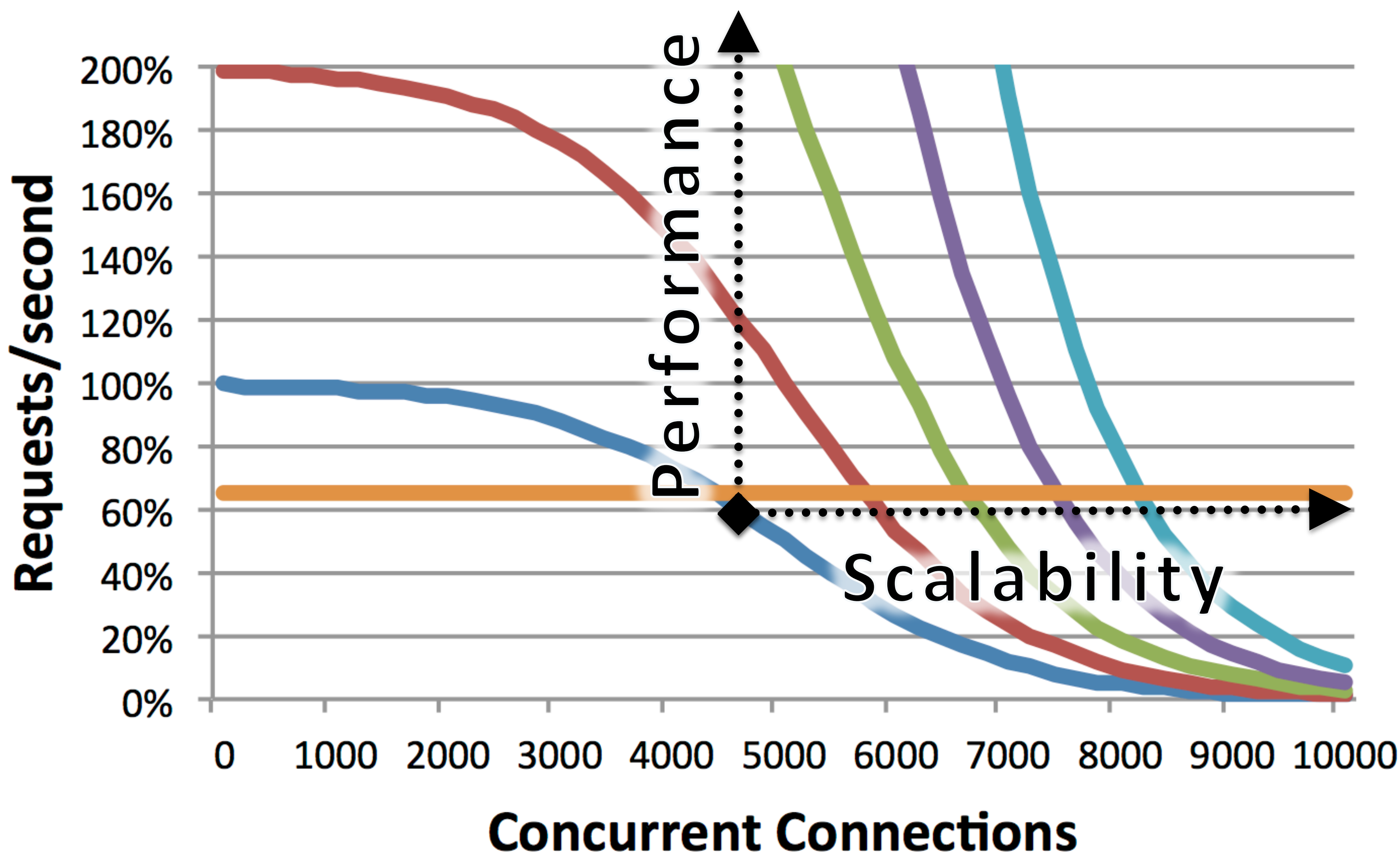
