple” is the watchword.

User Protocols

- FTP (RFC959) File Transfer
  - goes back to RFC114 April 1971, so very obviously not TCP-based
- SMTP (RFC5321) Simple Mail Transfer
  - goes back to RFC821 August 1982 — prior to that FTP was used for the purpose
- HTTP (RFC7230—7235) HyperText Transport
  - goes back to RFC2616 June 1999

User Protocols

- POP3 (RFC1939) Post Office
  - Simple, easy to implement
- IMAP4rev1 (RFC2060) Internet Message Access
  - Complex, Lisp-Like syntax, complete implementations are surprisingly rare

User Protocols

- ssh (RFC 4251–4253) Secure Shell
  - Replaces telnet, rlogin, rsh, other things, all of which are very insecure and should never be used
- Permits remote login (pseudo-tty), remote command execution, remote copying
- Encrypted, various secure authentication mechanisms

File Sharing

- NFS (originally from Sun)
  - v2 RFC1094, v3 RFC1813, v4 RFC5661
  - Implementing from earlier standard difficult, as lots of Unix-semantics assumptions
- SMB aka CIFS (originally from Microsoft)
  - **Not standardised**, lots of reverse engineering followed by proprietary documentation which may or may not be accurate.

Infrastructure

- DNS (lectures *passim*)
- SNMP Simple Network Management
  - Not Simple At All
- (S)NTP (Simple) Time

Conventions…

- Commands are case insensitive
    - “Four character” convention on older protocols (on mainframes, comparison of four-character strings is particularly efficient)
- Lines are terminated with \r\n, \015\012, control-m control-j, carriage return line feed, because Elvis is still alive.
- Responses often consist of a three-digit code followed by explanatory text
- SMTP is the best exemplar

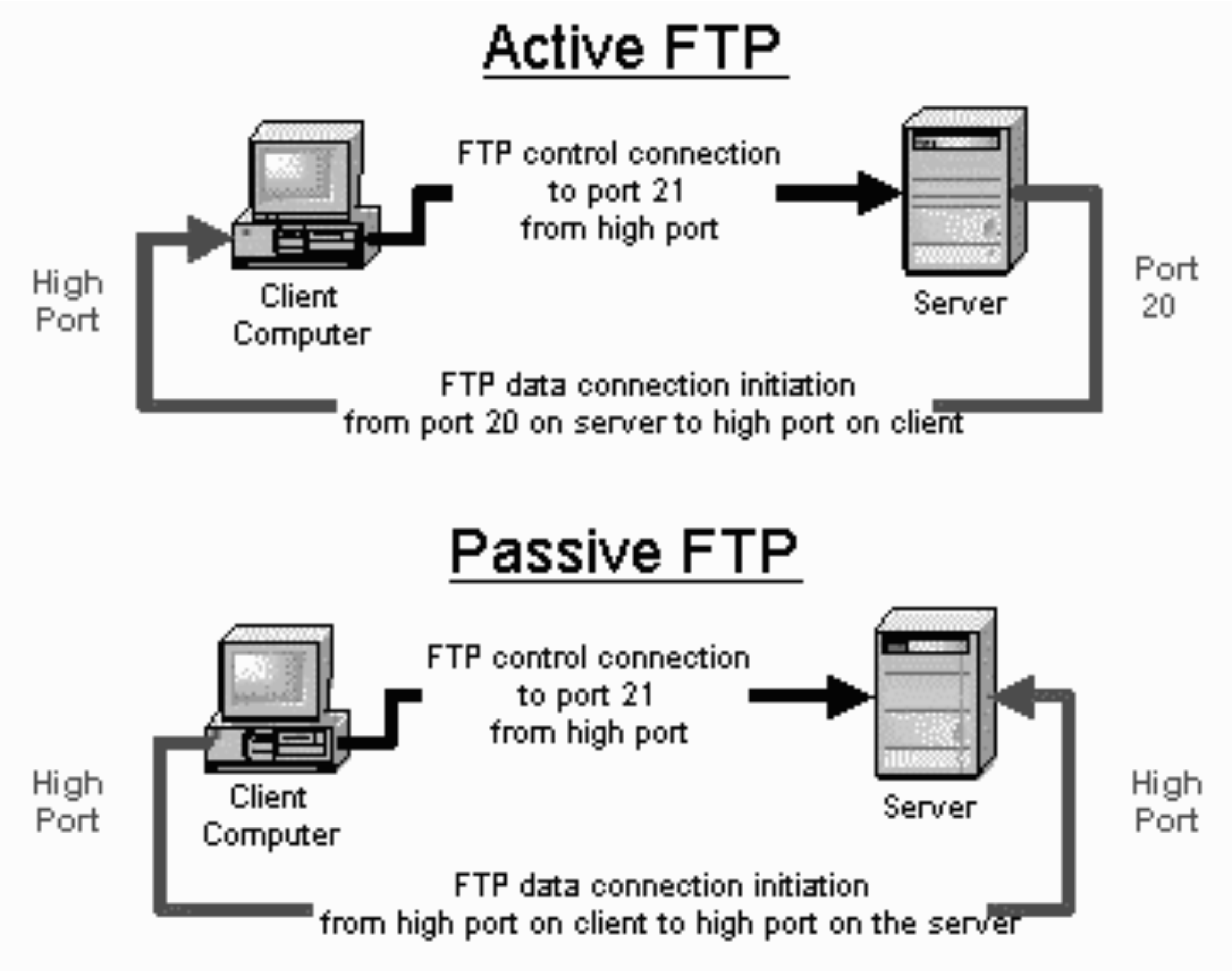
…there to be broken

- POP3 returns result codes in the first character of the response
- IMAP4 uses “tags” to match responses to requests
- And so on

FTP

- Very old protocol (first RFC nearly 45 years ago)
- Requires extensive support to pass through NAT, Firewalls, etc
- Large and complex server (difficult to audit and secure) supporting lots of conversion modes for TOPS-20, VMS, VM/CMS, block, record, stream…

FTP


Active FTP
FTP control connection
to port 21
from high port
High
Port
Client
Computer
Server
Port
20
FTP data connection initiation
from port 20 on server to high port on client
Passive FTP
FTP control connection
to port 21
from high port
High
Port
Client
Computer
Server
High
Port
FTP data connection initiation
from high port on client to high port on the server

Mechanism

- Control connection goes to port 21 on server
- When transferring files, CLIENT listen()s on a high numbered port, tells the server where it is, and the server calls back with a source port of 20
- This is all NCP-style, from the 1970s

New Mechanism

- PASV (Passive)
- Now the server listen()s on a high port, and the client calls to it from a high port
- Still requires firewall / NAT support, because high port to high port
- Firewall /NAT support involves snooping on the control connection

FTP: Avoid

- Doesn’t do anything that you both (a) need in 2018 and (b) you can’t do with something more modern (HTTP or scp)

- Firewall and NAT support for FTP is complex and arguably insecure: certainly open to abuse

- Probably not as true now, but a few years ago FTP NAT support was >50% of total LoC in a NAT implementation

