Wietse's mailer project: Turning off Sendmail forever

Wietse Venema
IBM T.J. Watson Research Center
Hawthorne, NY, USA

cover.fig

Expectations about VMail

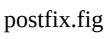
VMail: No experience yet, but I'd guess something like a wisened old man sitting on the porch outside the postoffice. Looks at everyone who passes by with deep suspicion, but turns out to be friendly and helpful once he realises you're not there to rob the place.

Posting in alt.sysadmin.recovery

vmail-quote.fig

~~Vmailer~~

Postfix



New code, new opportunities for bugs

- Solaris 7.0: 12 million lines, 10% new.
- Windows NT 5.0: 40+ million lines, 50+% new.
- Wietse’s long-term average: 1 bug per 1000 lines.
- Postfix: 30,000 lines of opportunity.

Bugs’r us - all code is suspect

- 10-year average BSD UNIX code size ratio

suid/daemon	:	libc	:	kernel
20	:	1	:	4

- 10-year average UNIX vulnerabilities ratio

suid/daemon	:	libc	:	kernel
25	:	1	:	2

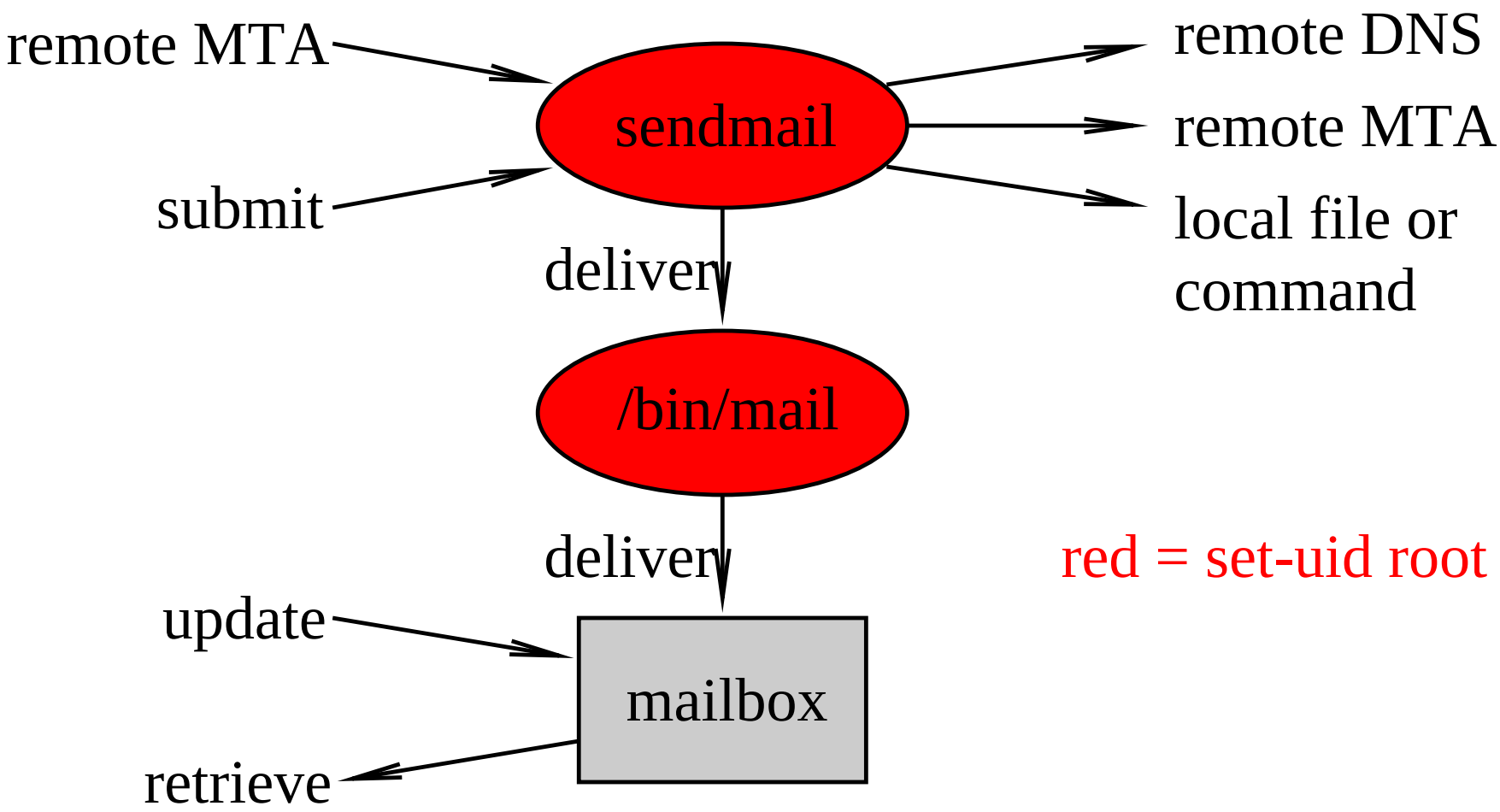
Based on SunOS 4, BSD/OS, FreeBSD; CIAC advisories; not including X Windows