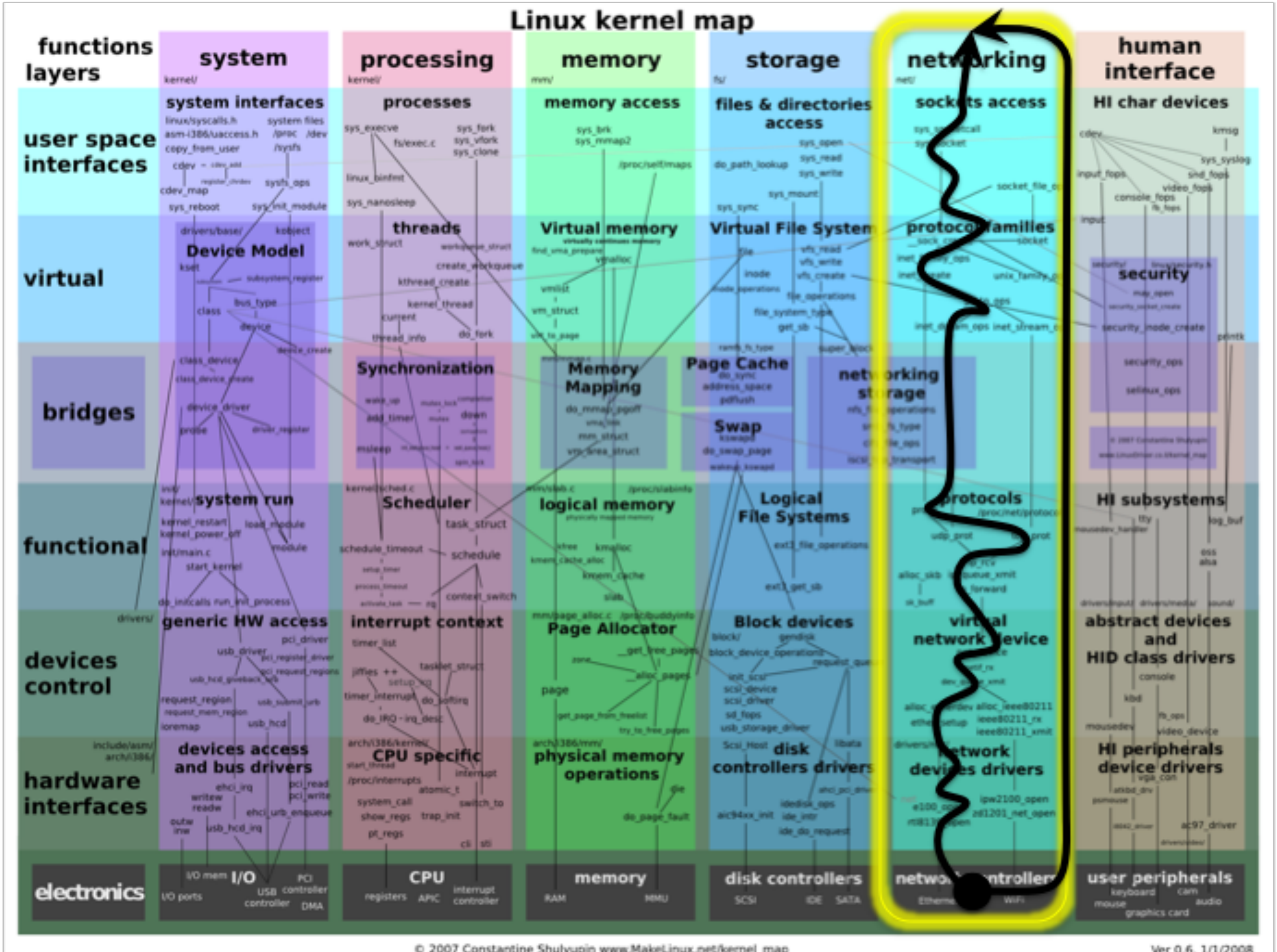
C10M:
radical parsers

