

euro-IX



euro-IX

NWI-9 IN-BAND NOTIFICATION MECHANISM

Stavros Konstantaras
Network Engineer

RIPE79 DB-WG
17-10-2019

AMS-IX initiative



Aris Lambrianidis via db-wg 

22 March 2019 at 20:30

[db-wg] In-band notification mechanism?

To: db-wg@ripe.net



Dear all,

Back in the day, RFC1996 introduced the NOTIFY mechanism in DNS, which significantly helped with information propagation delay, as it facilitated the transition from a pull (poll) to a push (interrupt) model.

The problem we, as AMS-IX, are facing is quite similar when it comes to polling the RIPE database for changes. This seems inefficient.

Although the analogy breaks down quickly, as there are no RIPE database "clients" similar to DNS slave servers parsing NOTIFY messages, we would love to see any RIPE API created or extended, or any other mechanism implemented by which a client "registers interest" for any objects it wants to be notified of changes.

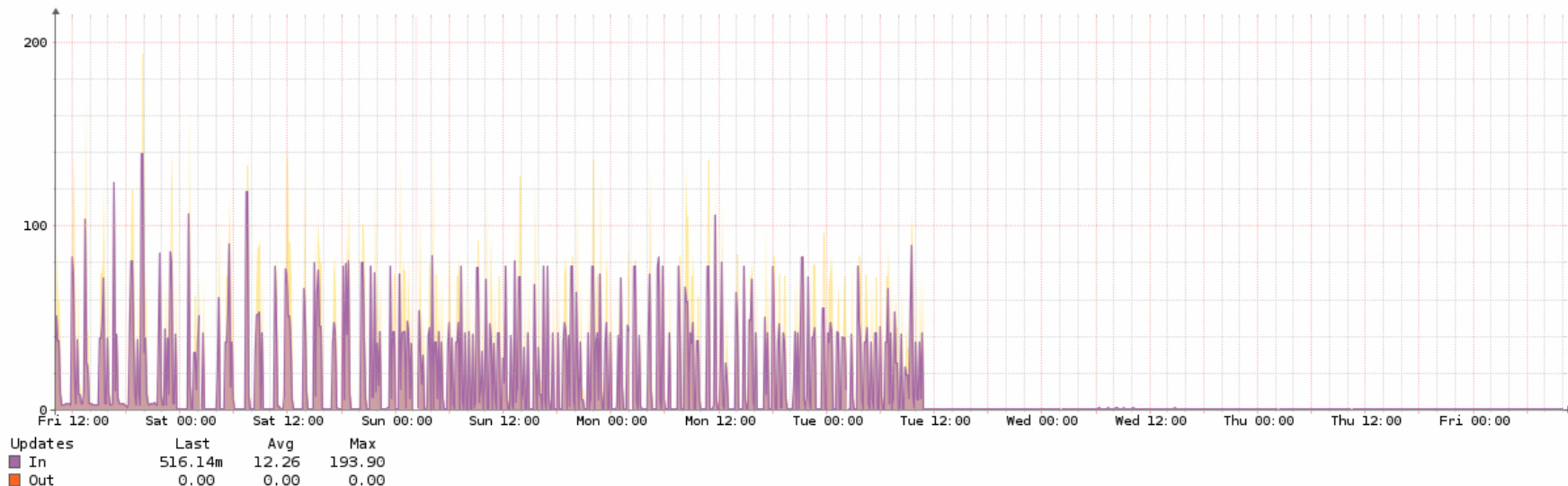
As a simple example, if we were to "register interest" (e.g. via a REST POST or PUT method) for the AS-AMS-IX-SET as-set object, we would be programmatically notified whenever the "last-modified" field of the as-set was changed.

Based on the above, I have 3 questions:

1. Does something like what is described above already exist?
2. If it doesn't exist, would others be interested on such functionality?
3. If it doesn't exist, while knowing that this is only a high level overview of the concept and many details are missing, is this generally feasible?

