
CERN

Michal Simon

XRootD – Releases, Status & Planning 2019

4.9.0 review

- Currently release candidate 4 is under testing
- Release Notes are available at:
  https://github.com/xrootd/xrootd/blob/stable-4.9.x/docs/ReleaseNotes.txt
- Big release that accumulated lots of new features both on client and server side

4.9.0 review

Client: **redirect trace back**:

export **XRD_LOGLEVEL=Debug**
xrdfs localhost stat /data/file.dat

...

[Debug][XRootD] **Redirect trace-back**:
[Debug][XRootD] **0. Redirected from: root://localhost:1094/ to: root://172.18.0.3:1094/**
[Debug][XRootD] **1. Redirected from: root://172.18.0.3:1094/ to: root://172.18.0.4:1094/**
[Debug][XRootD] **2. Redirected from: root://172.18.0.4:1094/ to: root://172.18.0.3:1094/**
[Debug][XRootD] **3. Redirected from: root://172.18.0.3:1094/ to: root://172.18.0.5:1094/**

4.9.0 review

Client modern **declarative API**.

- C++11 friendly
- Chaining of asynchronous requests (significantly reduces the amount of boilerplate code)
- Standard response handlers: functions, functor, **lambdas**, **std::futures**, std::packaged_task
- Utilities to execute pipelines in **parallel** and **forward** values within a pipeline

4.9.0 review

Client modern **declarative API**, example:

```
1
2   File f;
3   std::future<ChunkInfo> resp;
4
5   // open, read from and close the file
6   Pipeline p  = Open(file,url,OpenFlags::Read)
7               | Read(file,offset,size,buffer) >> resp
8               | Close(file);
9
10  auto status = WaitFor(p);
11
```

4.9.0 review

Client modern **declarative API**, example:

```
1
2   auto &&o1    = Open(file1,url1,OpenFlags::Read);
3   auto &&o2    = Open(file2,url2,OpenFlags::Read);
4   auto &&o2    = Open(file2,url2,OpenFlags::Read);
5
6   // open 3 files in parallel
7   Pipeline p  = Parallel(o1,o2,o3);
8
9   auto status = WaitFor( p );
10
```