by Robert David Graham
@ErrataRob
http://c10m.robertgraham.com/



Performance
Scalability
Requests/second
200%
180%
160%
140%
120%
100%
80%
60%
40%
20%
0%
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Concurrent Connections

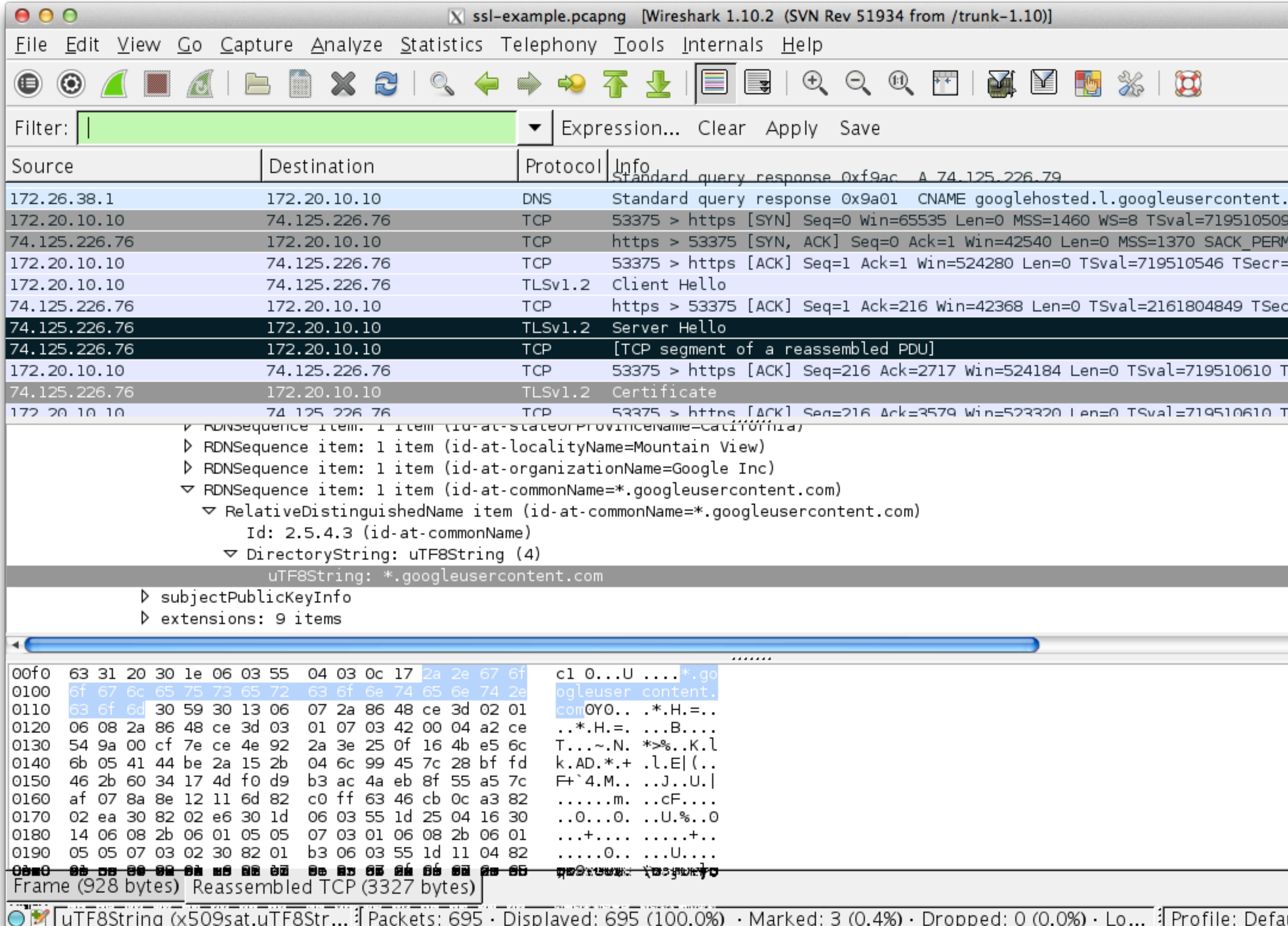

Linux kernel map
functions
layers
system
processing
memory
storage
networking
human interface
user space interfaces
system interfaces
processes
memory access
files & directories access
sockets access
HI char devices
virtual
Device Model
threads
Virtual memory
Virtual File System
protocol families
security
bridges
Synchronization
Memory Mapping
Page Cache
Swap
networking storage
functional
system run
Scheduler
logical memory
Logical File Systems
protocols
HI subsystems
devices control
generic HW access
interrupt context
Page Allocator
Block devices
virtual network device
abstract devices and HID class drivers
hardware interfaces
devices access and bus drivers
CPU specific
physical memory operations
disk controllers drivers
network devices drivers
HI peripherals device drivers
electronics
I/O
CPU
memory
disk controllers
network controllers
user peripherals
© 2007 Constantine Shulyupin www.MakeLinux.net/kernel_map
Ver 0.6, 1/1/2008

