- Current developments

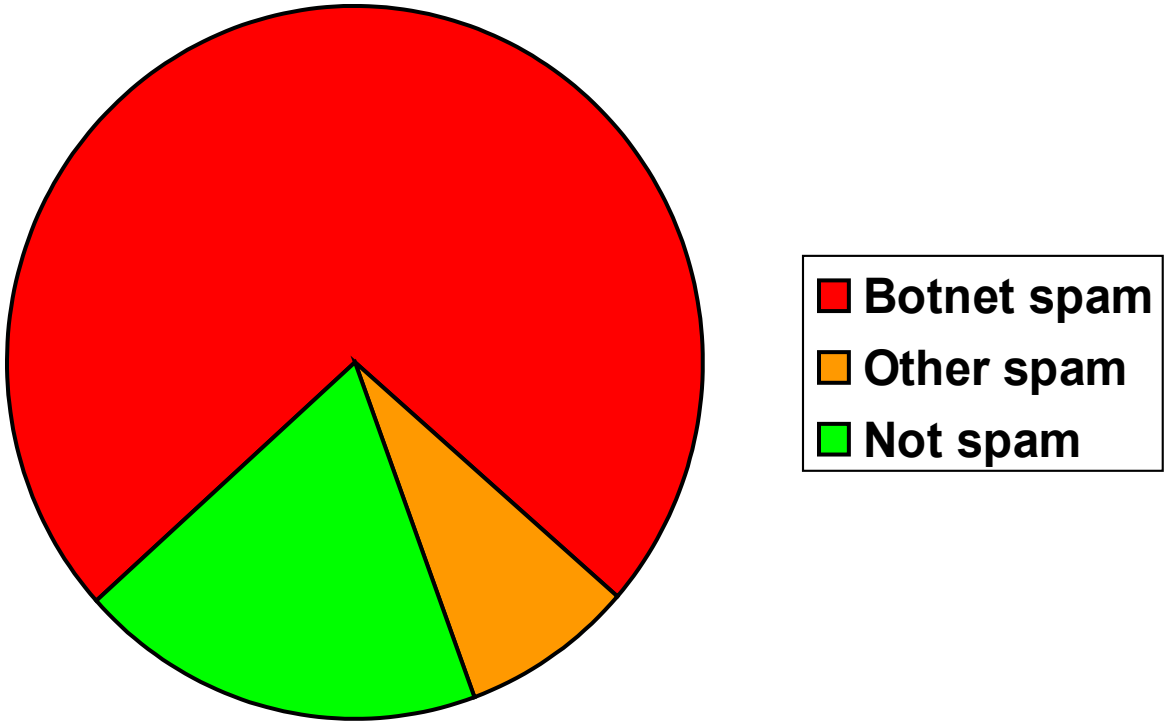
“Zombies suck the life out of the mail server.”

Wietse at mailserver conference, 2009

Changing threats

- 1999: You built a mail system that runs on UNIX, so you didn't have to worry about Windows viruses.
  - *Problem*: your UNIX-based MTA becomes a major distribution channel for Windows malware (Melissa).
  - *Solution*: outsourcing to external content filters.
- 2009: You built a mail system that has world-class email delivery performance.
  - *Problem*: your world-class performing mail system is now spending most of its resources not delivering email.

81% of email is spam, 90% is from botnets[1]



