What's Up With HTTP?

Mark Nottingham

Principal Technical Yahoo! <mnot@yahoo-inc.com>
IETF HTTPbis WG Chair <mnot@mnot.net>

Agenda

- HTTP in Theory: The Standards
- HTTP in Practice: The Implementations
- New Stuff: Fixing the Suck

(poorly hidden) Agenda

- Inform what HTTP (the protocol) can do
- Inform what HTTP implementations can't (yet) do
- Encourage implementers to close the gap

HTTP in Theory: The Standards

(and some history)

HTTP circa 1996

- HTTP/0.9 fading quickly
- HTTP/1.0 taking off
- HTTP/1.1 to contain the damage
  - virtual hosting
  - persistent connections
  - caching
- HTTP-NG discussions already underway
  - binary (i.e. length-delimited headers)
  - generic
  - ...

HTTP circa 1996

- Typical use
  - Browser client, static or CGI content
  - GET, POST
- WebDAV: Glimmer in Whitehead's eye
- Services: huh?

2002: BCP56

- "On the use of HTTP as a Substrate"
- Brought about by new (ab)use; e.g., IPP
- Reasonable advice for the IETF community, but failed to foresee "services" and "Web 2.0"
- Codified distaste with non-browser uses
  - A new port for every app
  - Probably a new URI scheme too
- Currently being considered for deprecation

HTTP in 2009

- HTTP/2.0 didn't happen
- WS-* debacle unfortunately did
  - PEP turned into SOAP
- "RESTful" APIs
- Pressure to extend
  - Bidirectional communication (AJAX, BOSH...)
  - New Web protocols (OAuth, CORS...)
- Explosion of implementations
  - new servers, clients
  - new frameworks, APIs

- Interop is OK for "traditional" usage, but...
- More implementations = more variance
- Use cases are getting more exotic
- Extensions are proliferating
- Underlying design is poorly documented

HTTPbis: Why

- IETF Working Group to
  - incorporate errata
  - clarify ambiguities
  - document extensibility
  - improve interoperability
- I.e., writing the recipe down more clearly
  - Specifications need to outlive their creators
  - Align theory with reality
  - NOT to extend HTTP (but wait...)

HTTPbis: Who

- “Core” Implementers
  - Apache (editing), Microsoft, Mozilla, Apple, Opera, Curl, Squid, WinGate, Serf
- Extension Authors
  - MetaLink, OAuth, WebDAV, PATCH
- Large Web Operators
  - PayPal, Google, Yahoo!
- Security Experts
  - Adam Barth, Amit Klein
- The “Old Guard”
  - W3C, HTTP authors, URI authors

HTTPbis: What

- Problem: RFC2616 is 176 pages of text/plain
- Solution: split it up
  - p1: messaging
  - p2: semantics
  - p3: payload
  - p4: conditional requests
  - p5: ranges
  - p6: caching
  - p7: authentication

HTTPbis: fixing...

- Currently ~200 issues, like
  - **editorial**: ABNF conversion (no implied LWS)
  - **procedural**: Registries for status, methods
  - **security**: WS between header name and colon
  - **i18n**: Header charset and folding
  - **html5**: Is Content Sniffing allowed?
  - **protocol**: Really, only two connections?
  - **semantic**: What is a PUT response w/ETag?
  - **caching**: Is the method part of the cache key?

HTTPbis: Status

- Editors: Roy Fielding, Julian Reschke, Yves Lafon, Mark Nottingham
- Currently on draft -08
- Major rewrites in progress
  - p1 messaging
  - p5 caching
- “six months”
- Also informal place for discussion of new extensions, liaison with HTML5 work, etc.

HTTP in Practice: The Implementations

Implementations

- **Clients**
  - IE, Mozilla, Opera, Safari, wget, curl, serf, Perl, Python, Ruby, Java
  - Abstractions: XmlHttpRequest, Prototype.js, Flash APIs
- **Servers**
  - Apache, IIS, Lighttpd, Tornado, your router, phone and fridge
  - Abstractions: filesystems, CGI, WSGI, Rack, Servlet
- **Intermediaries**
  - Squid, Traffic Server, Blue Coat, ISA, HAProxy, L7 load balancers, firewalls
  - Not many abstractions (yet)
  - 20%-30% of Web traffic goes through a proxy
- **Caches** in clients and intermediaries
  - starting to show up in Python, Ruby...

