

NSI Protocol

Clients, NATs, Firewalls & More

NSI Workshop, OGF34, Oxford
Henrik Thostrup Jensen, NORDUnet

Overview

Current protocol is not very firewall/NAT friendly

- Callbacks are main the problem

- URL generation can be tricky as well

- Isn't very application friendly

- NATs and Firewalls are the norm

Furthermore our protocol is quite complex to implement

This means our protocol isn't very client friendly

Two solutions

- A: Changing the NSI protocol

- B: Adding a dedicated client protocol

 - Would do roughly the same as the NSI protocol

- I think having two protocols is silly

NAT/Firewall friendliness

Client must initiate connections

Alternative state update mechanism needed

There are several options for this

- John presented most options in his presentation

- I'm partial to long-polling

 - Mailbox adds complexity to service implementation

 - Long-polling is connection-oriented

 - A client will only get updates while connected

 - This removes a lot of state management

 - Simplifies implementation a lot

Protocol strategy

Long poll after sending request

- One connection per reservation is bad idea though

Subscribe to multiple connections

- Requires initial states, and then subsequent updates

- One request - multiple responses

- I'm not fully sure how to do this in SOAP over HTTP

Further Protocol Issues

SOAP/WSDL support is sparse outside Java

- For most of us, this isn't a problem

- Not really for me either (SUDS is okay, but not great)

But it makes the bar for using NSI higher

Switching to HTTP+XML (REST) lower the bar a lot

- All big programming languages have support for these

- We can still use the XSD schema for reservation creation payload

- Would need different method for initiating state change

- Connections will have a URL, easy state retrieval and query (GET)

Decision points

Do we switch away from the callback?

Which strategy for state updates?

Change the protocol technology to something more client-friendly?