[1]MessageLabs 2008 annual report

Changing threats

Zombies suck the life out of the mail server

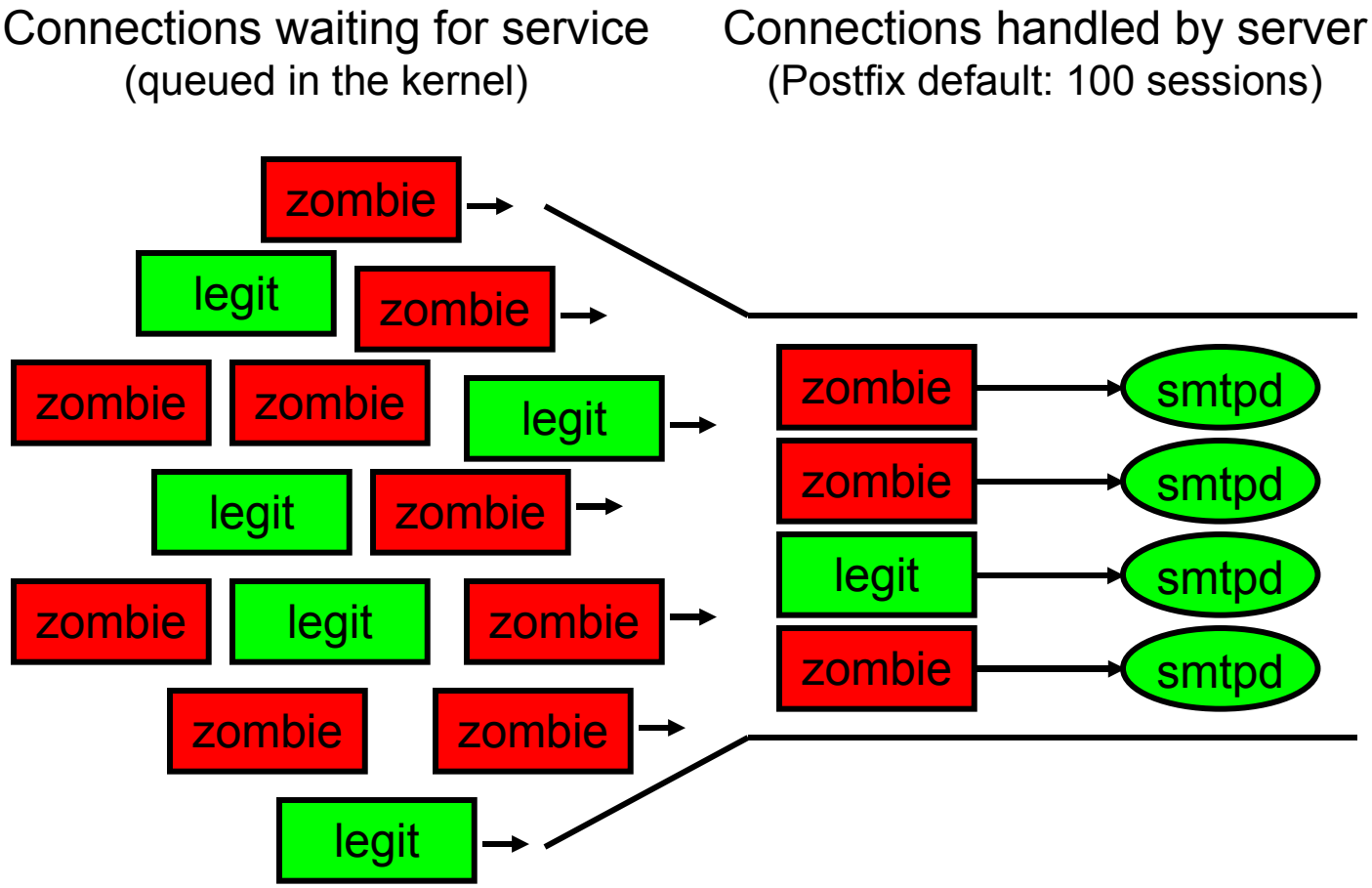
- Worst-case example: Storm botnet, August 2007.

```
15:16:55 postfix/smtpd: connect from [x.x.x.x]
15:16:56 postfix/smtpd: reject: RCPT from [x.x.x.x]:
        550 5.7.1 blah blah blah
15:21:56 postfix/smtpd: timeout after RCPT from [x.x.x.x]
```

- RFC 5321 recommends 5-minute server-side timeout.
  - Postfix implements SMTP according to the standard...
    - Result: all SMTP server ports are kept busy by zombies.

Changing threats

Zombies keep mail server ports busy



Changing threats

Symptoms of mail server overload

- Clients experience delays before the server responds.
  - Not to be confused with delays due to broken DNS configurations.
- Servers log large numbers of “lost connection” events.
  - Clients hang up before the server responds.
  - Not to be confused with “lost connection” due to portscanning activity.
- Postfix ≥ 2.3 logs “all server ports busy” warnings.

Overload handling strategies, part 1 of 2

- Strategies for *temporary* overload:
  - Work faster: spend less time per (zombie) client: reduce time limits, number of failed commands per session, etc.
    - May delay *some* legitimate email messages.
      - Better to receive most legitimate mail than almost no email.
    - OK if the overload condition is temporary.

Changing threats

Temporary overload - default “on” in Postfix 2.6

Off by default in Postfix 2.5, patches for Postfix 2.4 and 2.3.

- Postfix master(8) daemon sets “stress” configuration parameter on network daemon command line[1]:

```
smtpd -o stress=yes      (overload)
smtpd -o stress=         (normal)
```

- Postfix main.cf settings:

```
smtpd_timeout = ${stress?10}${stress:300}s
smtpd_hard_error_limit = ${stress?1}${stress:20}
smtpd_junk_command_limit = ${stress?1}${stress:100}
```

[1]Feature is called “stress”, and implemented in 21 lines, because of author overload.