Kind regards,
Aris Lambrianidis
AMS-IX

Motivation #1



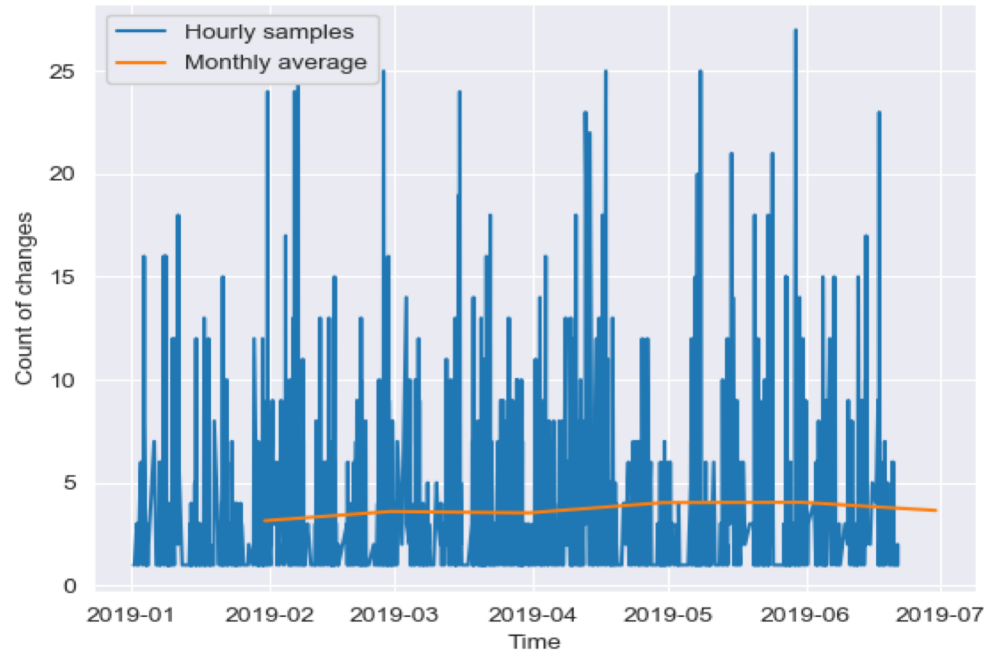
“The graph below shows the amount of updates we get from the route-servers (only). As you can see, we are regularly getting big chunks of updates from AMS-1 and AMS-2. These periods of updates last between 5 and 15 minutes and are at semi-irregular intervals as shown above...” (AMS-IX customer)

Motivation #2



- ***“Even though it makes little sense, could it be it was a temporary "leak"? I mean the change on our side was 16:09 UTC, looks like the export-via update script at your side picked up the aut-num change at 17:00 UTC...”*** (AMS-IX customer)

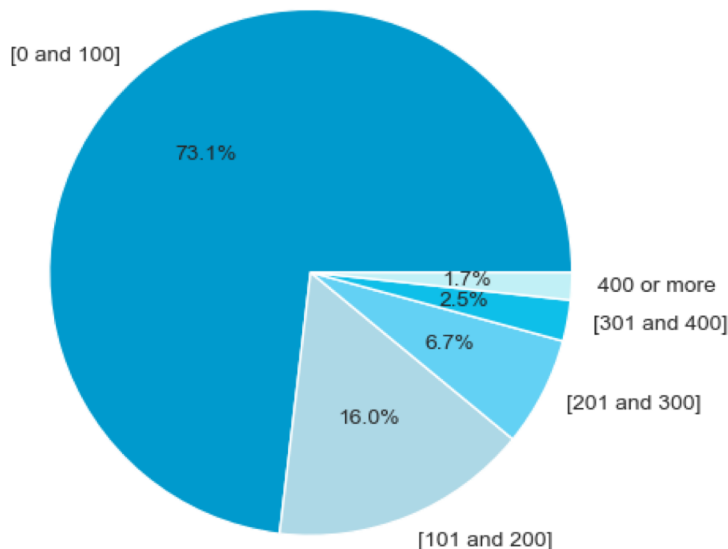
Rate of changes since 2019/01



How often did policy aut-num and route/route6 objects change on hourly basis, since 2019-01?