    - Prediction: it’ll be used in some attack

- That there is an IPv6 profile for FTP is beyond all reason

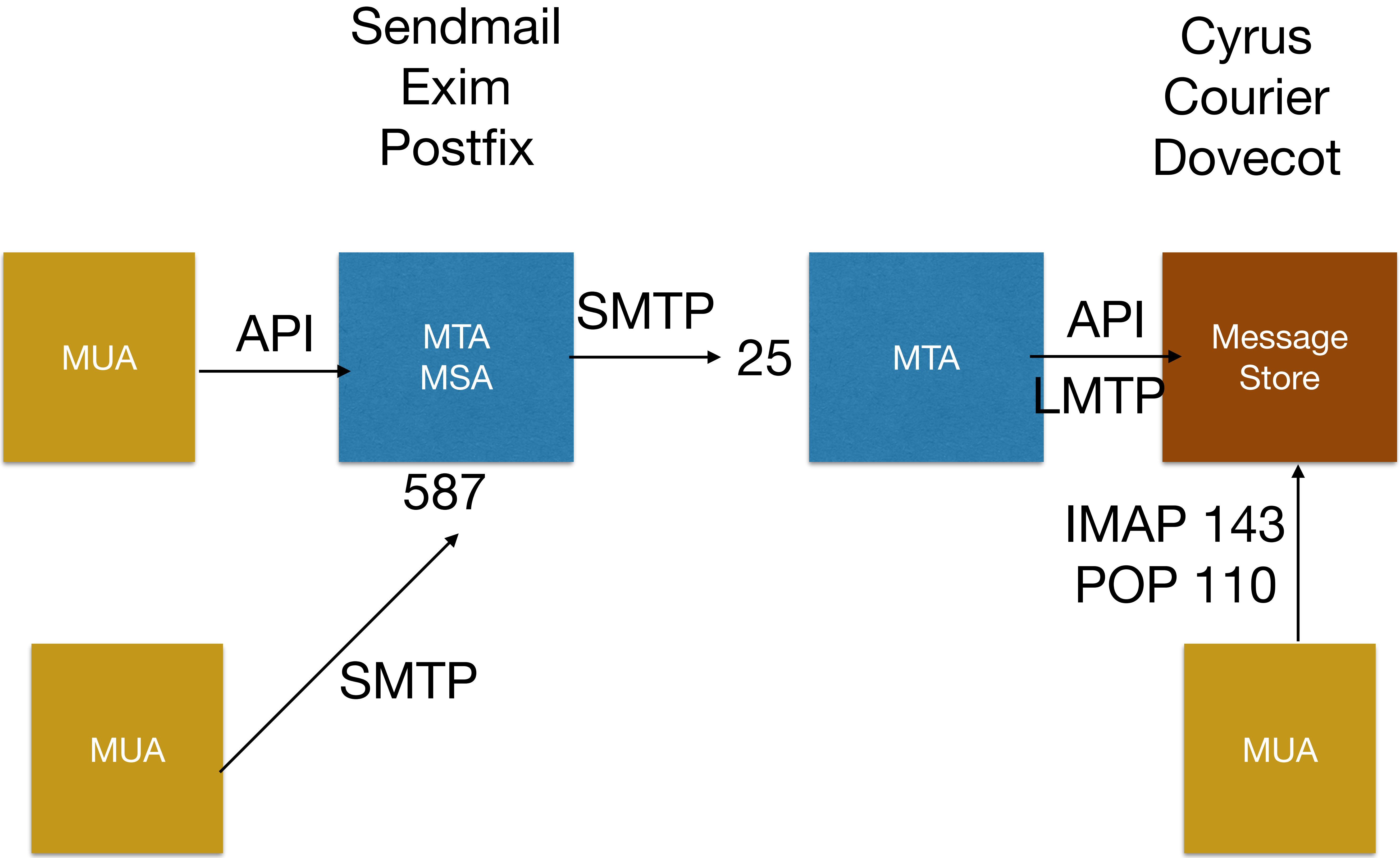
FTP Daemons

- System-shipped FTP daemons are usually frighteningly insecure
- ProFTPD is full featured but very complex to configure
- vsftpd is better and built for security
- As a minimum: drop root and chroot()

SMTP

- Handles sending of mail
- Has extensive extensions for authentication, encryption, etc
- Mail User Agents talk to Mail Transport Agents and Mail Submission Agents. MTAs talk to MTAs.

SMTP



Sendmail
Exim
Postfix

Cyrus
Courier
Dovecot

MUA

API

MTA
MSA

SMTP

25

MTA

API

LMTP

Message
Store

587

IMAP 143
POP 110

MUA

SMTP

MUA

SMTP

```
>>> EHLO research-1.batten.eu.org
250-mail.batten.eu.org Hello [IPv6:2001:630:c2:3263:8:20ff:fe89:b5a0], pleased to meet
250-ENHANCEDSTATUSCODES
(etc)
250 HELP
>>> MAIL From:<igb@research-1.batten.eu.org> SIZE=6
250 2.1.0 <igb@research-1.batten.eu.org>... Sender ok
>>> RCPT To:<igb@batten.eu.org>
>>> DATA
250 2.1.5 <igb@batten.eu.org>... Recipient ok
354 Enter mail, end with "." on a line by itself
>>> .
250 2.0.0 t257tb9n015843 Message accepted for delivery
igb@batten.eu.org... Sent (t257tb9n015843 Message accepted for delivery)
Closing connection to mail.batten.eu.org.
>>> QUIT
221 2.0.0 mail.batten.eu.org closing connection
```

SMTP

- HELO [sic]
- MAIL FROM
- RCPT TO
- DATA
- . (dot on a line on its own)
- QUIT

POP3

- Crude protocol for downloading email
- Connect, count messages, download
    - Assumes user has exactly one device to which they will be downloading the mail for further processing
- Can be grossly abused to provide sharing between devices, but this will end in tears. Always prefer…

IMAP

- Complex and full-featured protocol for access to mailboxes
- Can be used as POP for pure download if you want, but it is much better to leave your mail on the server for access from everywhere

IMAP

```
cmd1 login igb xxxpasswordxxx
cmd1 OK [CAPABILITY IMAP4rev1 LITERAL+ ID ENABLE ACL RIGHTS=kxte QUOTA MAILBOX-REFERRALS NAMESPACE UIDPLUS NO_ATOMIC_RENAME UNSELECT
CHILDREN MULTIAPPEND BINARY CATENATE CONDSTORE ESEARCH SORT SORT=MODSEQ SORT=DISPLAY THREAD=ORDEREDSUBJECT THREAD=REFERENCES ANNOTATEMORE
LIST-EXTENDED WITHIN QRESYNC SCAN XLIST URLAUTH URLAUTH=BINARY LOGINDISABLED AUTH=SCRAM-SHA-1 AUTH=DIGEST-MD5 AUTH=CRAM-MD5 AUTH=LOGIN
AUTH=PLAIN COMPRESS=DEFLATE IDLE] User logged in SESSIONID=<mail.batten.eu.org-15852-1425542394-1>
cmd2 examine INBOX
* 3236 EXISTS
* 0 RECENT
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen Junk NonJunk $MDNSent $NotJunk $Junk JunkRecorded $Forwarded NotJunk Forwarded Old
$MailFlagBit0 $MailFlagBit1 $MailFlagBit2 Redirected)
* OK [PERMANENTFLAGS ()] Ok
* OK [UNSEEN 2796] Ok
* OK [UIDVALIDITY 1371846329] Ok
* OK [UIDNEXT 118743] Ok
* OK [HIGHESTMODSEQ 42689] Ok
* OK [URLMECH INTERNAL] Ok
cmd2 OK [READ-ONLY] Completed
cmd3 fetch 2796 full
* 2796 FETCH (FLAGS ($NotJunk NotJunk) INTERNALDATE "21-Feb-2015 06:37:10 +0000" RFC822.SIZE 87547 ENVELOPE ("Sat, 21 Feb 2015 06:35:37
+0000" "@bbcquestiontime tweeted: Our most retweeted comment came from @NicolaSturgeon : catch up on @BBCiPlayer #bbcqt" (("Popular in you
network" NIL "info" "twitter.com")) (("Popular in your network" NIL "info" "twitter.com")) (("Popular in your network" NIL "info"
"twitter.com")) (("Batten TV Feed" NIL "tv" "batten.eu.org")) NIL NIL NIL "<E9.F1.35735.93728E45@twitter.com>") BODY (("TEXT"
"PLAIN" ("CHARSET" "UTF-8") NIL NIL "QUOTED-PRINTABLE" 1379 34)("TEXT" "HTML" ("CHARSET" "UTF-8") NIL NIL "QUOTED-PRINTABLE" 82934 1676)
"ALTERNATIVE"))
cmd3 OK Completed (0.000 sec)
cmd4 fetch 2796 body[header]
* 2796 FETCH (BODY[HEADER] {2909}
Return-Path: <b0487a57e25tv=batten.eu.org@bounce.twitter.com>
Received: from mail.batten.eu.org ([unix socket])
         by mail.batten.eu.org with LMTPA;
         Sat, 21 Feb 2015 06:37:10 +0000
X-Sieve: CMU Sieve 2.4
…
cmd4 OK Completed (0.000 sec)
```

IMAP

- Supports remote searching, which has proven to be very useful for phones although was originally intended for a very different model
  - Use “elm” while logged into someone else’s VAX
  - Author Marc Crispin is a big TOPS-20 man, so IMAP does not have quite the same Unix-centric feel of other protocols
- Also supports download messages by parts, which is useful on slow links

ssh

- Very complex commercial and licensing history, but now shipped on every *nix.
- Windows versions (PuTTY best known) widely available.
- Encrypted transport layer more suited to small messages (ie, character-by-character typing) than TLS
- Various authentication mechanisms (password, public key, certificates)
- Well audited, so far robust against attacks

NFS

- Ian’s favourite!
- Developed by Sun for access to filesystems over ethernet
  - Diskless workstations
  - Dataless workstations

NFS / ONC

- Part of suite called “ONC”, Open Network Computing.
- Included NIS, aka YP, for access to password files, host databases where DNS not available, “all the stuff usually in /etc”.
- Built on RPC (remote procedure call, Nelson and Birrell) and XDR (external data representation)

RPC / XDR

- Mechanism to perform procedure calls remotely
- Arguments are encoded with XDR to deal with cross-platform byte order, float precision, padding, alignment, etc.
- Servers written as daemons and register with the portmapper
- Clients contact the port mapper (port 111) to find port for version x of service y, then make procedure calls
- Given technology is 30+ years old, works surprisingly well.