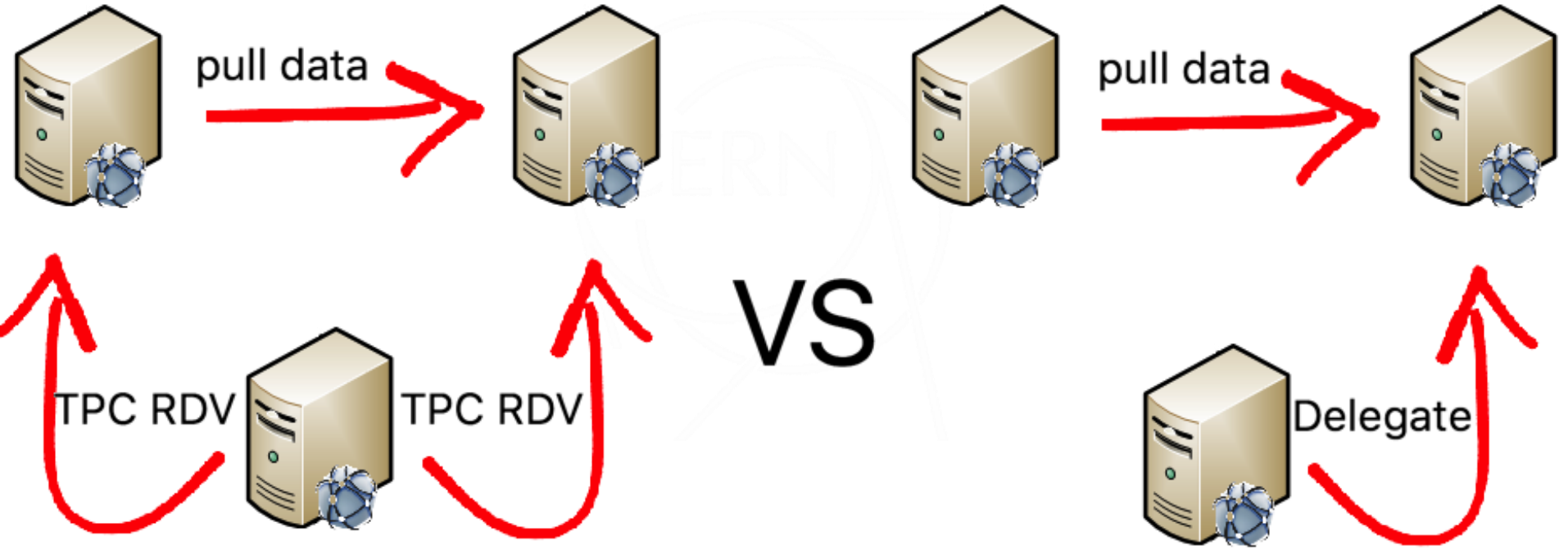
4.9.0 review

Third-party-copy with delegation

- xrdcp --tpc delegate only
- If destination supports delegation there is no need to set up a *randezvous* between source and destination
- If server does not support delegation we fall back to the standard TPC algorithm

4.9.0 review

Rendezvous TPC vs TPC with delegation



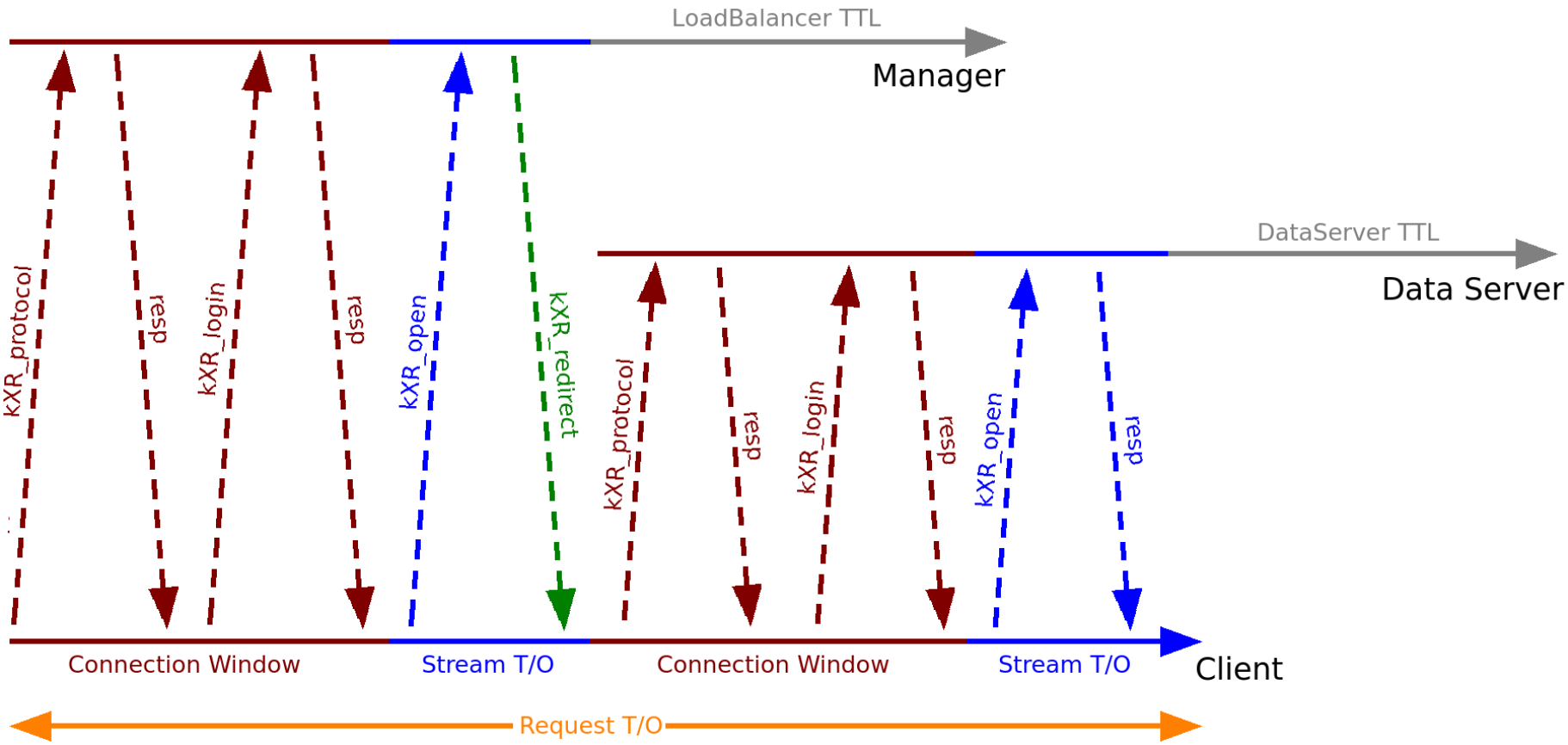
4.9.0 review

Other client enhancements:

