

Your File System on IPv6

Simon Wilkinson sxw@your-file-system.com

Overview

- Introduction
- Protocol
- Implementation

Introduction

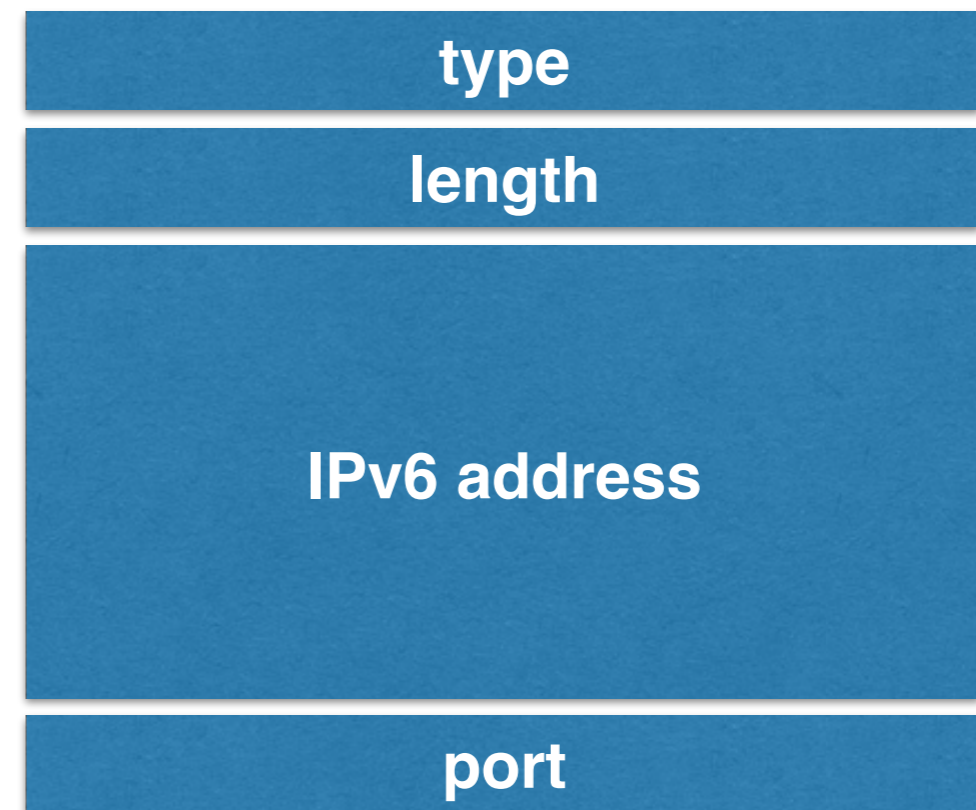
- Incremental Approach
- Disabled in first release
- Need to hear demand from customers

Protocol

- YFS uses private service IDs to enable us to innovate quicker than the standardisation process allows
- All IPv6 features deployed on private service IDs
- Happy to share protocol details in hope OpenAFS will benefit from our experiences

What is an endpoint?

```
ext-union endpoint
  switch (afs_int32 type) {
    case ENDPOINT_UDP_IPV4:
      afs_int32 host;
      short port;
    case ENDPOINT_UDP_IPV6:
      opaque addr[16];
      short port;
  }
```



RX Implementation

- Significant modifications to connection establishment and management API
- Internally, RX pretty much completely rewritten

Protocol Changes

- Replace use of int32 for addresses with endpoints
- Remove interface addresses from RXAFSCB_TellMeAboutYourself
- Forward a volume to a UUID, rather than an IP
- Create U variants of all VL RPCs

Configuration

- Rewritten afsconf package
- krb5.ini configuration files

```
[cells]
  example.org = {
    description = "An example cell"
    servers = {
      test.example.org = {
        addr = 192.168.0.1
      }
      udp/www.your-file-system.com:7003 = {
        type = vlserver
      }
      [ff00::10.10.0.3]:1025 = {
        type = ptserver
        clone = yes
      }
    }
  }
```


Implementation: vldb

- Initially overloaded existing vldb address records
 - Limited to two IPv6 addresses per server
- Production implementation reworks the vlserver database to remove this limit

Implementation: ubik

- “Lowest IP” is always an IPv4 address, if there is one
- Multi-homed ubik configuration simplified
- For maximum compatibility all ubik servers need both v4 and v6 addresses

Implementation

- Host package completely rewritten
- vos modified to handle multi-homed machines