NFS

- Standard Unix filesystem semantics, can be bodged to support other things
  - close() can error on disk-full conditions, still causing pain thirty years on
- Originally done over UDP for performance reasons, although RPC always supported TCP.
- Now almost always done over TCP

NFS

- v2: create, statfs, getattr, link, lookup, mkdir, null, read, readdir, readlink, rename, remove, rmdir, root (unused), setattr, symlink, wrcache (unused), write
- v3 adds: access, mknod, readdirplus, fsinfo, pathconf, commit

NFS Statelessness

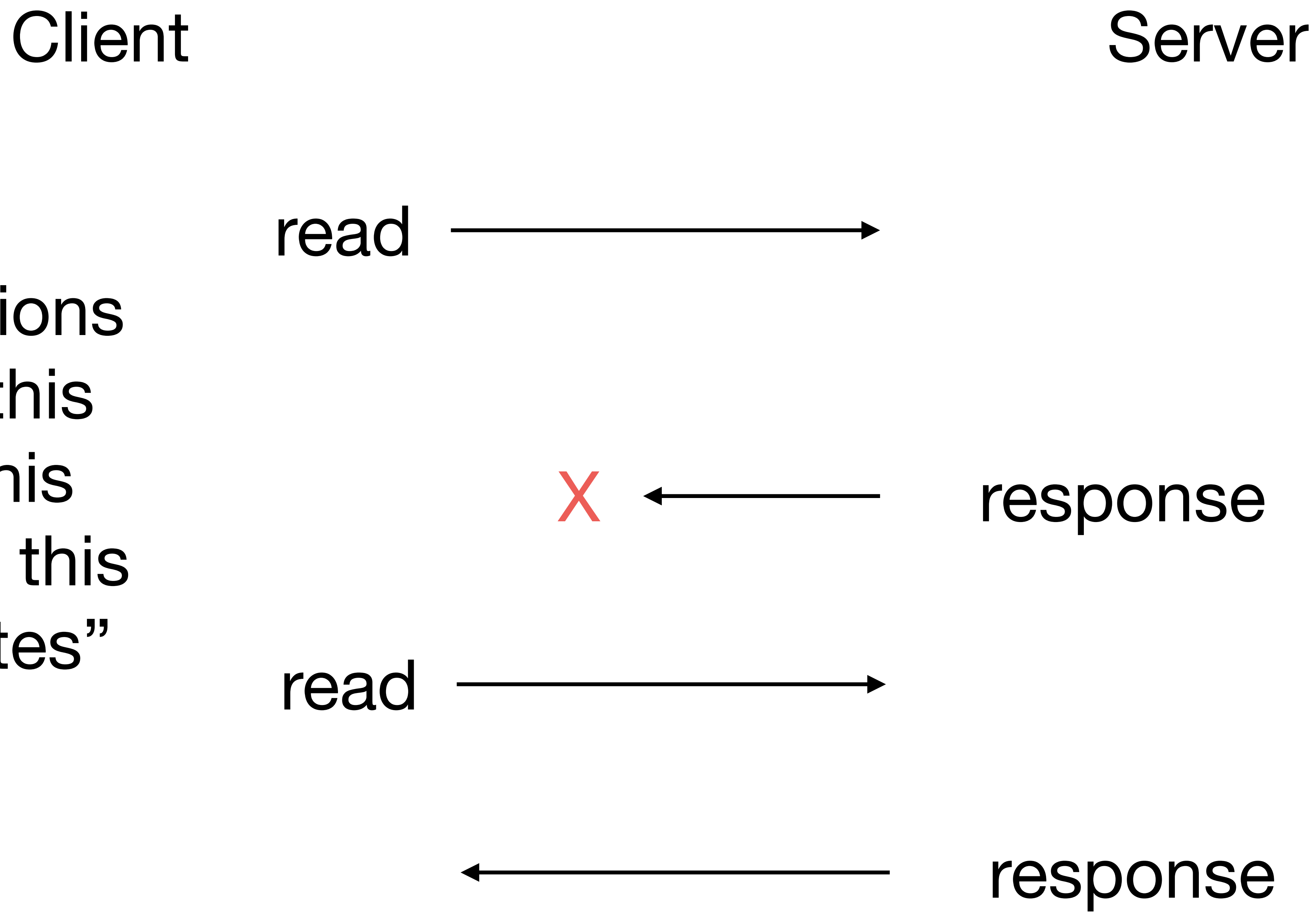
- Plan is to survive server crash and restart without needed (much) special handling on the client, and to avoid needing book-keeping on the server to track clients
  - Server notes when a client mounts a filesystem, but it’s not used for anything serious.
- Because initially done over UDP, responses to operations that have in fact been completed can be lost silently.
- TCP doesn’t fix this for persistent connections, as TCP doesn’t have synchronisation points
  - And roll-back of filesystem operations very difficult and invasive
- Idea is that most operations can safely be repeated and get the same result (repeated reads, repeated writes)
- Causes various complexities in implementation, because some operations are inherently stateful (although they are rarer and less performance critical)

Mount

- Mount involves sending requested pathname to server, returns “file handle” which encodes device number and other bookkeeping information

- No state on server: the filehandle is all you need, so you don’t need to remount after server crash and restart, you just continue using filehandles

Repeated Read OK



Client

Server

read

X

response

read

response

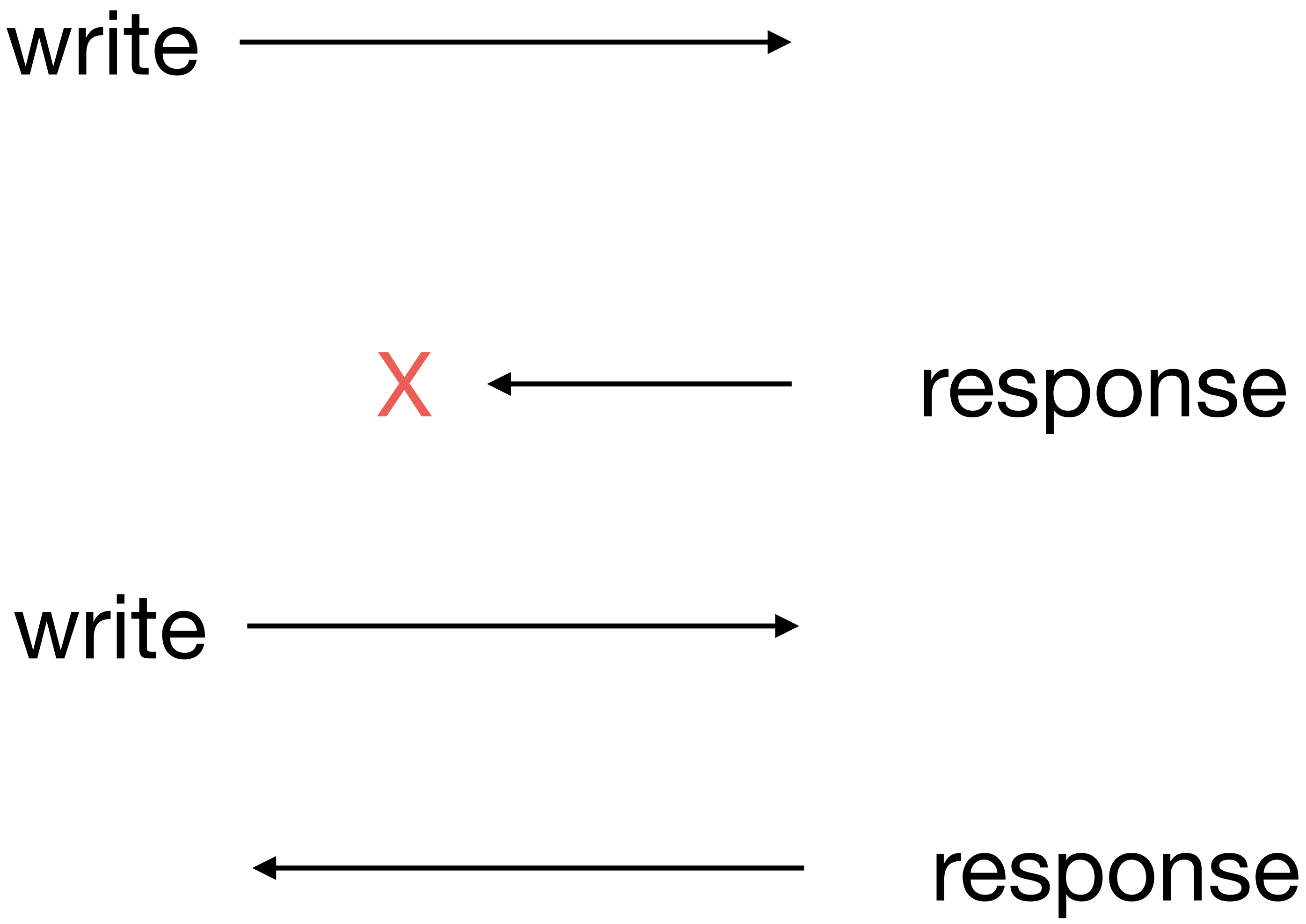
all operations are “on this file, at this offset, for this many bytes”