- Protocol level plug-in (e.g. davix based HTTP plug-in)
- **State redirections** (e.g. on read, on write etc.)
- An API to **force disconnect**
- **writev** request
- Client documentation

4.9.0 review

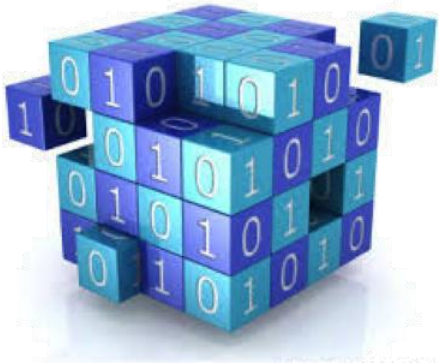
Client documentation (timeouts):



Plans: 5.0.0

- Planned for end of Q2 / beginning of Q3 2019
- Major release: an opportunity to review **ABI** and protocol constraints
- **Encryption**
- **Extended attributes** (already available in xrdR5 branch)

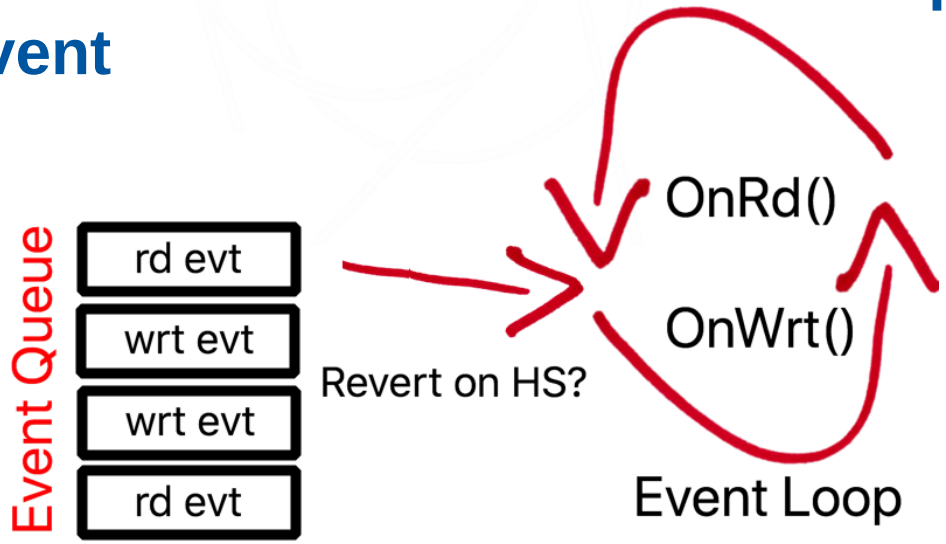
5.0.0: encryption



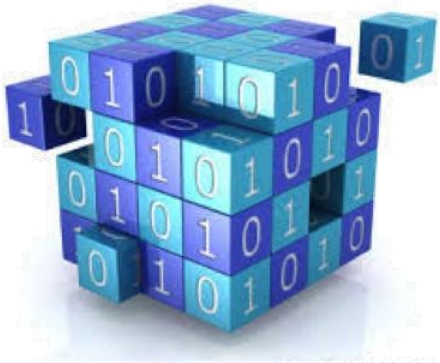
- Under development in **xrdtls** branch
- Requires ABI changes on the server side (reason for 5.0.0)
- Like HTTPS uses **Transport Layer Security**
  - **standard** and non-controversial,
  - independent from authentication method
- It is a prerequisite to support **security tokens** in xroot protocol

5.0.0: encryption

- Implementation is based on **OpenSSL asynchronous API** and an **event-loop**
  - TLS **hand-shake is carried out automatically**, however it might require reissuing a **read operation on write event** and vice versa a **write operation on read event**



