

**“Babel as PS Track WG”
a.k.a.
Major League
or
When the Going Gets Tough ...**

Tony Przygienda, Juniper Networks
& several participants in longish email threads

What's this all about ?

- In case of PS WG discussions are needed with reasonable level of participation
 - Document quality to meet Proposed Standard
 - Simulation/Serious Discussion about possible oscillations
 - “Reference implementation” vs. “Specification”
 - Future Extensibility provisions deserve considerations
 - What is Babel baseline ? What is optional and what is mandatory ?
 - What beyond Babel baseline should be in charter ?
 - The usual trade-offs:
 - A *Good Design Rule* for PS protocol is “you’re done if there’s nothing left to remove”
 - A contradicting *Good Design Rule* is also “put all the hooks in so people can extend it for next 20 years without forklifts”

“Reference Implementation” is not PS Specification and Specification is NOT “Implementation Guideline/Experience Report”

- Example 1: Something like “A node increments its sequence number (modulo 2^{16}) whenever it receives a request for a new sequence number” (in [Section 3.8.1.2](#)) led to faulty implementation and I also didn’t understand what it was supposed to be. This would have shown as interoperability problem in very subtle ways in 24x7 multi-vendor environment late in the game.
- Example 2: Rules for processing of updates are spread across the current RFC and some are out-of-date based on implementation (in Section 3.8.2.2). I could not figure out without help what is MUST/SHOULD/MAY based on the implementations neither could I implement the protocol correctly without looking @ the implementations (and even that seems not assured like the 3.8.1.2 issue)

Document Quality for PS; Examples

- Glossary:
 - Example: easy confusion due to loose use of id, router-id, neighbor-id
- Constant values that ensure interoperability “out-the-box”
- Strict numbering of clauses
 - Example: it is not easy to have a discussion about “how an update is processed precisely” and even to figure out whether all cases are covered
- Error handling of misformatted packets, especially due to complexity of format “compression”

Simulations & Oscillations

- A specification is normally prescribing a single metric/set of constants that work out the box
 - If multiple metrics are desired it must be guaranteed by PS that *any* combination of those in the network will work
- The “requesting of a new route” may lead to persistent oscillation and a PS would need to define the values/hysteresis beyond “keep the requests for a while and discard duplicates”

What's baseline and what comes later ?

- Should Babel even deal with Multicast ? If not, who will (in HomeNet at least) ?
- What about ECMP support ? Can that be patched in later ?
- Is security part of baseline PS ? (doesn't require merging security into the base document)

Other

- Format suggestions & observations
 - 32 bit metrics have proven a serious need in routing
 - 1 byte TLV length has proven a problem in routing
- Some ideas are so novel they need serious discussion/tightening
 - "simulation" of mandatory/transitive flags via "new TLV for everything" and "ignore all unknowns"
 - How will optional transitive even work in such scenario ?
 - Lack of error handling & exact description of the packet compression