bugs-r-us.fig

Overview of presentation

- Intro
- Why write yet another UNIX mailer?
- Why is the UNIX mail system so vulnerable?
- Postfix architecture and implementation
- Early experiences and plans

UNIX mail system: sendmail and /bin/mail



unix-mail.fig

Sendmail and /bin/mail advisories

CA-88:01	Sendmail 5.58	run any command (debug)
CA-90:01	SUN Sendmail	
CA-91:01a	SUN /bin/mail	root shell (local)
CA-91:13	Ultrix /bin/mail	root shell (local)
CA-93:15	SUN Sendmail	write any file (remote)
CA-93:16	Sendmail 8.6.3	run any command (from)
CA-94:12	Sendmail 8.6.7	root shell (-d bignumber)
		read any file (-oE filename)
CA-95:02	/bin/mail	write any file (race)

Sendmail and /bin/mail advisories (cont)

CA-95:05	Sendmail 8.6.9	any command/file (ident)
CA-95:08	Sendmail V5	any command/file
CA-95:11	SUN Sendmail	root shell (-oR host -f cmd)
CA-95:13	Sendmail 8.7.0	root shell (syslog)
CA-96.04	Sendmail 8.7.3	root shell (dns newline)
CA-96.20	Sendmail 8.7.5	root shell (fullname buffer)
		default uid/gid (getpwuid)
CA-96.24	Sendmail 8.8.2	root shell (argv[0])
CA-96.25	Sendmail 8.8.3	group id (:include:, .forward)
CA-97.05	Sendmail 8.8.4	root shell (mime buffer)

Wietse’s mailer primary goals

- Wide deployment by giving it away
- Compatibility: make transition easy
- Performance: faster than the competition
- Security: no root shells for random strangers
- Flexibility: C is not an acceptable scripting language
- Reliability: behave rationally under stress
- Example for book with Dan

The book with Dan

- Is late (so what's new)
- One new tool for each chapter?
- Postfix: programming chapter
- Coroner's Toolkit: investigating a break-in aftermath