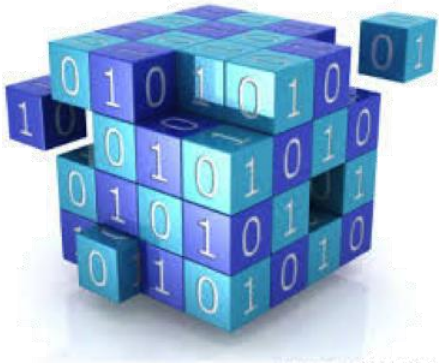
5.0.0: encryption



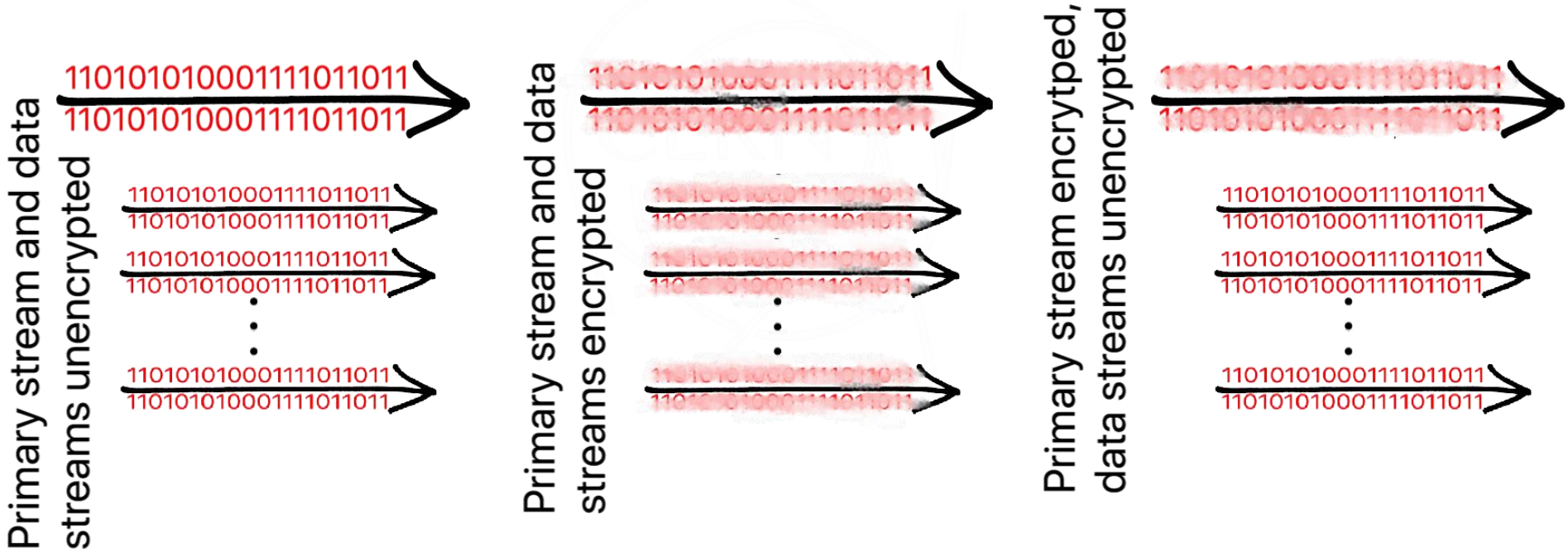
Design details:

- In order to **simplify operations** encrypted and unencrypted traffic uses **the same port**
- Encryption might be enforced by:
  - user by choosing **xroots/roots protocol**,
  - server by setting **wantTLS flag** in protocol response
- For performance reasons might be switch only after authentication
- Possibility to **encrypt only headers** and **transfer data unencrypted** (a better alternative for request signing)

5.0.0: encryption



Three possibilities:



Plans 5.x.x

5.x.x series (post R5 top priorities)

Client/Server:

- Bundled requests
- Recursive delete, extended stat, delete on close
- Partial response handling (streaming copy, ls)
- New TPC: TPC authentication, kXR_getfile / kXR_putfile

Plans 5.x.x

5.x.x series (nice to have)

Client:

- Extended ZIP support (append, compression)
- Extending testing infrastructure: **mock event-loop**
- Modernize code base (c++11)
- RDMA support (XCache for HPC use case)
- Native dynamic source selection

Packaging:

- Docker, cmake/pkg-config module

Summary

- **4.9.0** – soon (TPC delegation, force disconnect, redirect traceback, …)
- **4.10.0** – if need be
- **5.0.0** – encryption, extended attributes
- **5.0.1** – it would be nice to fit one more feature release into 2019 (especially request bundling)

Questions?