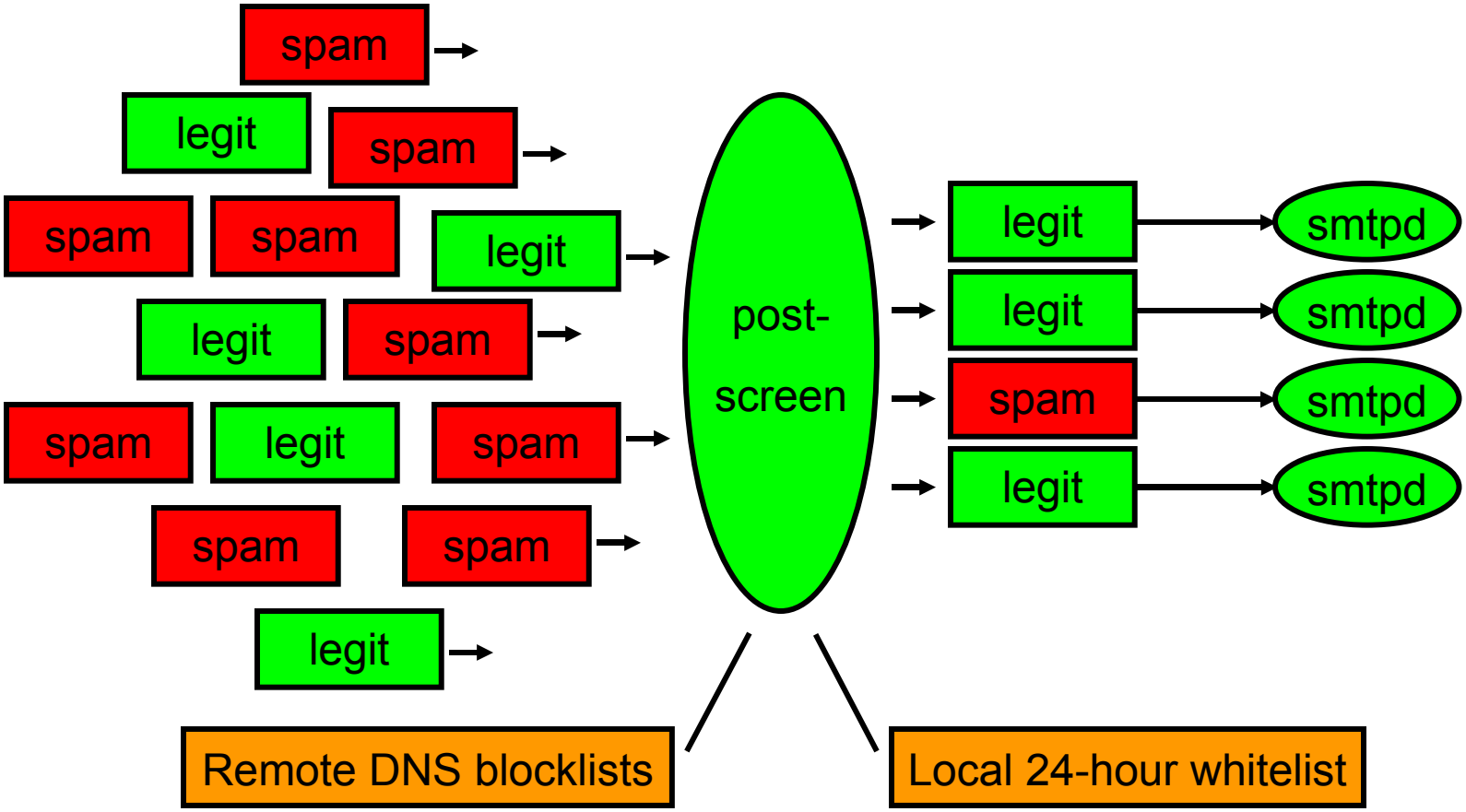
Changing threats

Overload handling strategies, part 2 of 2

- Strategies for *persistent* overload:
  - Work harder: configure more SMTP server slots.
    - OK if you can afford the memory, disk, and cpu resources.
  - Work smarter: keep mailbots away from SMTP server.
    - More SMTP server slots remain available for handling email.