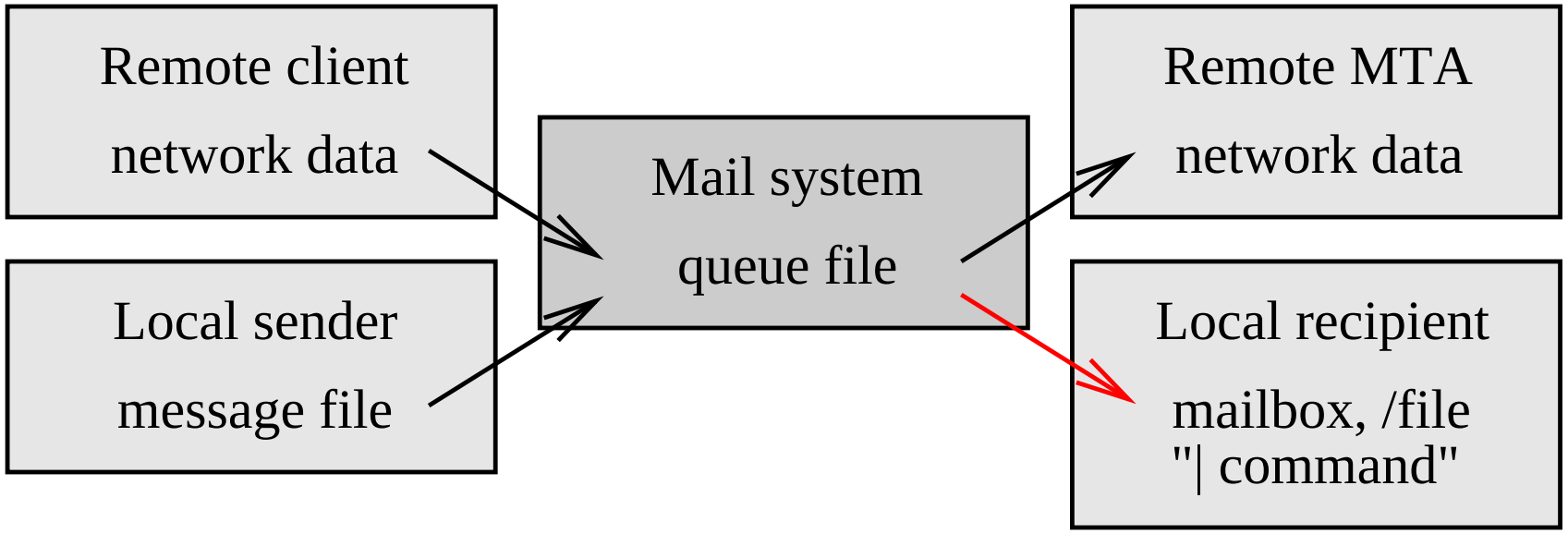
Challenges of implementing UNIX mail

- Network protocols (lots of broken software to talk to)
- Concurrent mail database access
- Mail address parsing, rewriting, and routing
- Queue management

All this plus: SPAM/relay control, security and performance

challenges.fig

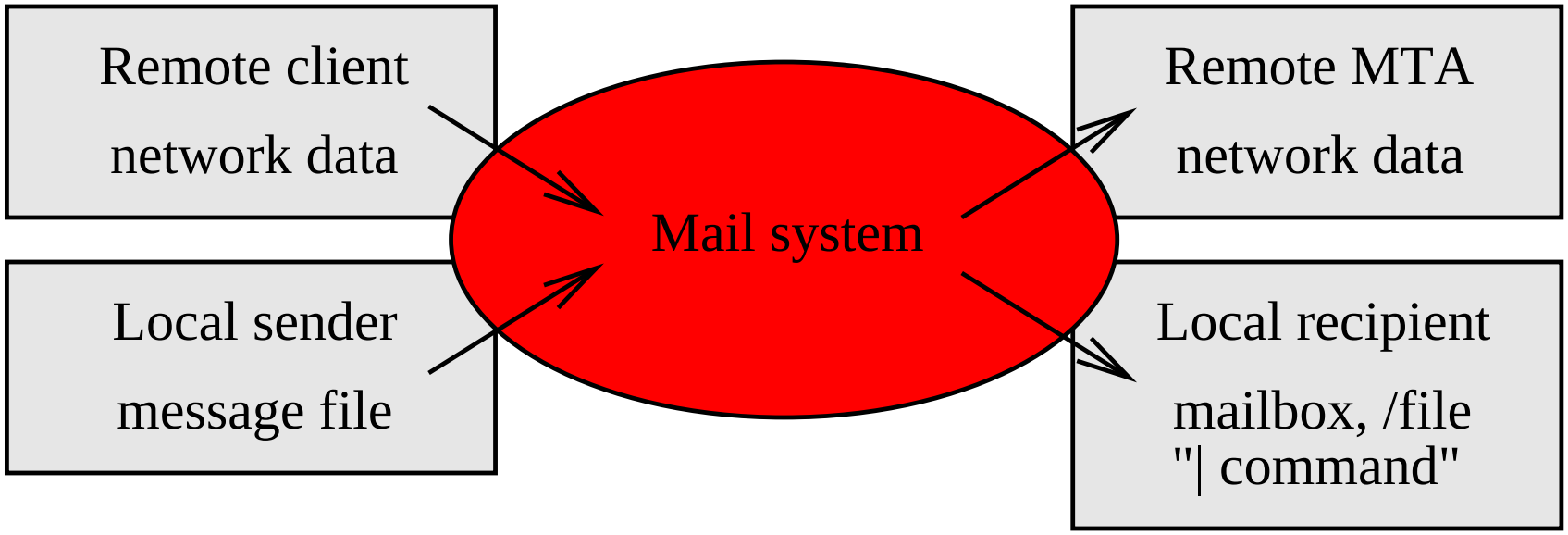
Security challenge: multiple privilege levels