HTTP Versions

- Most everything these days is HTTP/1.1, except...
  - Squid (full 1.1 coming)
  - wget
  - a few libraries
  - very old browsers, servers, libraries
- That's OK

Core Methods

- GET, POST - universally supported
- PUT, DELETE
  - A *few* clients can't generate (e.g., Safari2 XHR)
  - Intermediaries can be configured to block, but usually aren't (except the paranoid and mobile)
- Biggest limitation is W3C languages
  - XSLT, HTML forms
- Result: X-HTTP-Method header (Google) or query params (e.g., ?real-method=POST)

“Advanced” Methods

- OPTIONS
  - Hard to configure in servers
  - Isn’t cacheable... oops.
  - Result: only used for esoteric protocols (*DAV)
- Extension methods - FOO
  - A number of clients don’t allow (e.g., XHR)
  - Intermediaries often block (e.g., Squid, L4 balancers)
  - Result: This probably isn’t so horrible

URIs

- Mobile clients limit to as small as 256
- Browsers
  - IE: ~2k
  - The rest: really really big
- Intermediaries are OK up to about 4k; some go higher
- Servers can be configured (or replaced)
- Result: people putting queries in POSTs
  - application-specific and frameworks
  - frameworks doing this leads to gratuitous tunnelling
  - HTTPbis recommendation: 8k

Headers

- Some length limits (e.g. 20k total in Squid)
- Almost no-one handles line continuations
  - Result: effectively profiled out
  - Disallowed by latest HTTPbis changes
- Connection header control: not great
  - Result: extending protocol difficult
- Trailers aren't well-supported at all
  - Result: debug, status more difficult

Partial Content

- Content-Range / 206
- Biggest use: PDF
- Some caches don't store partial content
  - e.g., Squid
- Flash URL API can access ranges, but VideoPlayer, etc. don't use it
- Result:

```
$vidID = $_GET["vidID"];
$vidPosition = $_GET["vidPosition"];
```

Redirection

- Most* current browsers will preserve POST when they get a 307 Temporary Redirect
  - ... but not PUT or DELETE
  - ... and not a 301 or 302
  - * except Safari - it doesn't even do 307
- HTTPbis redefining 301, 302 to reflect reality

Connection Handling

- Browsers limited to two concurrent connections to each server
  - ouch!
  - Result: BATCH, hosting on multiple names, etc.
- Being fixed in HTTPbis
  - no particular limit
  - IE8 already running with this

Pipelining

- Clients
  - Only Opera does by default (lots of heuristics)
  - The brave can turn it on in Mozilla
  - A few libraries allow (e.g., Serf)
- Most intermediaries will be OK with it, but won't forward
- Many servers handle it just fine; a few don't
- Risks: interleaved or out-of-order responses
- Predominant use today: SVN (thanks to Serf)
- Result: "waterfall" of requests; CSS spriting

Cookies

- There is no cookie specification.
  - Netscape isn't complete
  - RFC2109 doesn't reflect current practice
  - Opera only major implementation of RFC2965
- Parsing raw dates is painful
  - `Set-Cookie: a=1; Expires=Thu, 24 July 2008 00:00:00`
  - requires special case handling
- Result: libraries required.
- New IETF Working Group contemplated

New Stuff
(a.k.a. fixing the suck)

Authentication

- Basic is interoperable, but not secure
- Digest is more secure, but not terribly interoperable
- Many newer requirements not addressed
  - Phishing
  - Delegated auth
- OAuth IETF Working Group
- "two-legged"
  - Other efforts still coalescing

Security Model

- Origin Header
- Strict Transport Security (STS)
- Content Security Policy (CSP)
- Cross-Origin Resource Sharing (CORS)
- Server auth without SSL?
- W3C may be starting a WG.

PATCH

- “Restful” APIs are starting to abuse PUT
    - “update that with this...”
- PATCH allows you to apply a diff to a resource
- Currently in IETF Last Call

Prefer Header

- Lets a client state what it wants;
  - Full content in response body
  - Status message in response body
  - No response body
- E.g., POST /order-handler
- Currently a (quiet) Internet-Draft

Link Header

- Under-developed part of the Web arch: typed links
- Advertise/discover links in HTTP headers
  - "this invalidates <foo>"
  - "the previous one is <bar>"
  - "edit this over at <baz>"
- In RFC2068, taken out of RFC2616
- In IETF Last Call

HyBi: Bidirectional HTTP

- "Short-Term" Solution: Comet
  - Long polling optimisations
  - Connection use hints
  - Intermediary coordination
- "Long term": WebSockets
  - New, very low-level protocol
  - Already in browsers
  - Likely to be an IETF WG very soon

Better Transport

- head-of-line blocking STILL an issue
  - Pipelining isn't well-supported, and doesn't completely solve the problem
- HTTP doesn't guarantee integrity
  - except with Content-MD5 (which no one does)
- HTTP over TCP sucks
  - on lossy links
  - on high latency links
  - on low bandwidth links

HTTP/2.0?

- Re-framing HTTP semantics onto better transport
- HTTP-over-SCTP (uDel, Cisco)
  - Better over long-distance / lossy nets
- WAKA (Roy Fielding)
  - Still probably TCP
  - Allow new message patterns, more efficient implementation and network use

Take-Aways

- Implementations are (obviously) usable, but
  - They sometimes impose arbitrary limits
  - They don’t expose some important controls
- HTTPbis is an opportunity to
  - get implementers together
  - clarify ambiguities
  - improve interop
  - make HTTP a more stable basis for the next 10+ years
- We need to start thinking about HTTP evolution NOW.