

Service Discovery for IoT devices through PSync FullSync

9th NDN Hackathon

Saurab Dulal, Muktadir R Chowdhury, Laqin Fan,

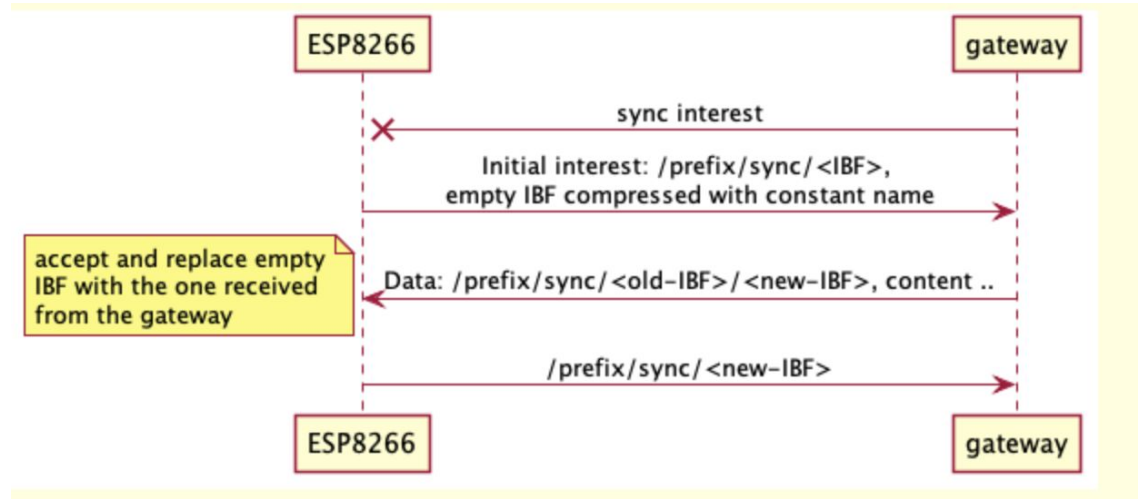
Why Service Discover for IoT using sync?

- In sensor networks, 100s of devices need to discover gateways for the future communication.
- Discovery can be realised in term of publisher-subscriber model
 - Some devices wants to publish, others wants to subscribe, or some wants to do the both
- Thus, any sync protocol can be used as an underlying mechanism for the discovery process
- Why choose PSync?
 - Completely implemented, documented and available for use
 - Expand the existing PSync code for multiple use case

Problems with current PSync implementation for IoT

- It doesn't efficiently support an application to be a full consumer only - one who doesn't produce any data
- Harder to deploy PSync in low constraints IoT devices (consumer only) because of its intensive IBF related computation (compression, hashing, insertion) - mostly not needed if no data is produced

Design of FullSync Consumer for IoT



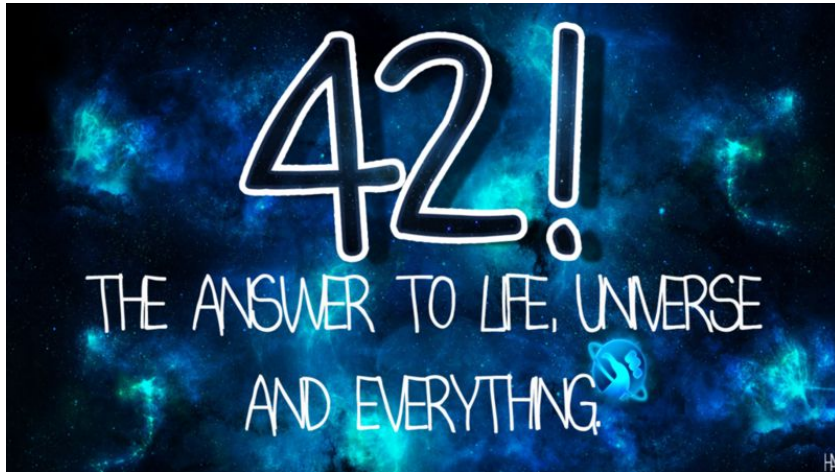
Details here: <https://redmine.named-data.net/issues/4987>

Deployment approach

Out of few available options () we choose ndn-lite () - Alex A
Suggested

What we did?

“Learning basics of ndn-lite, debugging and understanding error code”



It works on my machine...

Also, what we did?

- Finalized the design
- Developed quick and dirty PSync fullsync consumer for POC in C++/ndn-cxx and ported it into ndn-lite © [porting not working fully]
- Add segment fetcher feature for fullSync consumer in ndn-lite.
 - Send interest to retrieve segment
 - Stop sending interest until the last data name component (sequence number) is equal to the data FinalBlockId.

Snapshots

```
INFO: [examples.FullSyncConsumerApp] Received: /a-0/1/1
INFO: [examples.FullSyncConsumerApp] Received: /a-1/1/1
INFO: [examples.FullSyncConsumerApp] Received: /a-0/2/2
INFO: [examples.FullSyncConsumerApp] Received: /c-1/1/2
INFO: [examples.FullSyncConsumerApp] Received: /c-0/1/2
INFO: [examples.FullSyncConsumerApp] Received: /a-1/2/2
INFO: [examples.FullSyncConsumerApp] Received: /a-0/3/3
INFO: [examples.FullSyncConsumerApp] Received: /c-0/3/3
INFO: [examples.FullSyncConsumerApp] Received: /c-1/3/3
INFO: [examples.FullSyncConsumerApp] Received: /a-1/3/3
```

Full Consumer node receiving all the published data by the producer

```
INFO: [examples.FullSyncApp] Publish: /a-0/1
INFO: [examples.FullSyncApp] Update /c-0/1
INFO: [examples.FullSyncApp] Update /c-1/1
INFO: [examples.FullSyncApp] Publish: /a-1/1
INFO: [examples.FullSyncApp] Update /c-1/2
INFO: [examples.FullSyncApp] Update /c-0/2
INFO: [examples.FullSyncApp] Publish: /a-0/2
INFO: [examples.FullSyncApp] Update /c-0/3
INFO: [examples.FullSyncApp] Publish: /a-0/3
INFO: [examples.FullSyncApp] Publish: /a-1/2
INFO: [examples.FullSyncApp] Update /c-1/3
INFO: [examples.FullSyncApp] Publish: /a-1/3
```

Producer node publishing data

(near) Future Work

- Complete the porting of full consumer in ndn-lite and release the code
- Integrate it in esp8266ndn library as it support ndn-lite.
- Develop a sync protocol adaptor to support various sync protocol (similar to what NLSR is doing right now)

FullSync Consumer source code

C++/ndn-cxx: <https://gerrit.named-data.net/c/PSync/+/5670>

C/ndn-iot-package: <https://github.com/alvyC/ndn-iot-package-over-posix>

C/ndn-lite: <https://github.com/dulalsaurab/ndn-lite/tree/abc>

Redmine: <https://redmine.named-data.net/issues/4987>

Code not available in hackathon-github due to some technical issue, we will upload there soon.

Thank you