- Entry into mail queue involves "mail system" privilege
- Local delivery involves root privilege

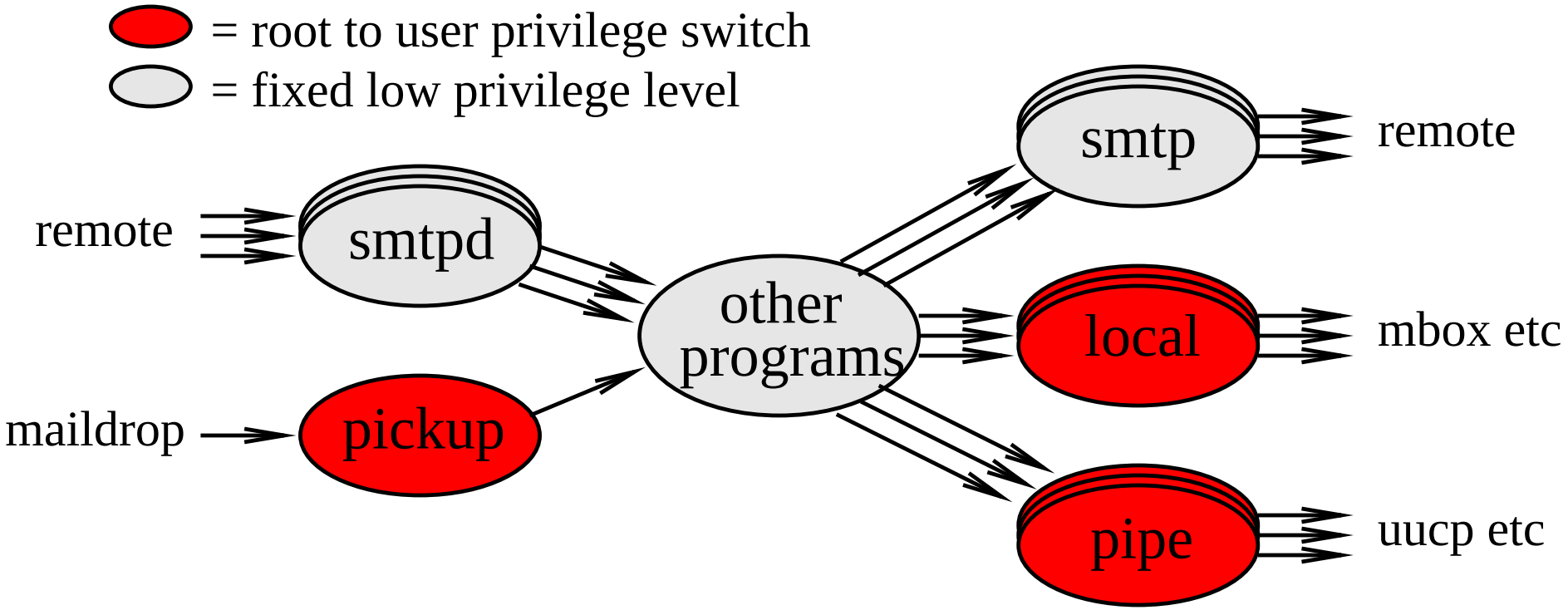
security-challenge.fig

Monolithic mailer: no damage control



- The entire mail system runs at the highest privilege level
- Result: one vulnerability compromises the entire host