Repeated Write OK

Client

Server

all operations are “on this file, at this offset, for this many bytes”



write

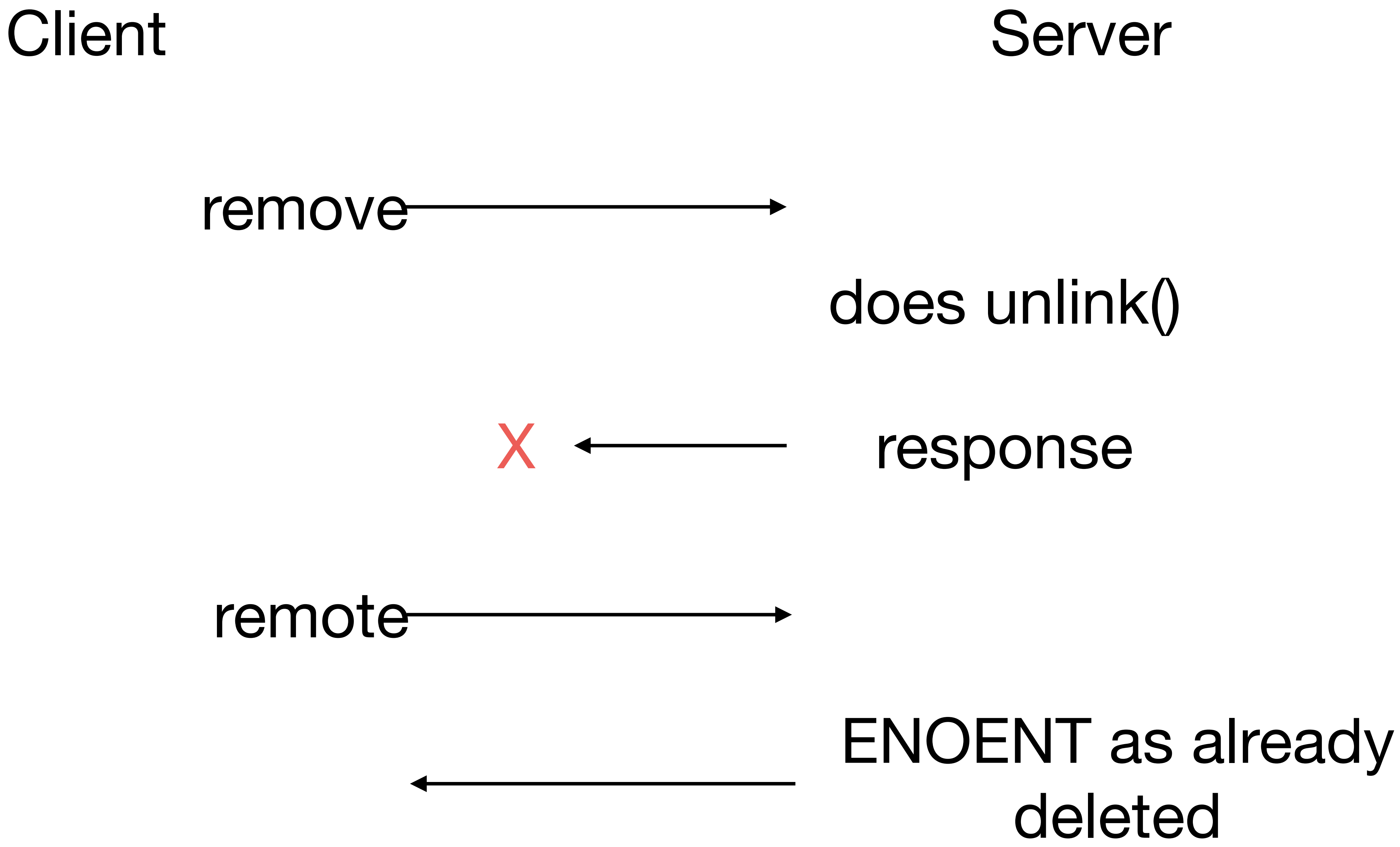
X

response

write

response

Repeated Remove Not OK



Client

Server

remove

does unlink()

X

response

remote

ENOENT as already deleted

Why does TCP not fix this?

- You only find out that a TCP segment wasn’t delivered to the other end when you call close() or shutdown() and get an error

- Some of the more complex OSI transport layers included “synchronisation points” to address this issue, where you can write a marker and get confirmation marker was received

  - Requires tricky API extensions to do this without blocking

Idempotency Cache

- Server caches the responses it previous sent to stateful operations (also mknod, create, etc)
- If it receives a repeat request, it sends response from cache
- Fun ensues in the sequence:
    - remove() (etc)
    - server does unlink() and then crashes
    - server restart with empty idempotency cache
    - Client resends remove()

NFS performance

- write() insists that data has gone to stable storage
- Big market in “Prestoserve” NVRAM caches
    - Auspex and others made a business out of it
- NFSv3 has “commit” so you can send data which you don’t demand is written, and then commit as appropriate
    - Big NVRAM caches were big business, Pillar, ZFS Appliances, NetApp, etc all survived until recently (NetApp still going, Pillar sort-of still going)
    - SSD means need for specialist fileservers somewhat diminished

NFS Performance

- Good NFS servers faster than local disk (<3ms write common, difficult to match with rotating disk)
  - Writing to/reading from exotic 10-way stripes with large NV/RAM caches in front of them via GigE
- Good NFS servers easier to manage and provide good facilities for backup, thin provisioning, etc
  - Like having an EMC without needing expensive Fibre Channel infrastructure, and giving sharing as well
- SSD changes the equations, jury still out on best approaches, but hard to see how network filesystems can compete with latency of SSD

NFS Locking

- Locking (of files or regions of files) is inherently stateful
- NFS cheats (no!) by claiming locking isn’t part of the file serving protocol and bolting a separate locking protocol on the side
- Client crashes now a problem: lock file, crash, what happens?
- Often subtly broken outside homogenous environments, and always has been. Auspex cheated by using Sun code on embedded Solaris processor, and when they moved to their own code it broke. Linux, NetApp and Pillar have their own implementations which have varying problems

SMB

- Similar to NFS, but stateful (clients have to remount after server crash, server knows who has what mounted and what open).

- Because stateful, incorporates locking model and better caching model.

- SMB harder to do very quickly, but (it pains me to say) probably easier to use correctly.

SMB v CIFS

- SMB: implementations other than Microsoft's own can be temperamental, but most of them have had the bugs worked out by now

- Driving NFS out of the marketplace because there's a lot more Windows than there is Unix: NFS not shipped as standard on any Windows build.

- Almost all servers now "bi-lingual": NetApp, Oracle ex-Pillar Axioms, Oracle ex-Sun ZFS Appliances, etc.

- Apple moving to SMB as well (previously supported NFS and AFS)

SNMP

- Simple Network Management Protocol
- Response to various OSI initiatives, notably CMIP
  - Common Management Information Protocol: frighteningly complex, running over an OSI session layer that was completely impractical on embedded devices
  - Even CMOT ("CMIP over TCP") is impractical
- The *Simple* is relative to CMIP: it's actually rather complex.
- Much more formality than the typical Internet Protocol

SNMP

- Intention was to use to manage switches, routers and other hardware, including configuration
- In reality, 90% of usage is monitoring, 9% is fault reporting, 1% (if that) is configuration.
- *Agents* sit on equipment and answer requests, *managers* or *clients* make the requests.
- Agents can also raise asynchronous alerts (“traps”) when they act as *trap sources* and send them to *trap sinks*.

