GNUnet: The Bad

GNUnet Hacker Meeting Day 1

Christian Grothoff

GNUnet e.V.

23.6.2018

The Plan

- **Saturday:** Discuss long-term strategy, open design issues, major bugs
- **Sunday:** Continue Mantis Bug-fixing/fighting session!
- **Monday:** CodeSonar (and if you can get it to work, Coverity) static analysis, improve CI
- **Tuesday:** Documentation and new Web site
- **Wednesday:** GNS Context and more bugfixing
- **Thursday:** Future technical direction and funding discussion

util/

- `epoll`-based event loop plugin, `poll`-based event loop plugin, glib-based event loop plugin; more testing of event loop and performance
- Build system switch for GNUnet as-a-library
- Service registration and internal message dispatch for GNUnet-as-a-library build option
- Use asynchronous DNS lookups in gnunet-resolver-service on platforms where this is possible

ATS

- By API, we currently only select ONE transport per peer; should not select transports (use all!), but only allocate bandwidth between transports and peers.
- MLP/RIL are extremely crash prone, horrific code quality
- proportional heuristic does not devalue increasing number of connections once we reached saturation point
- Auto-tune available bandwidth based on observations (may alternatively be in transport/NAT, the key point is that users should not have to manually set the ATS bandwidth option)

Transport

- Plugins should be moved into separate processes, connecting to transport service via an API
- Plugins should not have bi-directional semantics as a requirement
- Plugins should encrypt (ECDHE+AEAD), but without replay protection (uni-directionality!)
- testcases insufficiently systematic
- transport manipulation functions should be moved into a plugin-proxy to simplify code

DV

- Not working
- Should not use SET
- Should use new transport plugin architecture

NAT

- Various known NAT traversal algorithms not implemented
- Needs to implement method for detecting physical context (i.e. MAC of current router)
- Needs to trigger (via ARM?) re-start of peer with different PID on context switch to prevent tracking
- Improve auto-configuation logic, make sure it almost always works without any user configuration

HELLO

- HELLOs should have signatures per address
- Verification by checking signature (no more PING/PONG)
- Address should be human-readable (no more talking to plugins to convert to human-readable)

CORE

- Eliminate use of timestamps to detect replay issues (some systems do not have an RTC)
- Need to signal to other peers whether we are “infrastructure” (datacenter, high bandwidth) or on battery (mobile) and allow subsystems like DHT and CADET to use this information for more intelligent choices for relaying
- Replace TypeMap with list of subsystems and protocol version ranges

DHT

- Larger-scale benchmarking
- Understand performance limitations, including in datacache
- Auto-tune datacache heap size based on available RAM (current default is often tiny, users should not have to manually set option)
- Introduce signatures on DHT paths

CADET

- Implement, test and verify net2o-style congestion control
- Implement weaker semantics (low-latency/unbuffered/out-of-order)
- Larger-scale benchmarking and performance analysis
- Optimize timeouts, retransmissions, DHT usage, etc.

SET

- Not yet Byzantine fault-tolerant against stuffing attack

RPS

- sub-sampling (sampling peers from specified groups)
- thorough evaluation (leakage)
- (minor architectural improvements)

TOPOLOGY

- Not sure if we want to keep the current mixture of functions.
- Does not implement robust modern ways of maintaining a mesh; some of those require RPS first

FS

- Block size is too small in practice
- Datastore is bad at keeping quota set by user
- Service should be split into two: one running in user-context (what is currently libgnunetfs; for access to shared files, active downloads) and one for system-wide components
- Metadata and files are currently stored in a disassociated way, resulting in search results that then fail to download; keep blocks and metadata of files together
- Use GNS instead of SKS

Conversation

- Dropping silence on large buffers during replay to avoid accumulating latency
- Need to implement ring tones

Secushare

- new approach because current design and implementation is far from finished: missing features as in spec (e.g. state sync) and as in future requirements (ansync messaging) and too buggy (2 years of expirience)
- chat including image presentation
- multi-device support
- adaptive anonymous multicast
- asynchronous messaging (permanent online relay, seperation of multicast origin and channel owner, owner C&Cs origin)
- E2E encryption and forward/future secrecy for 1:1 and 1:m
- data base with query language (decentralized state)
- adequate programming language: favorites are rust and python

VPN/EXIT/PT

- Does not properly support fragments/fragmentation deltas between IPv4 and IPv6
- Exit discovery via RegEx should eventually be replaced with range query mechanism (once we have a(nother?) DHT that can do range-query)
- Deterministic assignment of EXIT IP addresses based on CADET port (shared secret) and source peer identity
- Maintain GNS names for EXIT IP addresses that are mapped to VPN records going to those PIDs and Ports.

Identity-provider/credential/identity-attribute

- Authorization token database (issued, received)
- ABE dependency review (currently libgabe)
- 3rd party credential support (e.g. x509 certs, SNARKs)
- OpenID-Connect JWT review (contents, algorithms)
- Revocation review (database for revoked tokens)

REST

- AGPL handler missing
- Case-insensitive headers (bug)
- Default local name for service (gnunet.api)
- Plugins for subsystems missing
- Partially addressed by GSoC

Web site

- Kill Drupal, generate HTML
- Need new front page
- Clearer introduction to the project