Postfix: multiple layers of defense



- Partitioned architecture for damage control
- Most programs run in chroot/low privilege jail
- No trust in queue files or in IPC messages

damage-control.fig

Eliminating security problems

- No set-uid programs
- No /tmp race conditions
- No remote data in shell variables or shell commands
- No fixed-length string buffers
- No unbounded string sizes, either :-)

Compatibility

All the good things from sendmail, none of the pain

- /etc/aliases, NIS aliases, even NetInfo aliases
- /var/spool/mail/user, /var/mail/user, user.lock files
- $HOME/.forward, :include:/file/name
- Delivery to /file/name
- Delivery to "| command"
- No sendmail.cf file

Address rewriting and routing

- No rewriting language in version 1 (only few need it)
- Table-driven operation (db, dbm, NIS, NetInfo)
  - Canonical: substitute any address/domain (S3)
  - Virtual: redirect any address/domain (S0)
  - Transports: route any domain to transport:relayhost
  - Relocated: address change notice per address/domain
  - Aliases: redirect or expand local recipient

rewriting.fig

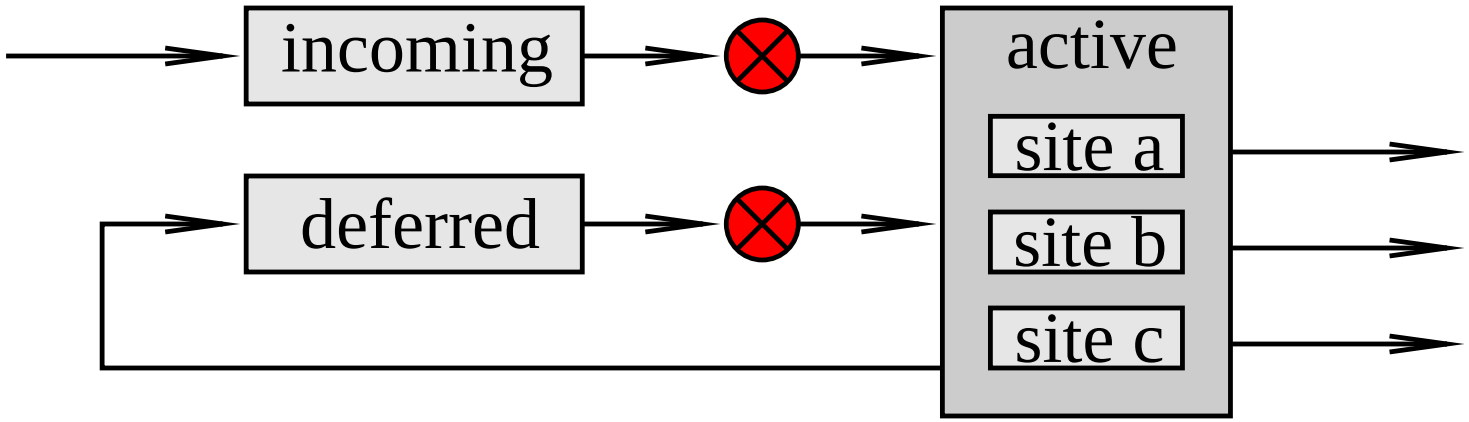
SPAM mail

Basic spam controls:

- Relay control (default: from/to own network/domain)
- Helo/client/sender/recipient black/white list
- MX backup, RBL lookups
- Helo/sender domain DNS lookup (soft failure)
- Tarpit: sleep(5+) before every [45]xx response

spam-control.fig

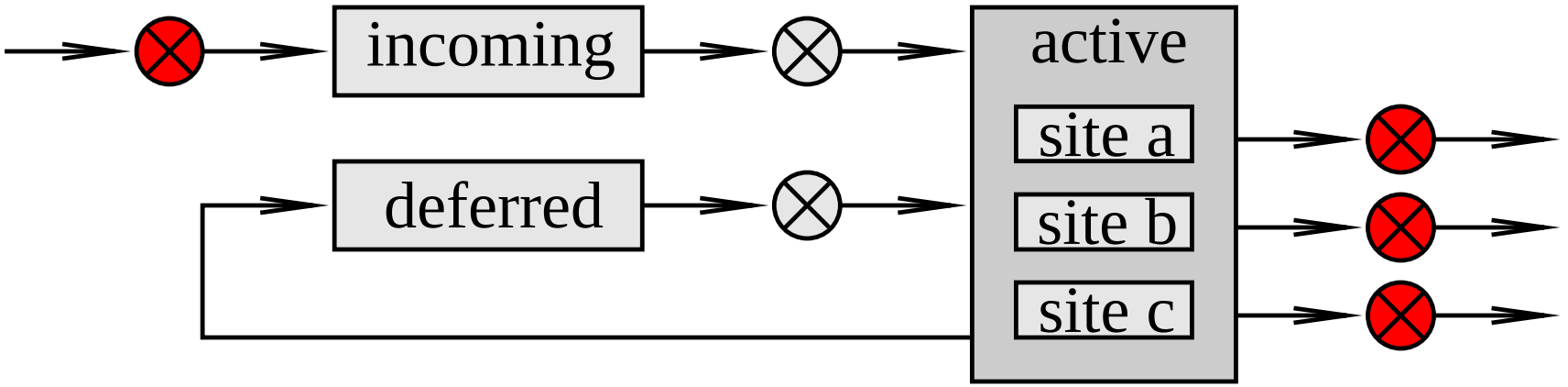
Queue management



- Leaky bucket: new/delayed mail to small in-core queue
- Round-robin walk to select destination site queue
- Random walk within destination site queue
- Dead site list (up to some # sites)
- Per-message exponential backoff

queue-mgt.fig

Connection management



- No crashing when mail starts pouring in after outage

  Controlled number of inbound connections

- No annoying thundering herd deliveries

  "Slow start" of parallel connections to the same site

connection-mgt.fig

Lies, damned lies, and benchmarks

Good news: up to three times faster than qmail:

- From network to local recipient
- From network to remote recipient

Good news: at least as fast as qmail:

- From network to mailing list
- Does multi-recipient transfers by default

Bad news: the mail system is horribly slow

benchmarks.fig

Mail system performance limits (TP760)

- It takes gross inefficiency to saturate the CPU
  FreeBSD does 100 fork()+exec()/s on a P133

- It takes a moderate effort to saturate the network
  Postfix/qmail do 75 deliveries/s on 10Mb/s LAN

- It’s a no-brainer to saturate the BSD fast file system
  Postfix saturates at 5.3 messages/s (1 queue file)
  qmail saturates at 1.8 messages/s (3 queue files)

bottleneck.fig

Hardware performance evolution in 10 years

Typical hardware in 1988		Improvement
CPU speed	5 MIPS	100
Disk size	300 MB	10-100
Memory	8 MB	10-100
Disk bandwidth	2 MB/s	10
Light speed	3.10^8 M/s	1

hardware.fig

Apples vs oranges: observed speedup factors

2 x	Soft metadata updates (Ganger and Patt, with metadata-intensive applications) To be adopted in FreeBSD FFS soon
5-10 x	Append/truncate, not create/delete, small files
7 x	Prestoserve non-volatile cache, small files

Mail performance is limited by latencies, not by bit rates!

Combining tricks doesn’t necessarily give more speedup!

Postfix status and future plans

- December 1997: Sendmail turned off forever
- January 1998: closed alpha release
- 1998: public beta release
- BSDI DEC *BSD HP IBM LINUX NEXT SGI SUN
- Will be a bear to port to Windows
- More info on: http://www.postfix.org/