SNMP Concepts

- Data is grouped into a MIB, Management Information Base.

- The contents are described using a subset of ASN1, the Abstract Syntax Notation from OSI.

- Data is encoded on the wire using BER, Basic Encoding Rules, again from OSI.

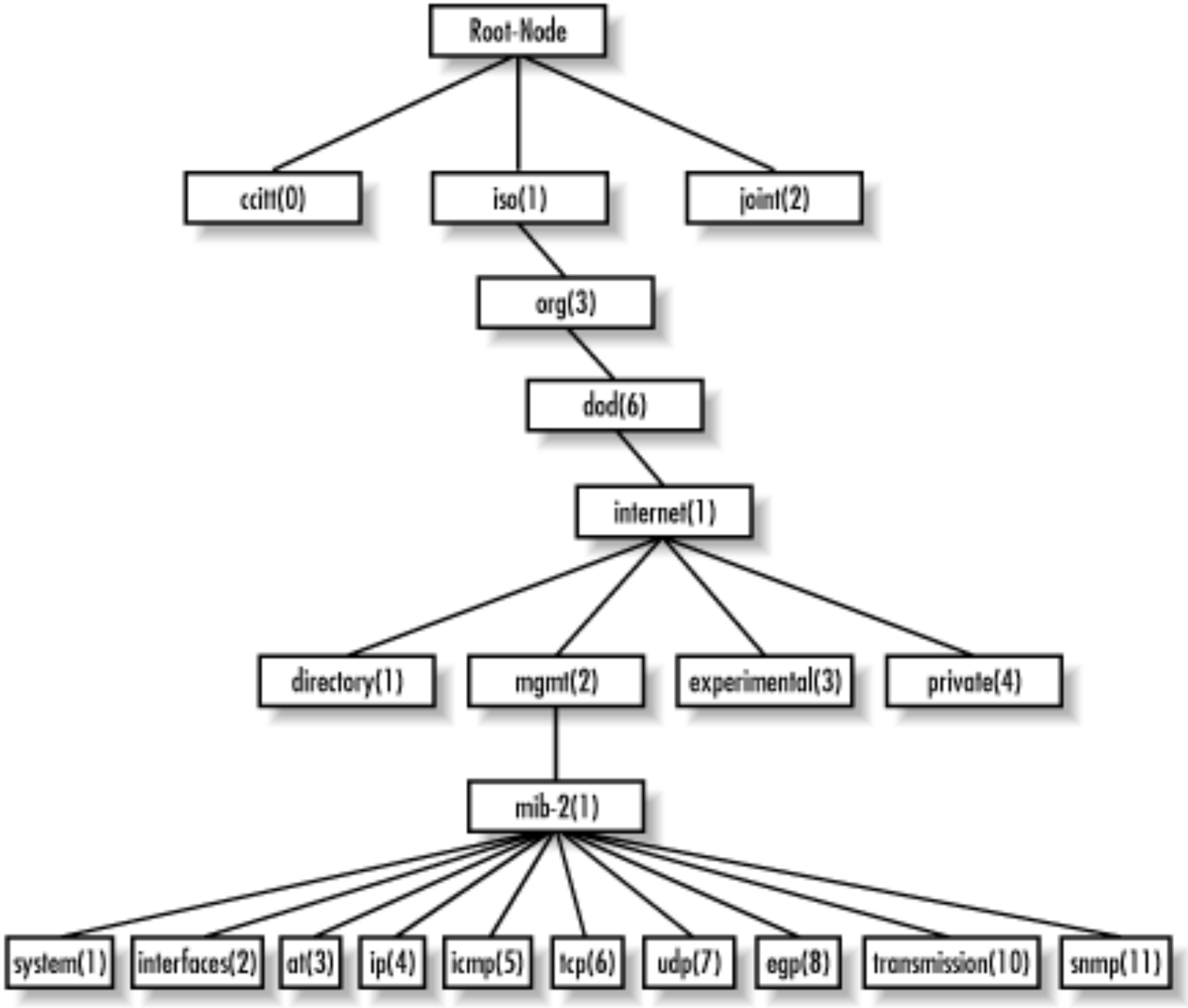
MIBs

```
IfEntry ::=
    SEQUENCE {
        ifIndex                 InterfaceIndex,
        ifDescr                 DisplayString,
        ifType                  IANAifType,
        ifMtu                   Integer32,
        ifSpeed                 Gauge32,
        ifPhysAddress           PhysAddress,
        ifAdminStatus           INTEGER,
        ifOperStatus            INTEGER,
        ifLastChange            TimeTicks,
        ifInOctets              Counter32,
        ifInUcastPkts           Counter32,
        ifInNUcastPkts          Counter32,  -- deprecated
        ifInDiscards            Counter32,
        ifInErrors              Counter32,
        ifInUnknownProtos       Counter32,
        ifOutOctets             Counter32,
        ifOutUcastPkts          Counter32,
        ifOutNUcastPkts         Counter32,  -- deprecated
        ifOutDiscards           Counter32,
        ifOutErrors             Counter32,
        ifOutQLen               Gauge32,    -- deprecated
        ifSpecific              OBJECT IDENTIFIER -- deprecated
    }
```

IF-MIB::OutOctets

```
[igb@offsite7 ~]$ snmpbulkwalk  -v3 -u cacticacti -l authpriv -X ‘-deleted-‘ -A ‘-deleted-‘ \
    udp6:rb2011-1.batten.eu.org IF-MIB::ifOutOctets
IF-MIB::ifOutOctets.1 = Counter32: 0
IF-MIB::ifOutOctets.2 = Counter32: 0
IF-MIB::ifOutOctets.3 = Counter32: 0
IF-MIB::ifOutOctets.4 = Counter32: 997357259
IF-MIB::ifOutOctets.5 = Counter32: 890378998
IF-MIB::ifOutOctets.6 = Counter32: 0
IF-MIB::ifOutOctets.7 = Counter32: 18064840
IF-MIB::ifOutOctets.8 = Counter32: 70886570
IF-MIB::ifOutOctets.9 = Counter32: 61591757
IF-MIB::ifOutOctets.10 = Counter32: 55158595
IF-MIB::ifOutOctets.11 = Counter32: 318834666
IF-MIB::ifOutOctets.17 = Counter32: 93797603
IF-MIB::ifOutOctets.18 = Counter32: 47336681
IF-MIB::ifOutOctets.20 = Counter32: 2673743280
IF-MIB::ifOutOctets.24 = Counter32: 47324115
IF-MIB::ifOutOctets.26 = Counter32: 10708521
IF-MIB::ifOutOctets.15728640 = Counter32: 3100226
[igb@offsite7 ~]$
```

MIB Structure



Numeric OIDS

```
[igb@offsite7 ~]$ snmpbulkwalk -On  -v3 -u cacticacti -l authpriv \
   -X ‘-deleted-‘ -A ‘-deleted-‘  udp6:rb2011-1.batten.eu.org IF-MIB::ifOutOctets
.1.3.6.1.2.1.2.2.1.16.1 = Counter32: 0
.1.3.6.1.2.1.2.2.1.16.2 = Counter32: 0
.1.3.6.1.2.1.2.2.1.16.3 = Counter32: 0
.1.3.6.1.2.1.2.2.1.16.4 = Counter32: 997366016
.1.3.6.1.2.1.2.2.1.16.5 = Counter32: 890411588
.1.3.6.1.2.1.2.2.1.16.6 = Counter32: 0
.1.3.6.1.2.1.2.2.1.16.7 = Counter32: 18070148
.1.3.6.1.2.1.2.2.1.16.8 = Counter32: 70903630
.1.3.6.1.2.1.2.2.1.16.9 = Counter32: 61702815
.1.3.6.1.2.1.2.2.1.16.10 = Counter32: 55172277
.1.3.6.1.2.1.2.2.1.16.11 = Counter32: 318884246
.1.3.6.1.2.1.2.2.1.16.17 = Counter32: 93833332
.1.3.6.1.2.1.2.2.1.16.18 = Counter32: 47411302
.1.3.6.1.2.1.2.2.1.16.20 = Counter32: 2673780229
.1.3.6.1.2.1.2.2.1.16.24 = Counter32: 47398246
.1.3.6.1.2.1.2.2.1.16.26 = Counter32: 10714986
.1.3.6.1.2.1.2.2.1.16.15728640 = Counter32: 3106731
[igb@offsite7 ~]$
```