pointer chasing

- At scale (millions of connects) nothing is in high-speed cache
- Packet-descriptors -> TCB -> socket -> thread

layers

- Layer1 – Physical
  - Bits onto the wire
- Layer 2 – Local link
  - Frames to the next hop (local address)
- Layer 3 – Internet
  - Packets across the net to the remote machine
- Layer 4 – Transport
  - Payload to the target app


ssl-example.pcapng [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
Source Destination Protocol Info
172.26.38.1 172.20.10.10 DNS Standard query response 0x9a01 CNAME googlehosted.l.googleusercontent.
172.20.10.10 74.125.226.76 TCP 53375 > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=8 TSval=71951050
74.125.226.76 172.20.10.10 TCP https > 53375 [SYN, ACK] Seq=0 Ack=1 Win=42540 Len=0 MSS=1370 SACK_PERM
172.20.10.10 74.125.226.76 TCP 53375 > https [ACK] Seq=1 Ack=1 Win=524280 Len=0 TSval=719510546 TSecr=
172.20.10.10 74.125.226.76 TLSv1.2 Client Hello
74.125.226.76 172.20.10.10 TCP https > 53375 [ACK] Seq=1 Ack=216 Win=42368 Len=0 TSval=2161804849 TSec
74.125.226.76 172.20.10.10 TLSv1.2 Server Hello
74.125.226.76 172.20.10.10 TCP [TCP segment of a reassembled PDU]
172.20.10.10 74.125.226.76 TCP 53375 > https [ACK] Seq=216 Ack=2717 Win=524184 Len=0 TSval=719510610 T
74.125.226.76 172.20.10.10 TLSv1.2 Certificate
RDNSequence item: 1 item (id-at-localityName=Mountain View)
RDNSequence item: 1 item (id-at-organizationName=Google Inc)
RDNSequence item: 1 item (id-at-commonName=*.googleusercontent.com)
RelativeDistinguishedName item (id-at-commonName=*.googleusercontent.com)
Id: 2.5.4.3 (id-at-commonName)
DirectoryString: uTF8String (4)
uTF8String: *.googleusercontent.com
subjectPublicKeyInfo
extensions: 9 items
00f0 63 31 20 30 1e 06 03 55 04 03 0c 17 2a 2e 67 6f c1 0...U ....*.go
0100 6f 67 6c 65 75 73 65 72 63 6f 6e 74 65 6e 74 2e ogleuser content.
0110 63 6f 6d 30 59 30 13 06 07 2a 86 48 ce 3d 02 01 com0Y0.. .*.H.=..
0120 06 08 2a 86 48 ce 3d 03 01 07 03 42 00 04 a2 ce ..*.H.=. ...B....
0130 54 9a 00 cf 7e ce 4e 92 2a 3e 25 0f 16 4b e5 6c T...~.N. *>%..K.l
0140 6b 05 41 44 be 2a 15 2b 04 6c 99 45 7c 28 bf fd k.AD.*.+ .l.E|(..
0150 46 2b 60 34 17 4d f0 d9 b3 ac 4a eb 8f 55 a5 7c F+`4.M.. ..J..U.|
0160 af 07 8a 8e 12 11 6d 82 c0 ff 63 46 cb 0c a3 82 ......m. ..cF....
0170 02 ea 30 82 02 e6 30 1d 06 03 55 1d 25 04 16 30 ..0...0. ..U.%..0
0180 14 06 08 2b 06 01 05 05 07 03 01 06 08 2b 06 01 ...+.... .....+..
0190 05 05 07 03 02 30 82 01 b3 06 03 55 1d 11 04 82 .....0.. ...U....
Frame (928 bytes) Reassembled TCP (3327 bytes)
uTF8String (x509sat.uTF8Str...
Packets: 695 · Displayed: 695 (100.0%) · Marked: 3 (0.4%) · Dropped: 0 (0.0%) · Lo...
Profile: Defa