Persistent overload - before-smtpd connection filter

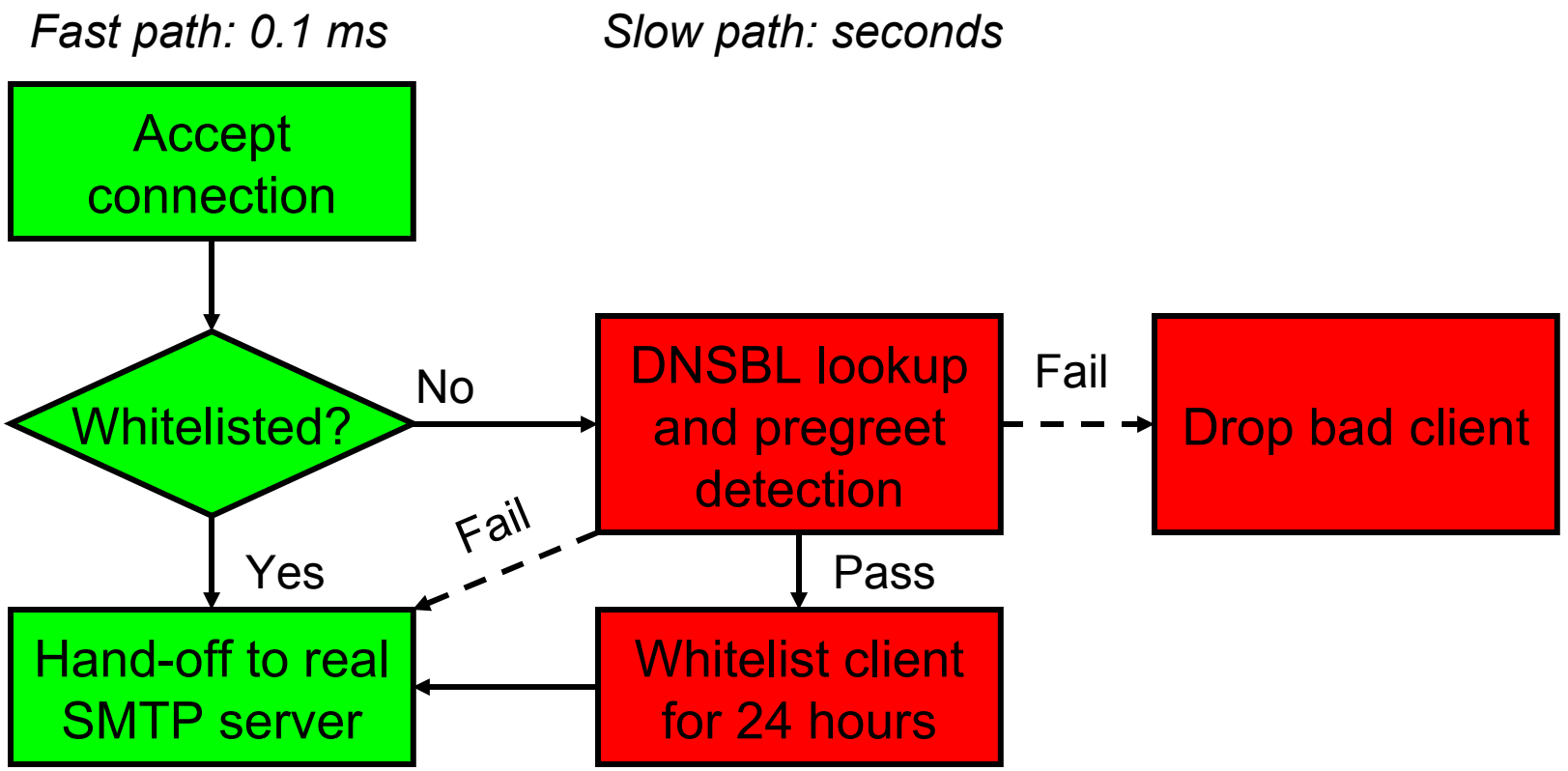
Prior work: OpenBSD spamd, MailChannels TrafficControl, M.Tokarev



Changing threats

Prototype postscreen architecture

All lookups and connections are handled in parallel



Non-production prototype (source code is on-line)

- Code name *postscreen*, but name will likely change.
- Single daemon checks all connections *first,* so that Postfix SMTP server processes waste less time.
  - PREGREET detection (bots that start talking too soon).
  - Parallel lookups for multiple DNS blocklists.
  - Fast-path cache (24 hours) for clients that pass the tests.
- Not a proxy. No need to handle STARTTLS, etc. Just send the network socket to the real SMTP server.
  - Add mini-SMTP engine to log rejected sender/recipient.

Changing threats

Pregreet detection

Botnet/proxy SMTP clients that speak before their turn

- Good SMTP clients wait for the SMTP server greeting:

  SMTP server: **220 server.example.com ESMTP Postfix<CR><LF>**

  SMTP client: **EHLO client.example.org<CR><LF>**

- Poor results with the Sendmail *greetpause* approach: wait several seconds before sending the 220 greeting.
  - Some clients spontaneously send QUIT after 5 seconds.
  - Some clients spontaneously hang up after few seconds.
- Bad idea to do such delays in the SMTP server itself!

Quick question

- Q1: How do you find out if a house has a dog?
- A1: You listen and wait until a dog barks.

- Q2: What if I don't want to wait?
- A2: You ring the doorbell, and it will bark immediately.

Pregreet detection improved - multi-line reply trap

Catching more SMTP clients that speak before their turn

- Good clients wait for the end of multi-line server reply:

postscreen: **220–server.example.com ESMTP Postfix<CR><LF>**

postscreen: **[pause a few seconds here]**

real smtpd: **220 server.example.com ESMTP Postfix<CR><LF>**

good client: **HELO client.example.org<CR><LF>**

- Some 50% of the bots starts talking immediately:

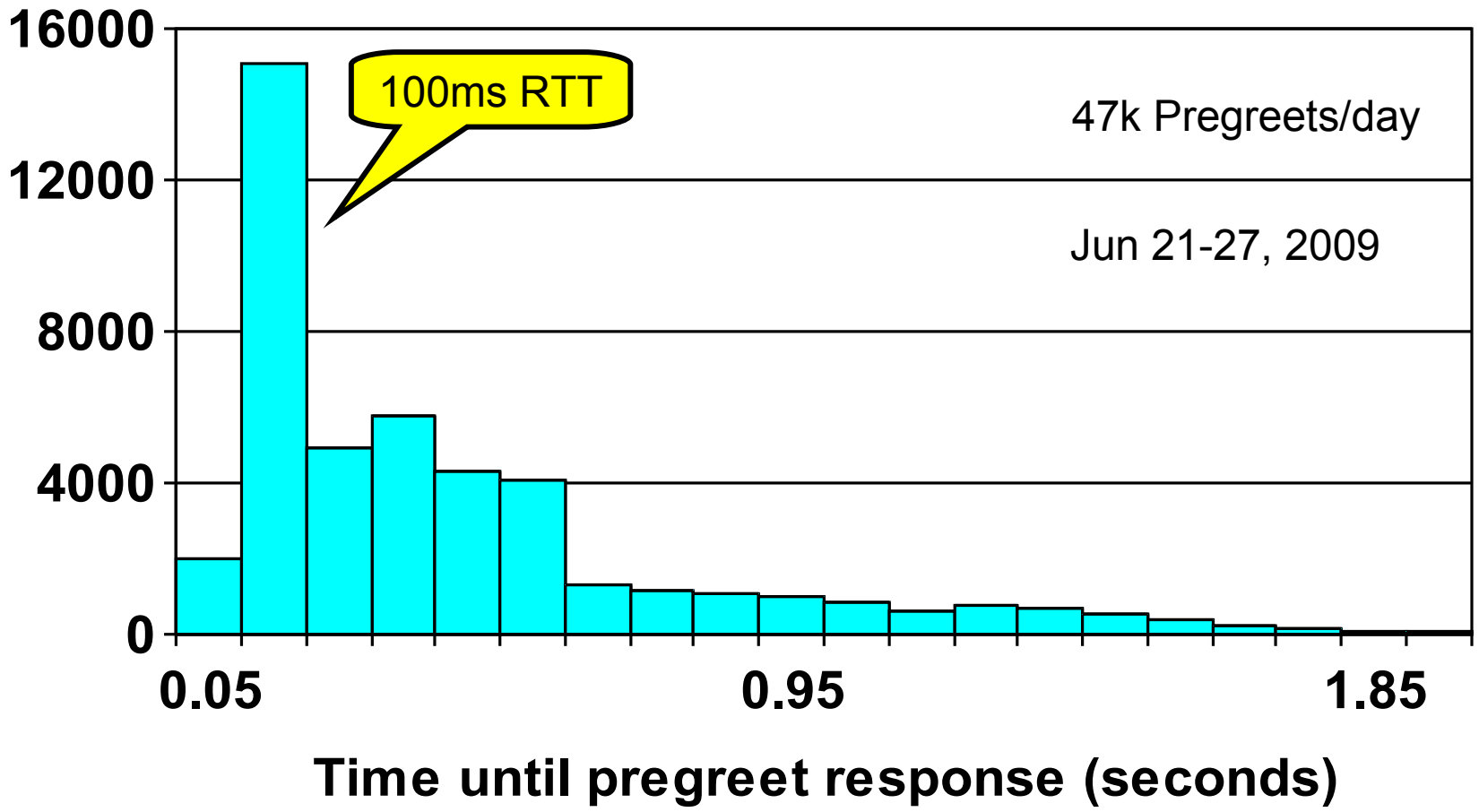
postscreen: **220–server.example.com ESMTP Postfix<CR><LF>**