Descriptions

```
[igb@offsite7 ~]$ snmpbulkwalk … IF-MIB::ifDescr
.1.3.6.1.2.1.2.2.1.2.1 = STRING: sfp1-spare
.1.3.6.1.2.1.2.2.1.2.2 = STRING: ether1-imac
.1.3.6.1.2.1.2.2.1.2.3 = STRING: ether2-switch-master
.1.3.6.1.2.1.2.2.1.2.4 = STRING: ether3-airport
.1.3.6.1.2.1.2.2.1.2.5 = STRING: ether4-netgear-link
.1.3.6.1.2.1.2.2.1.2.6 = STRING: ether5-spare
.1.3.6.1.2.1.2.2.1.2.7 = STRING: ether6-printer
.1.3.6.1.2.1.2.2.1.2.8 = STRING: ether7-pi-one
.1.3.6.1.2.1.2.2.1.2.9 = STRING: ether8-pi-two
.1.3.6.1.2.1.2.2.1.2.10 = STRING: ether9-pi-three
.1.3.6.1.2.1.2.2.1.2.11 = STRING: ether10-dsl320b
.1.3.6.1.2.1.2.2.1.2.17 = STRING: pppoe-aa-uplink
.1.3.6.1.2.1.2.2.1.2.18 = STRING: bridge-vlan-5-redzone
.1.3.6.1.2.1.2.2.1.2.20 = STRING: bridge-default-vlan
.1.3.6.1.2.1.2.2.1.2.24 = STRING: vlan-5-pi-one
.1.3.6.1.2.1.2.2.1.2.26 = STRING: vlan-5-mac-mini
.1.3.6.1.2.1.2.2.1.2.15728640 = STRING: <l2tp-igb-l2tp>
[igb@offsite7 ~]$
```

SNMP Operations

- GET fetches a requested OID
- GETNEXT fetches the next OID after the requested one
- snmpwalk (and friends) can fetch entire MIB, or subtrees thereof, using GETNEXT
- GETBULK asks for a number of GETNEXT operations to be be performed and returns them all in one operation (not always available)
    - Absolutely fantastic for Amplification Attacks

Counters

- Interface statistics are provided as counters since restart, not as rates or counts in interval
    - No guarantee device has accurate clock
    - May be multiple clients
- 32-bit counters wrap around trivially (every 4GB)
- Heuristic is that if a counter has gone backwards it’s actually a wrap-around
    - If there’s a risk of wrap around inside polling interval, hardly difficult with 5 minute polling, you need 64 bit counters

SNMP Implementation

- Originally UDP, because of fear that TCP was too hard for embedded devices and TCP handshakes too inefficient
- Most implementations ended up being UDP only
- With hindsight, wrong decision: you need management information to be reliable
- SNMP now often available over TCP, which is preferred
  - SNMP Proxying allows TCP over WAN and then UDP over LAN to less-capable devices

SNMP Proxy

```
[igb@offsite7 ~]$ snmpbulkwalk  -v3 -u cacticacti -l authpriv \
   -X ‘-deleted-‘ -A ‘-deleted-‘  \
   -n gs108tv2-1 tcp6:snmp-proxy.batten.eu.org IF-MIB::ifOutOctets
IF-MIB::ifOutOctets.1 = Counter32: 2487668549
IF-MIB::ifOutOctets.2 = Counter32: 875703010
IF-MIB::ifOutOctets.3 = Counter32: 0
IF-MIB::ifOutOctets.4 = Counter32: 374489695
IF-MIB::ifOutOctets.5 = Counter32: 2888671020
IF-MIB::ifOutOctets.6 = Counter32: 1245556106
IF-MIB::ifOutOctets.7 = Counter32: 1879710128
IF-MIB::ifOutOctets.8 = Counter32: 4050225960
IF-MIB::ifOutOctets.13 = Counter32: 273726231
IF-MIB::ifOutOctets.14 = Counter32: 0
IF-MIB::ifOutOctets.15 = Counter32: 0
IF-MIB::ifOutOctets.16 = Counter32: 0
IF-MIB::ifOutOctets.17 = Counter32: 0
[igb@offsite7 ~]$
```

SNMP Security

- snmp v1, v2, v2c: absolute joke

- A “community string” is placed, in clear, in the packet. Any packet with the right community string is OK.

  - Killed use of SNMP for SET operations stone dead, and makes traps (UDP!) dangerous

  - Community string is usually “public”, and you rely on firewalls to provide some privacy

SNMP v2

```
mini-server:~ igb$ snmpget -v2c -c public \
  batten-eu-org-dual-band.home.batten.eu.org SNMPv2-MIB::sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: Apple AirPort - Apple Inc., 2006-2012.
mini-server:~ igb$
```

SNMPv2

```
09:49:24.465620 IP mini-server.home.batten.eu.org.49233 > batten-eu-org-dual-
band.home.batten.eu.org.snmp:
    GetRequest(28)  system.sysDescr.0
   0x0000:  0024 369e 04f6 0026 bb60 07ce 0800 4500  .$6....&.`....E.
   0x0010:  0047 ab68 0000 4011 0000 0a5c d5b1 0a5c  .G.h..@....\...\
   0x0020:  d59a c051 00a1 0033 c048 3029 0201 0104  ...Q...3.H0)....
   0x0030:  0670 7562 6c69 63a0 1c02 045b a9e9 cd02  .public....[....
   0x0040:  0100 0201 0030 0e30 0c06 082b 0601 0201  .....0.0...+....
   0x0050:  0101 0005 00                             .....
09:49:24.466352 IP batten-eu-org-dual-band.home.batten.eu.org.snmp > mini-server.home.batten.eu.org.
49233:
    GetResponse(88)  system.sysDescr.0="Apple AirPort - Apple Inc., 2006-2012.  All rights Reserved."
   0x0000:  0026 bb60 07ce 0024 369e 04f6 0800 4500  .&.`...$6.....E.
   0x0010:  0083 8d81 0000 4011 2ce5 0a5c d59a 0a5c  ......@.,..\...\
   0x0020:  d5b1 00a1 c051 006f 6f4e 3065 0201 0104  .....Q.ooN0e....
   0x0030:  0670 7562 6c69 63a2 5802 045b a9e9 cd02  .public.X..[....
   0x0040:  0100 0201 0030 4a30 4806 082b 0601 0201  .....0J0H..+....
   0x0050:  0101 0004 3c41 7070 6c65 2041 6972 506f  ....<Apple.AirPo
   0x0060:  7274 202d 2041 7070 6c65 2049 6e63 2e2c  rt.-.Apple.Inc.,
   0x0070:  2032 3030 362d 3230 3132 2e20 2041 6c6c  .2006-2012...All
   0x0080:  2072 6967 6874 7320 5265 7365 7276 6564  .rights.Reserved
   0x0090:  2e
```

SNMPv2

- Clearly unacceptable for use in WAN environments
- Lack of error reporting in most agents plus lack of rate limiting means community-string guessing is also a reasonable attack
- Exposes data useful to attacker (descriptions, topology)