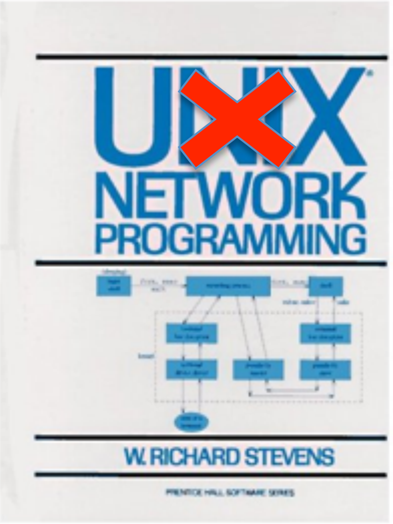
“block” parsers

- Cast “struct ip_hdr” over bytes
- Backtracking

“streaming” processors

- Byte at a time

Parsing the bytes



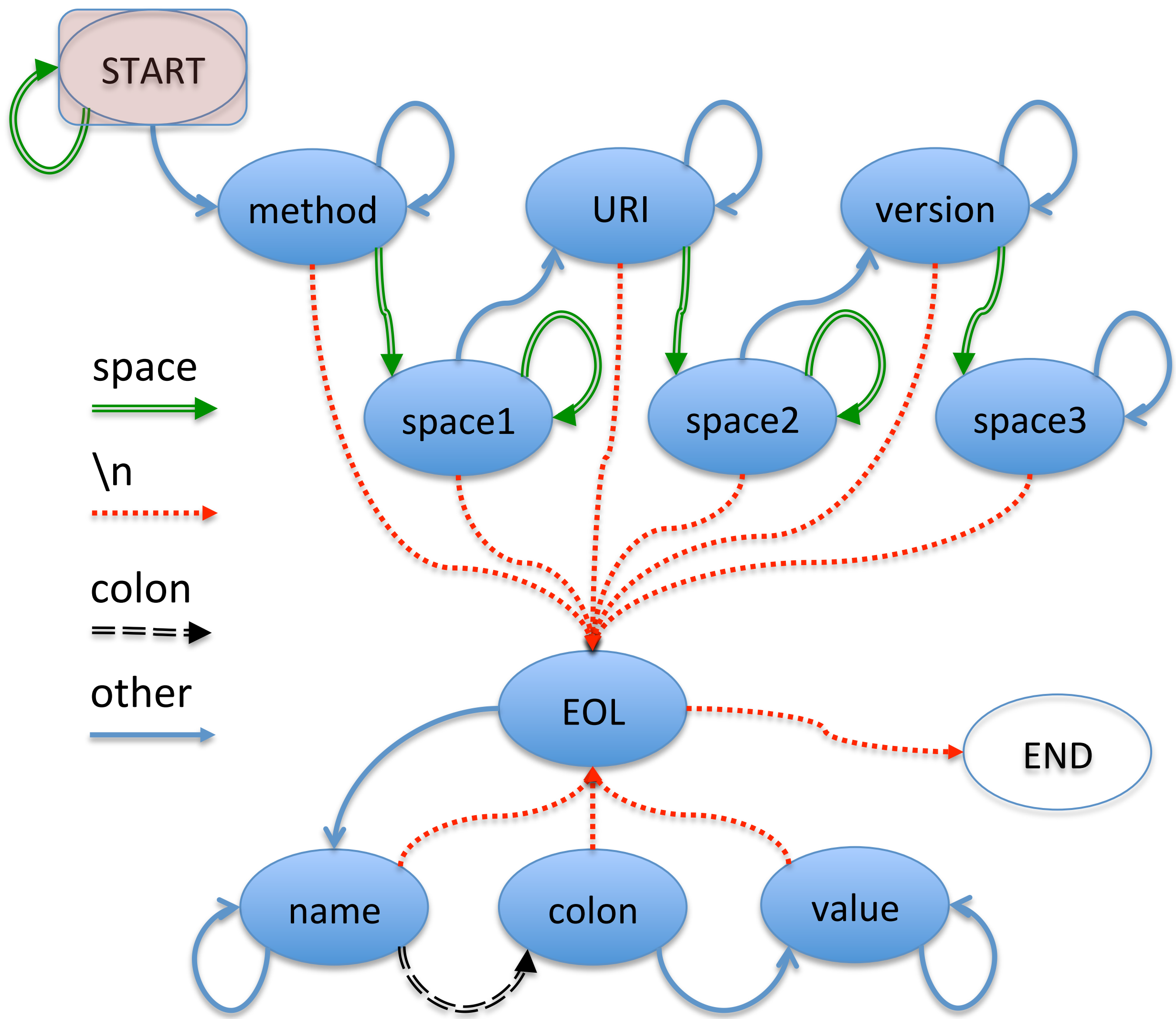
If you are using `ntohs()`,

you are doing it wrong

```
short port = ntohs(*(short*)p);
```

```
int port = p[0]<<8 | p[1];
```