spambot: **HELO i-am-a-bot<CR><LF>**

Changing threats

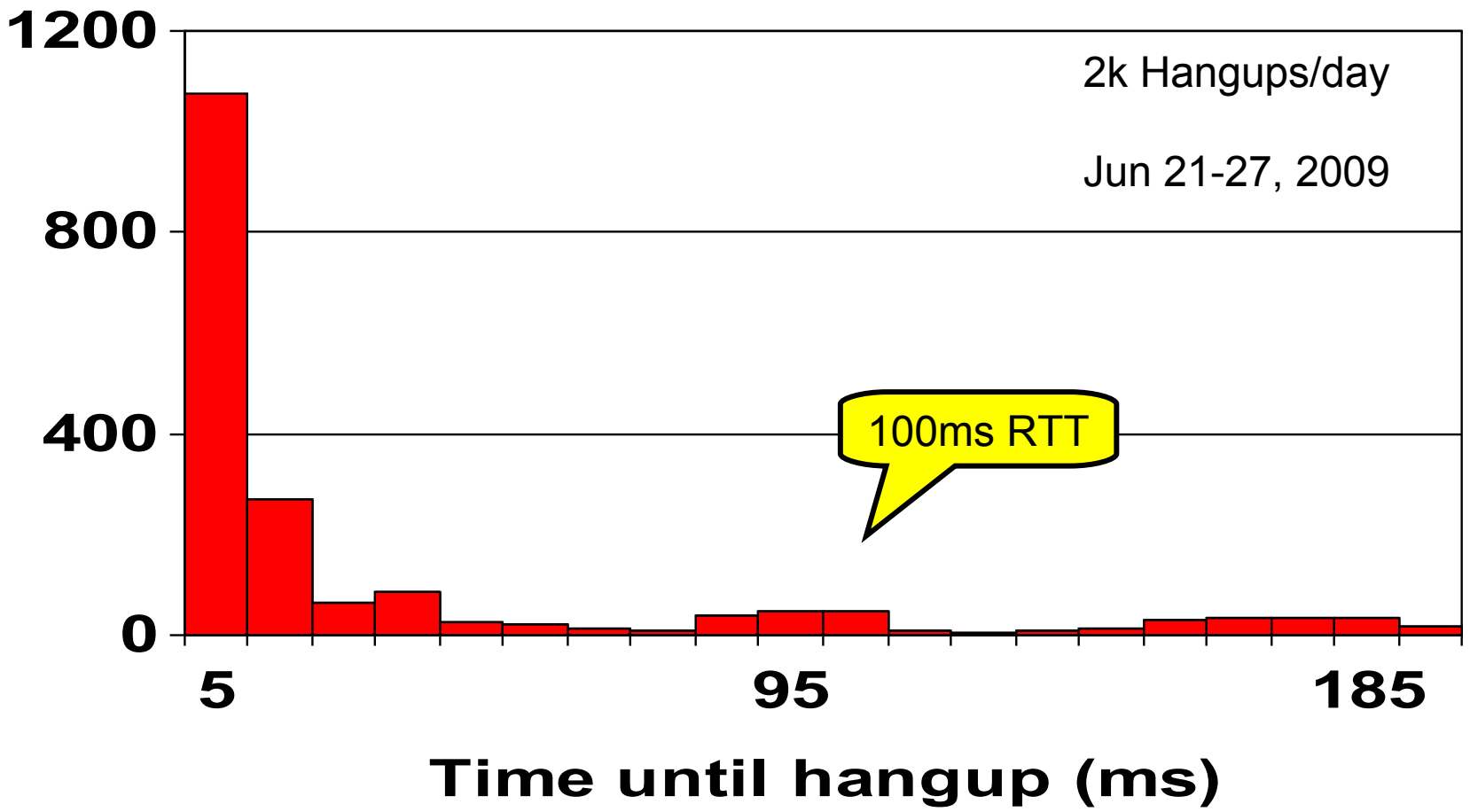
Results: >50% of bots pregreet (charite.de)