Fig 2: Changes in aut-num/rpsl objects since 2019-01 - without outliers

Distribution of amount of change ranges

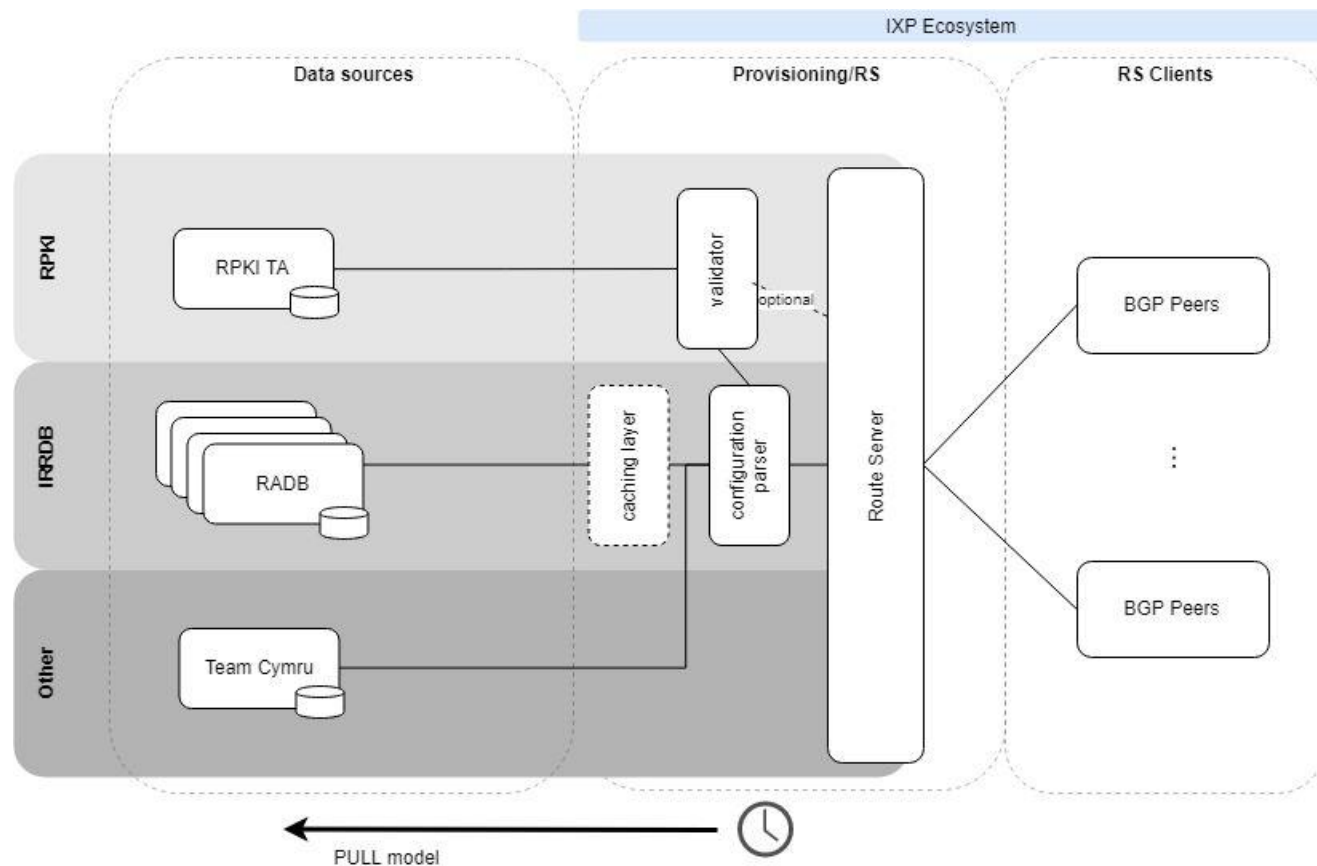


How many changes per hour are more frequent?

- ~ 73% of times the quantity of changes remain under 100 (refer to histogram).

Fig 3: Frequency distribution in aut-num/rpsl object changes since 2019-01

The model doesn't help



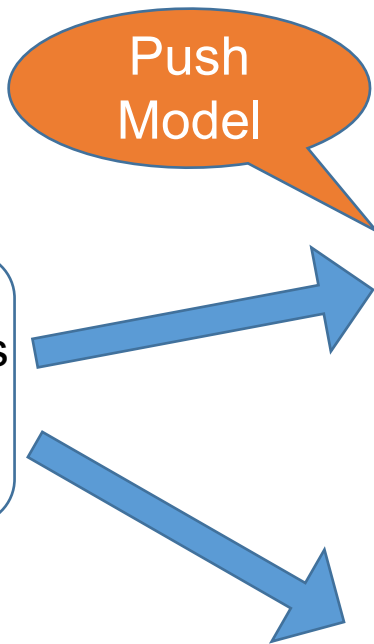
Requirements

Push
Model

- Having RSs always available
- Less BGP updates to customers
- Fast filtering update
- Fast BGP convergence time

Abandon Pull model

Abandon monolithic
configuration pipeline



To make the push model work



NWI-9

- **Subscription**

- People/Software agents sign-up for changes for objects that are interested for

- **Signaling**

- The server side notifies the client/subscriber about a changed that occurred

- **Transport**

- The server side delivers the deltas or the new object to the client side

But we have NRTM



- **Focuses on transporting data**
- **It's a workaround on top of TELNET**
- **Not stateful, not stateless**
(Requires an initial setup from the client)
- **Routing information objects are still subjected to RIPE memberships**
- **In any case, as RIPE engineers say, current NRTM is not suitable to solve NWI-9 (27 May 2019)**

Looking for solutions...



