The Building-Block Approach to RM Standardization.

Mark Handley
ACIRI
mjh@aciri.org

Goals

- **To standardize building blocks for RM protocols.**
- **To standardize RM protocols built from these building blocks.**
- **To allow new RM protocols to be standardized later with minimal effort.**
- **To allow new (presumably better) building blocks to be standardized later to replace existing ones.**

The Problem

- **Can we suitably subdivide the functionality:**
  - to produce useful building blocks
  - to not add significantly to the complexity or overhead of the protocols built from the building blocks.

Strawman Target Protocols

- **Can we design building blocks that let us build:**
  - SRM (Scalable Reliable Multicast)
  - RMTP-II (Reliable Multicast Transport Protocol - II)
  - PGM (PGM)
  - RMDP/RLC (Reliable Multicast Distribution Protocol)

- **Without adding too much to the complexity of each?**

Application Assumptions

- **For now, we're primarily focusing on one-to-many transfer of large quantities of data.**
  - This constraint is imposed by what we know how to do with congestion control.
- **Where possible, building blocks shouldn't prohibit other applications though.**

Application Assumptions (2)

- **We're designing protocols that transport a series of ADUs.**
  - It's up to the application to fragment data into ADUs for transport.
  - The application is aware of the data naming scheme.
- **These are ALF ideas, but don't prohibit implementation as a separate middleware/transport layer, although in such cases they do imply that the API is fairly rich.**

Simple RM Taxonomy

- **Nack-based end-to-end (eg SRM)**
  - Easily deployed, fairly scalable.
- **Tree/Ack-based (eg RMTP-II)**
  - Moderate to deploy, controllable, higher delivery assurance, less scalable.
- **Nack-based with router support (eg PGM)**
  - Hard to deploy, not for public internet, high performance
- **Layered (eg RMDP/RFC, DF)**
  - Easy to deploy, very scalable, for a limited range of apps.

Commonalities

- **1. Nack-based end-to-end (eg SRM)**
- **2. Nack-based with router support (eg PGM)**
- **3. Ack-based with tree-based feedback aggregation (eg RMTP-II)**
- **4. Layered with no feedback (eg RMDP/RLC, DF)**

  - 1 and 2 have much in common in terms of mechanisms (although they differ greatly in network requirements). Differ primarily in feedback suppression.
  - 1, 2, and 3 share common FEC mechanisms, and much of congestion control. Perhaps common data packet formats, and data naming scheme.
  - 4 is significantly different from all the rest.

Strawman Building Blocks

- **Tree-based ACK mechanism.**
- **NACK mechanism**
- **End-to-end NACK suppression mechanism.**
- **Router-based NACK suppression mechanism.**
- **Data naming and framing mechanism.**
- **Scalable session messages mechanism.**
- **Packet-level FEC mechanism.**
- **Congestion control mechanism.**
- **Security mechanisms.**

Still research, possible future building blocks:

- **Local Recovery Mechanism**

1. Nack-based end-to-end

- **Standardize:**
  - Data Packet formats.
  - Nack Packet formats.
  - Session Message format.
  - Data Naming scheme.
  - Nack suppression scheme.
  - FEC Scheme.
  - Congestion Control Scheme.

2. Nack-based with PGM router support

- **Standardize:**
  - Data Packet formats (as 1.)
  - Nack Packet formats (as 1.)
  - NCF Packet format
  - SPM Packet format
  - Session Message format (as 1.)
  - Data Naming scheme (as 1.)
  - Nack suppression scheme. (some commonality with 1.)
  - FEC Scheme (as 1.)
  - Congestion Control Scheme (much commonality with 1.)

3. Tree/Ack based.

- **Standardize:**
  - Data Packet formats (as 1,2.)
  - Ack Packet formats.
  - Session Message format (as 1,2.??)
  - Data Naming scheme (as 1,2.)
  - Ack aggregation mechanism.
  - Tree formation mechanism.
  - FEC Scheme (as 1,2.)
  - Congestion Control Scheme (much commonality with 1,2.)

4. Layered

- **Standardize:**
  - Data Packet formats (small amount of commonality with 1,2,3)
  - Session Message format (as 1,2,3.??)
  - Data Naming scheme (possibly some commonality with 1,2,3.)
  - FEC Striping Scheme (different from 1,2,3)
  - Congestion Control Scheme (different from 1,2,3)

Summary

- **We probably need to standardize (at least) four protocols.**
- **There's a lot of commonality between three of the four.**
- **Is there enough to make it worthwhile aiming for standardizing the building blocks separately from the protocol, and then standardizing the protocols using these building blocks?**
  - The only aspect of this which is still research is congestion control.