GET /index.html HTTP/1.0
Host: www.example.com

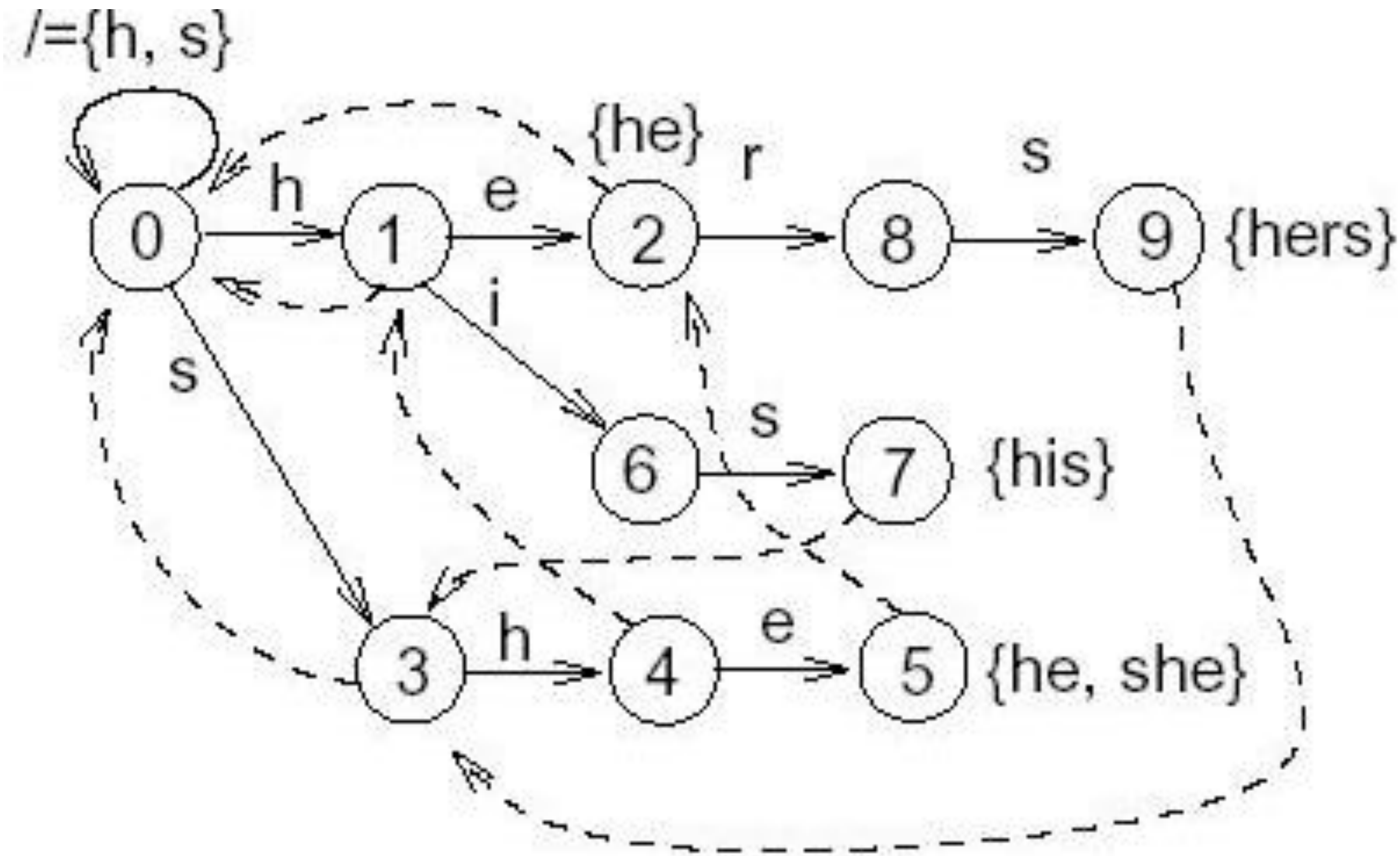


parser state machine

```
void http_parse(int *state, unsigned char c, ...)
{
    switch (*state) {
        case START: ...
        case METHOD: ...
        case URI: ...
        case VERSION: ...
        case VALUE: ...
        case NAME: ...
        case COLON: ...
        case SPACE1: ...
        case SPACE2: ...
        case SPACE3: ...
        case EOL: ...
    }
 }
```

```
void http_parse(int *state, unsigned char c, ...)
{
    switch (*state) {
        ...
        case METHOD: /*GET, POST, HEAD, ...*/
            if (c == '\n') {
                *state = EOL;
            } else if (strchr(WHITESPACE, c)) {
                *state = SPACE1;
            } else {
                ; /*no transition*/
            }
            ...
    }
}
```

alert tcp any any -> any 80 (content:"GET"; http_method;)



```
void http_parse(int *state, unsigned char c,
                int *ac, ...)
{
    switch (*state) {
        ...
        case METHOD:
            if (c == '\n') {
               search_end(xxMethods, ac);
               *state = EOL;
            } else if (strchr(WHITESPACE, c)) {
               search_end(xxMethods, ac);
               *state = SPACE1;
               search_start(xxSpaces, ac, c);
            } else {
               search_continue(xxMethods, ac, c);
               if (length++ > 128) …;
            }
            ...
    }
```

IDS and parser state machines

- “Do you reassemble TCP?”
- “Do you swap bytes?”
- “Do you handle data normalization?”

Properties

- Low memory
  - We don't buffer the HTTP method, we match the pattern as bytes arrive.