99% are blocklisted, but that may change



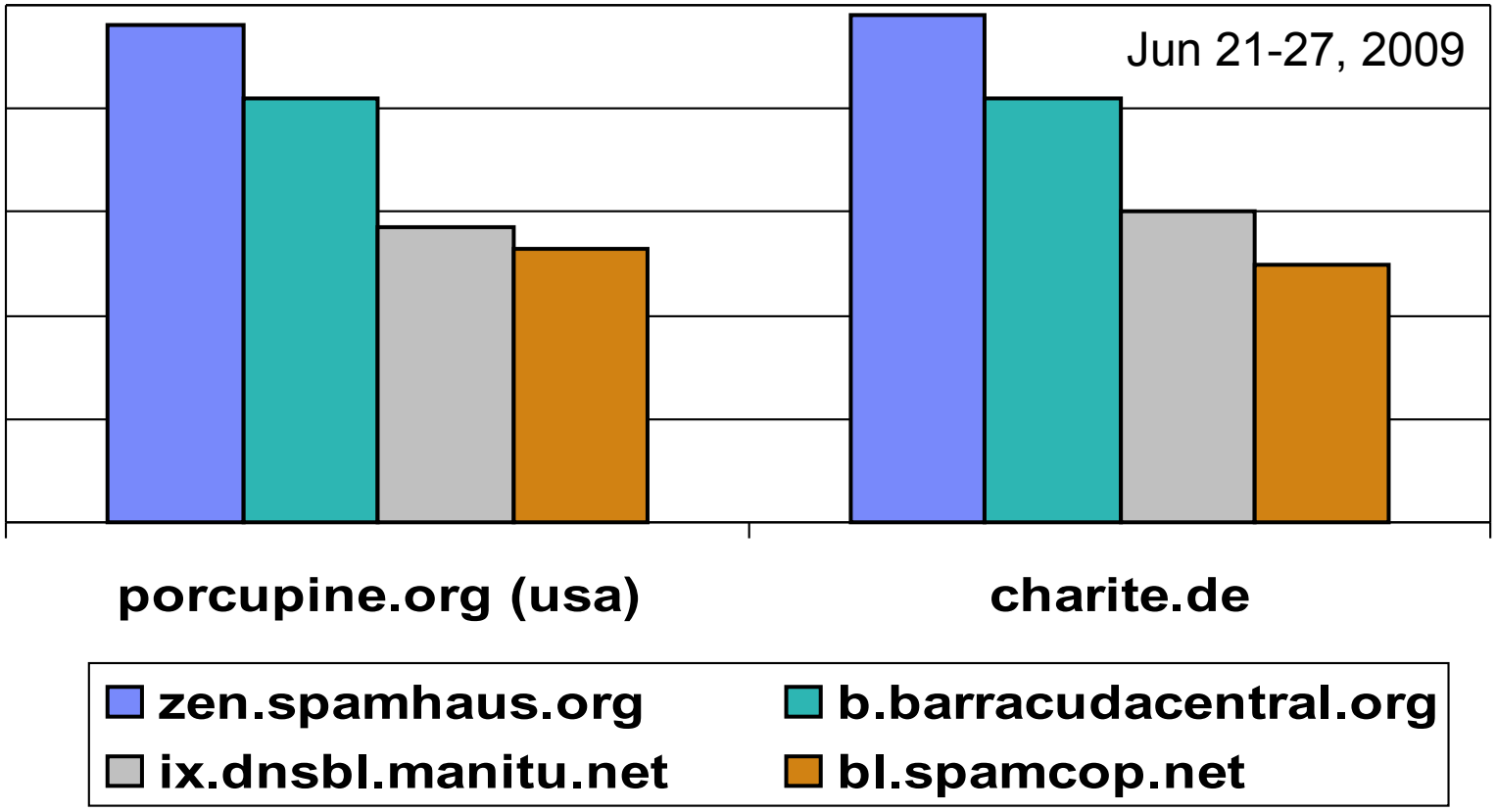
Changing threats

Result: 2% of bots hang up quickly (charite.de)



Changing threats

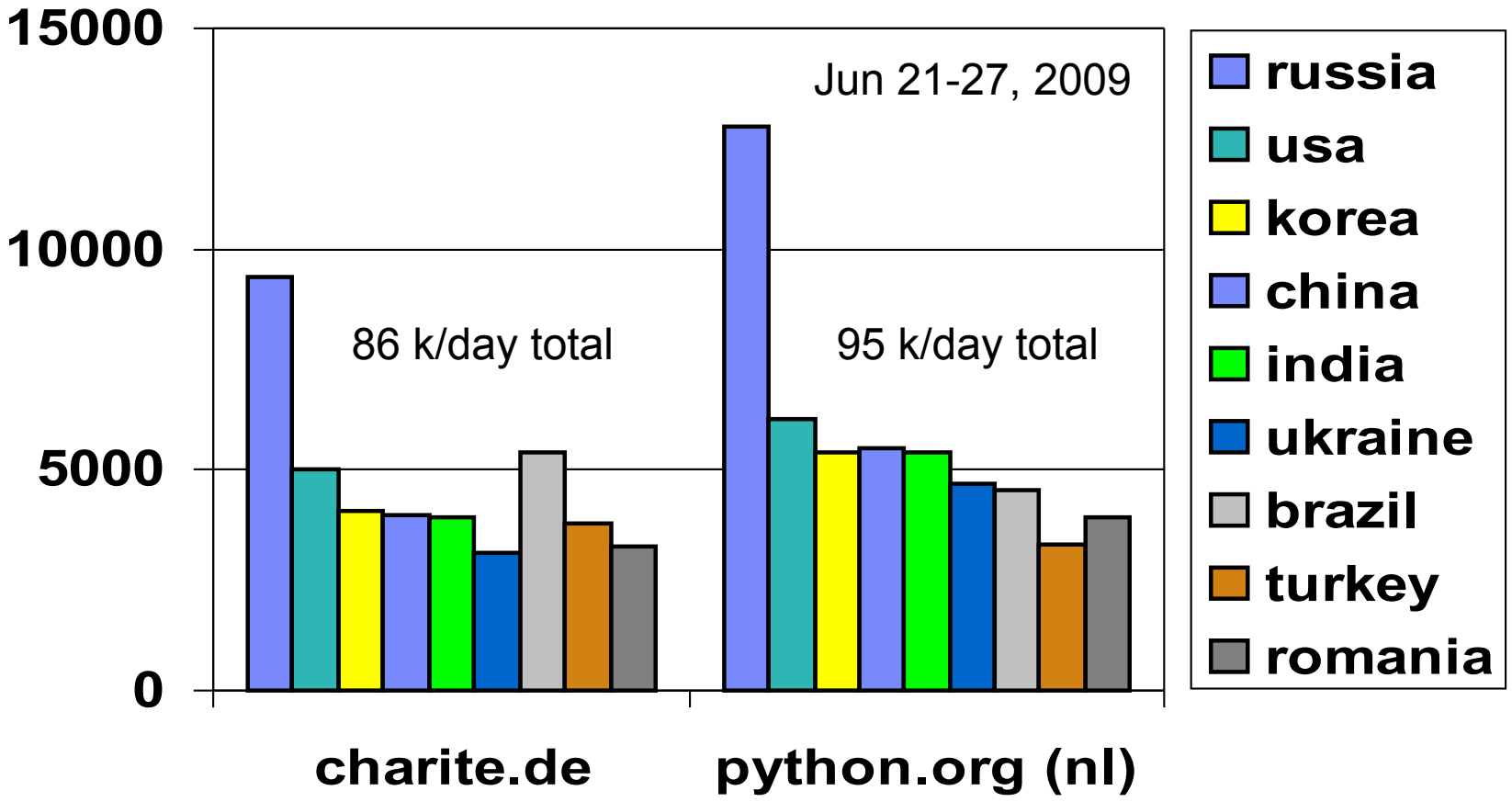
Relative DNS blocklist client IP address coverage