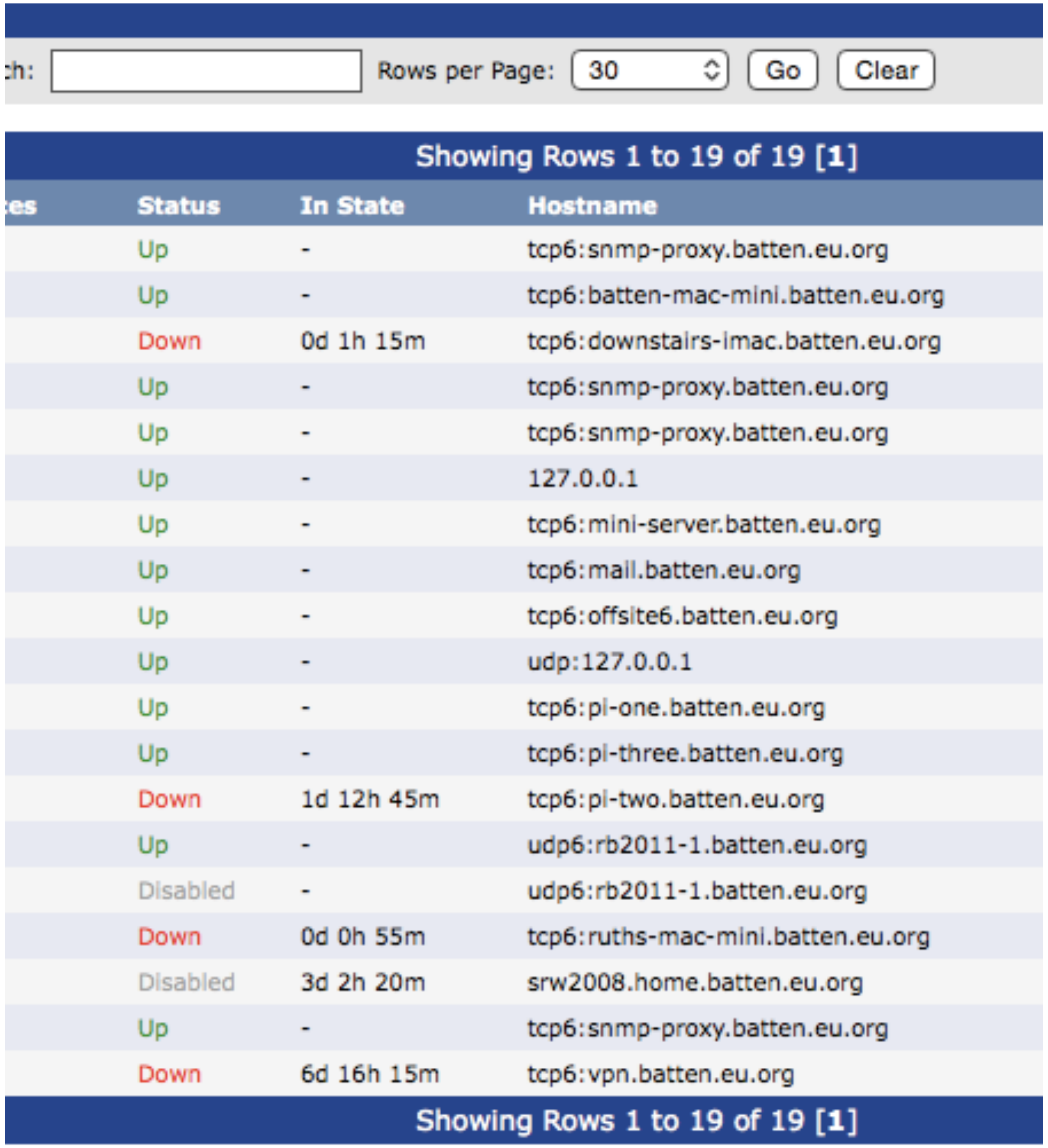
SNMPv3 security

- Multiple usernames with different views and access, “contexts”, etc.
- Authentication and encryption
- Authentication:
    - Insert string into packet, hash the whole packet, replace where the string was with the hash
    - Encryption using pre-shared key
- Also now available over TLS

Sadly…

- SNMPv3 is tricky to set up, as a result
    - Not all devices support it (depressingly common to only have SNMPv2 over UDP IPv4, cf. Apple Airports, Netgear switches, etc)
    - Alternatively, it's there, but it's buggy, and it's CLI-only with the GUI only doing v2
    - Not easy manager side, either
- Even systems that do support v3 end up being used with SNMPv2.

Lots of broken implemenations



Showing Rows 1 to 19 of 19 [1]

es	Status	In State	Hostname
	Up	-	tcp6:snmp-proxy.batten.eu.org
	Up	-	tcp6:batten-mac-mini.batten.eu.org
	Down	0d 1h 15m	tcp6:downstairs-imac.batten.eu.org
	Up	-	tcp6:snmp-proxy.batten.eu.org
	Up	-	tcp6:snmp-proxy.batten.eu.org
	Up	-	127.0.0.1
	Up	-	tcp6:mini-server.batten.eu.org
	Up	-	tcp6:mail.batten.eu.org
	Up	-	tcp6:offsite6.batten.eu.org
	Up	-	udp:127.0.0.1
	Up	-	tcp6:pi-one.batten.eu.org
	Up	-	tcp6:pi-three.batten.eu.org
	Down	1d 12h 45m	tcp6:pi-two.batten.eu.org
	Up	-	udp6:rb2011-1.batten.eu.org
	Disabled	-	udp6:rb2011-1.batten.eu.org
	Down	0d 0h 55m	tcp6:ruths-mac-mini.batten.eu.org
	Disabled	3d 2h 20m	srw2008.home.batten.eu.org
	Up	-	tcp6:snmp-proxy.batten.eu.org
	Down	6d 16h 15m	tcp6:vpn.batten.eu.org

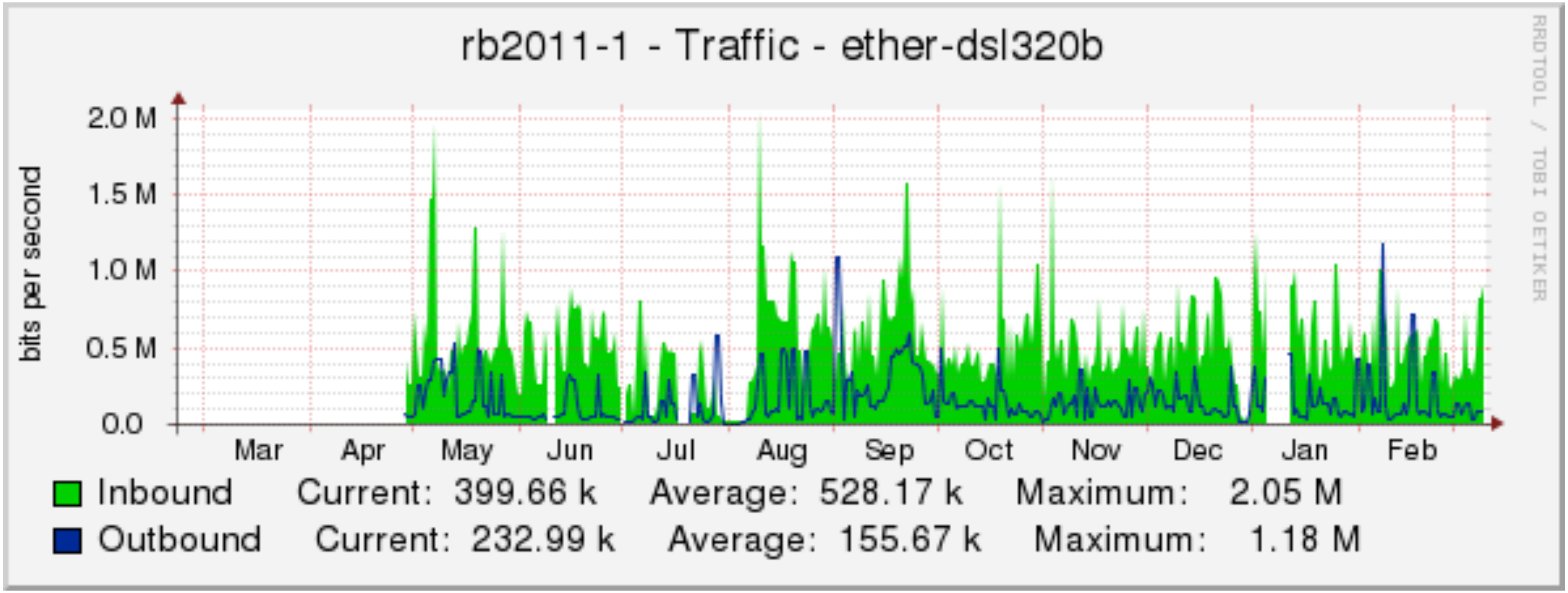
Showing Rows 1 to 19 of 19 [1]

Airports: v2 over UDP over IPv4 only

Netgear: UDP IPv4 only, v3 limited to only one user and therefore needs to know “admin” password

MikroTik: Does v3 over IPv6, but does not support TCP.