- **A topic based publish-subscriber model can be adopted by RIPE NCC**
 - People/Agents express their interest for a specific topic e.g. “changes of AS-SET AS-AMS-IX-RS”
 - When there is a change submitted for this topic, the People/agents receive a notification.
 - *Pub/sub provides the opportunity for better scalability than traditional client-server, through parallel operation, message caching, tree-based or network-based routing (Wikipedia)*

Three approaches to go further



- **Langzaam (slow)**
- **Snel (fast)**
- **Te snel (too fast)**

- **Replace NRTM with a new protocol (similar to RRDP) that accommodates all 3 requirements.**
 - We need to deal with the legal part first
 - We need to design the new protocol (if needed), prototype, test, standardize, etc.
 - But is a future-proof solution

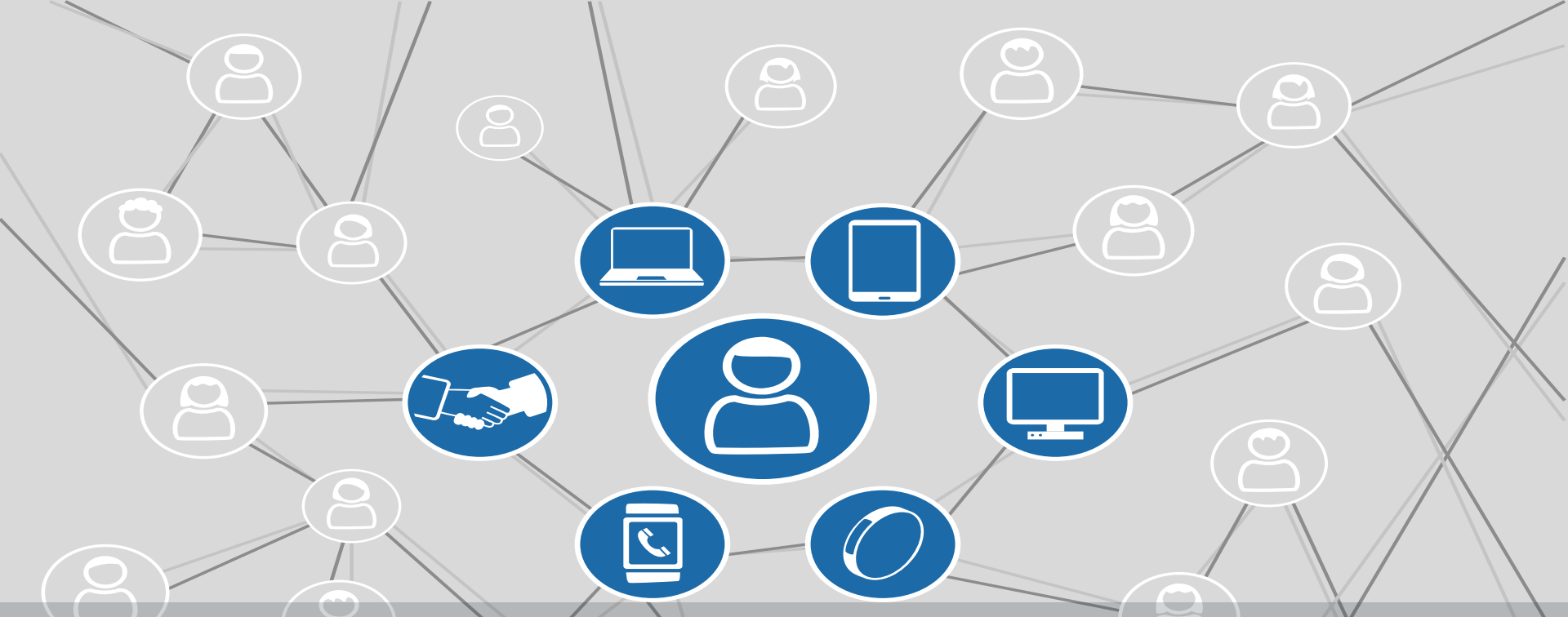
- Implement a replacement NRTM **interface**, based on modern standards
 - The interface is generally available without needing to be a member, or to sign an agreement.
 - The protocol uses HTTPS and WebSockets, with object data in JSON format.
 - A client can request the current available range of updates, and also request a closed or open range of updates.
- There is a concern of having RIPE engineers doing double work.

- **We consume Google's(?) existing Pub/Sub infrastructure**
 - RIPE engineers just write a connector that sends the updated object to Pub/Sub
 - Clients receive it and trigger filter automation immediately
 - Signaling part is solved, we still need to use NRTM and/or whois for the transport part
- Drawback: Google Positive: We can scrap it easily

It's a common issue



- **During the last EURO-IX RS workshop**
 - Many IXP members expressed similar (bad) experiences and issues
 - Many members expressed demand from their customers for faster and more accurate RS filters
- **We decided:**
 - NWI-9 has the official support of the EURO-IX community
 - The community foresees value on this item and asked RIPE NCC to schedule an implementation on that.



Stavros Konstantaras (AMS-IX)
- stavros.konstantaras@ams-ix.net
Emil Palm (NETNOD)
- emil@netnod.se

euro-IX