Changing threats

Results: spam connections/day

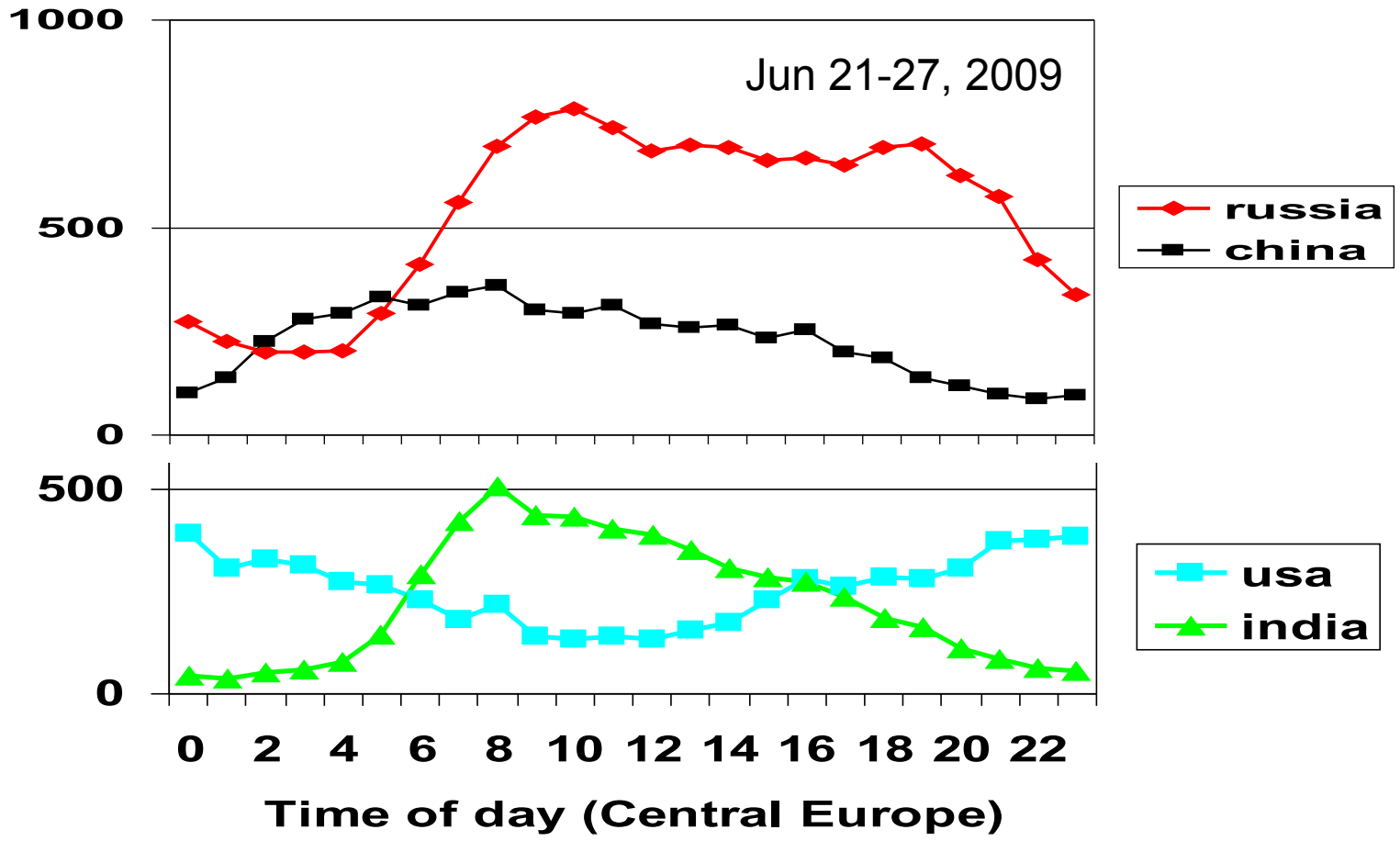
Spam according to zen.spamhaus.org DNS blocklist



Changing threats

Results: spam connections/hour (python.org)

Spam according to zen.spamhaus.org DNS blocklist



Changing threats

Results: spam connections/hour (charite.de)

Spam according to zen.spamhaus.org DNS blocklist