SNMP Uses


rb2011-1 - Traffic - ether-dsl320b
bits per second
2.0 M
1.5 M
1.0 M
0.5 M
0.0
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
🟩 Inbound Current: 399.66 k Average: 528.17 k Maximum: 2.05 M
🟦 Outbound Current: 232.99 k Average: 155.67 k Maximum: 1.18 M
RRDTOOL / TOBI OETIKER

SNMP Traps

- Less widely used, because only sent once over insecure and unreliable transport
- Much better to poll devices for problems than rely on single cry for help
- Typical use is to use traps to provoke polling, but poll anyway
  - “Cold Start” is useful because it indicates things like interface numbers might have changed

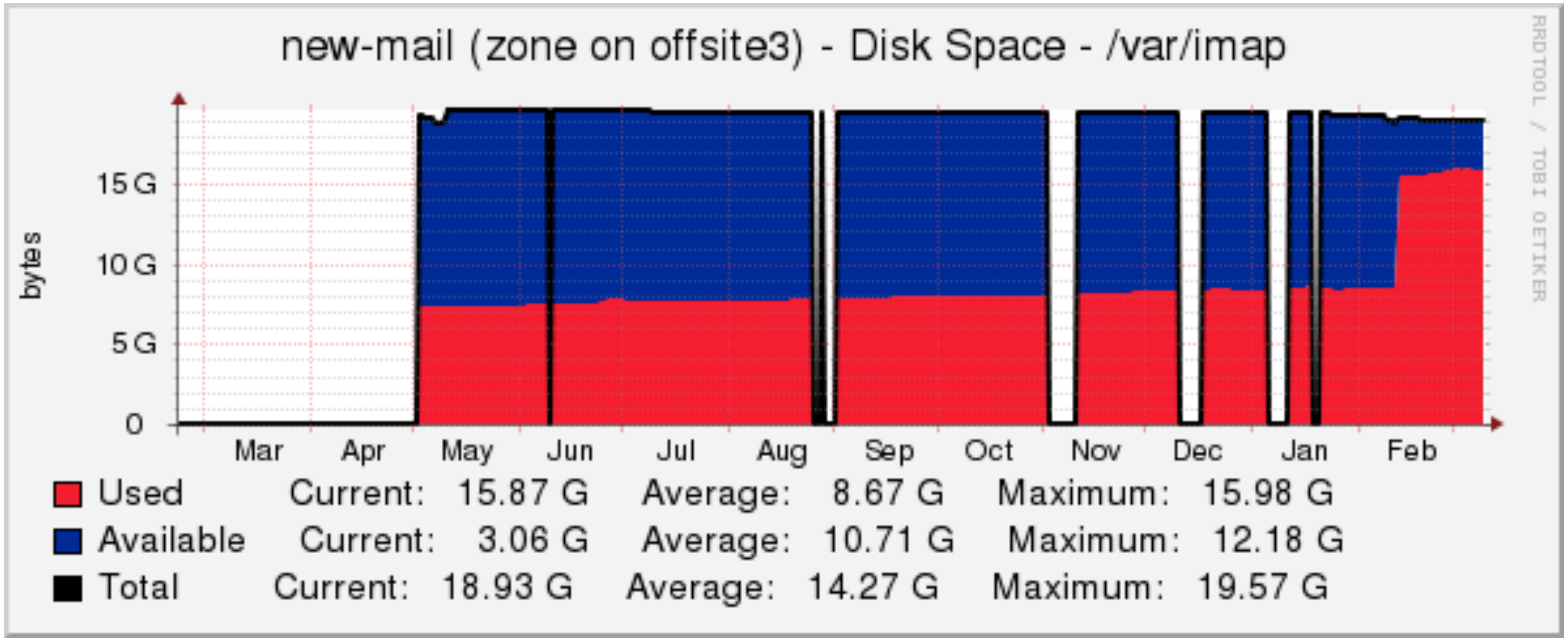
Not-network use

- Handy for monitoring and graphing disk, memory, temperature, etc on hosts

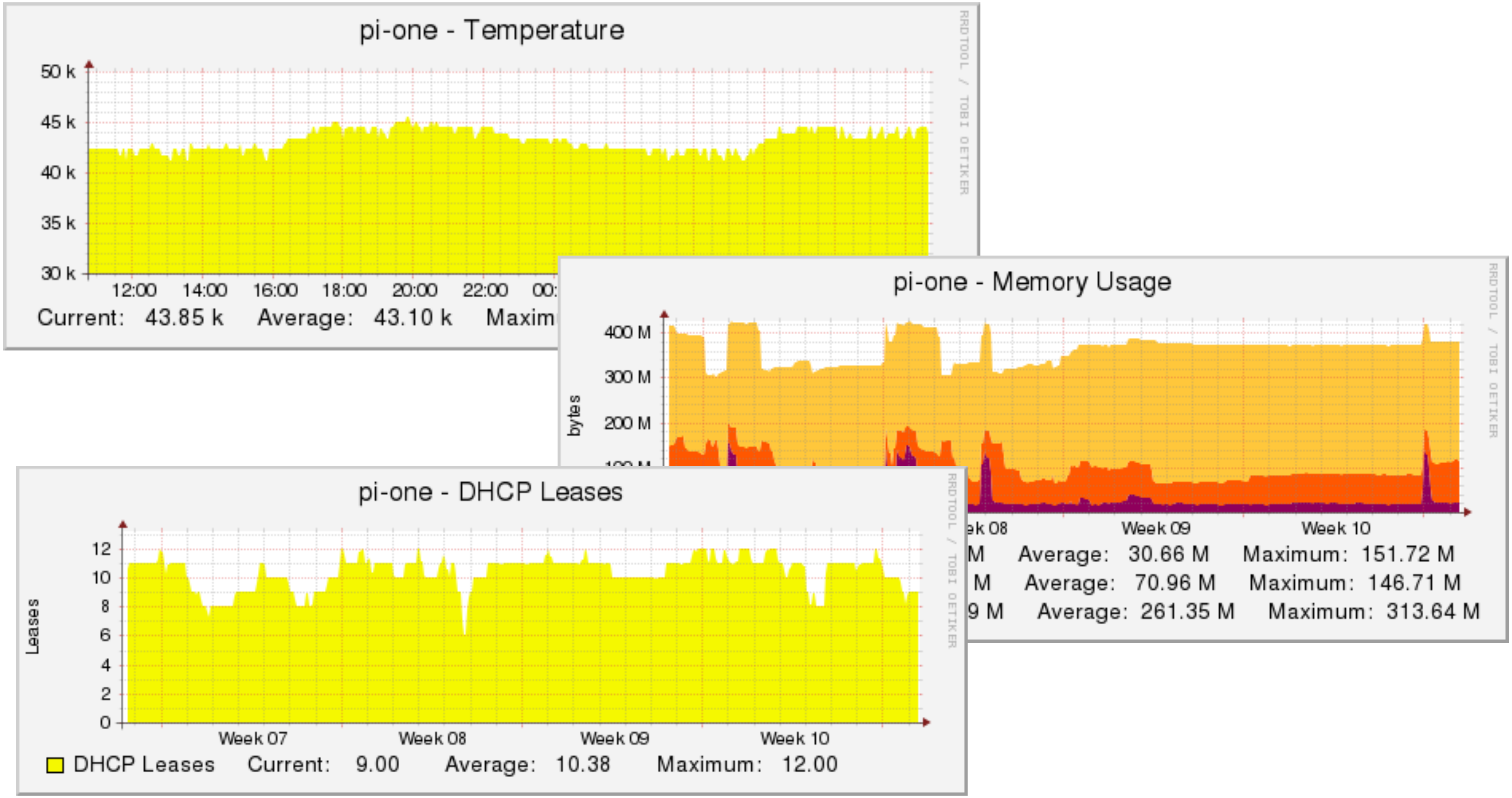
- Handy for getting reboot notifications via Cold Start traps

- MIBs exist for Apache, Cyrus, etc, etc, mostly to tie into graphing packages like Cacti.

Disk Usage


new-mail (zone on offsite3) - Disk Space - /var/imap
bytes
15 G
10 G
5 G
0
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
RRDTOOL / TOBI OETIKER
🟥 Used Current: 15.87 G Average: 8.67 G Maximum: 15.98 G
🟦 Available Current: 3.06 G Average: 10.71 G Maximum: 12.18 G
⬛ Total Current: 18.93 G Average: 14.27 G Maximum: 19.57 G

Stuff


pi-one - Temperature
50 k
45 k
40 k
35 k
30 k
12:00
14:00
16:00
18:00
20:00
22:00
00:
Current: 43.85 k
Average: 43.10 k
Maxim
RRDTOOL / TOBI OETIKER
pi-one - Memory Usage
bytes
400 M
300 M
200 M
ek 08
Week 09
Week 10
M Average: 30.66 M Maximum: 151.72 M
M Average: 70.96 M Maximum: 146.71 M
9 M Average: 261.35 M Maximum: 313.64 M
RRDTOOL / TOBI OETIKER
pi-one - DHCP Leases
Leases
12
10
8
6
4
2
0
Week 07
Week 08
Week 09
Week 10
DHCP Leases
Current: 9.00
Average: 10.38
Maximum: 12.00
RRDTOOL / TOBI OETIKER