  - Don't need to reassemble TCP
- Robust
  - Can't have a buffer overflow in code that doesn't buffer things
  - All sorts of edge cases disappear
  - Gracefully "falls off the end" of a packet
  - Don't need to free stuff

Who does this?

- Most everyone in closed-source
  - IDS: Proventia, Intruvert, Palo Alto
  - Web: IIS
  - … and so much more
- Some open source
  - IDS: Snort, Suricata
  - ferret.googlecode.com
    - (this sucks BTW)

DFA Performance

- state = table[state][c];
- mov ebx[eax+ecx],%eax

One L1 cache hit per byte

- mov ebx[eax+ecx],%eax
  mov ebx[eax+ecx],%eax
  mov ebx[eax+ecx],%eax
  mov ebx[eax+ecx],%eax
  mov ebx[eax+ecx],%eax
-
- 3 GHz CPU with 3 Hz cache – 8-gbps

switch/case

- 15 Hz branch misprediction
- 0 Hz branch prediction

Special case

- Theory: byte-at-a-time
- Practice: grab multiple bytes that are part of the same state

[http-parse]

- https://github.com/robertdavidgraham/papers/blob/master/state-machine-perf/http-parse.c

[masscan]

root@supermicro1: ~/masscan

```
root@supermicro1:~/masscan# bin/masscan 0.0.0.0/0 -p80 --max-rate 30000000 --pfring
/etc/masscan/exclude.txt: excluding 3880 ranges from file

Starting masscan 1.0 (http://bit.ly/14GZzcT) at 2013-09-14 22:59:14 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 3508758232 hosts [1 port/host]
rate:25011.09-kpps, 56.72% done,   0:00:49 remaining, 0-tcbs,
```

https://github.com/robertdavidgraham/masscan

- SSL state machine parser:
- https://github.com/robertdavidgraham/masscan/blob/master/src/proto-ssl.c
- X.509 certificate parser:
- https://github.com/robertdavidgraham/masscan/blob/master/src/proto-x509.c