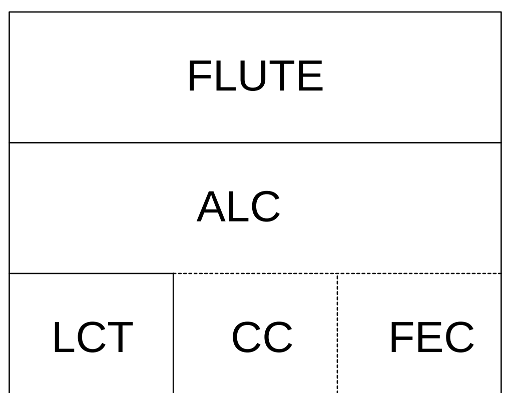
Analysis of the FLUTE Data Carousel

FLUTE Basics

- **File Delivery over Unidirectional Transport (FLUTE) protocol**
  - RFC 3926 [1]
- **To unidirectional delivery of files**
  - Files are delivered as transport objects, with optional content encoding (e.g. gzip)
- **Distribution to large number of receivers**
  - No feedback from receivers
- **Supports IPv4 and IPv6**
  - No IP version specific parts in the FLUTE header
- **Can be used with both multicast and unicast User Datagram Protocol (UDP) delivery**
  - Particularly suited to multicast networks
- **Supports Any Source Multicast (ASM) and Source Specific Multicast (SSM) models**
- **Reliability through retransmissions and Forward Error Correction**

FLUTE Building Block Structure

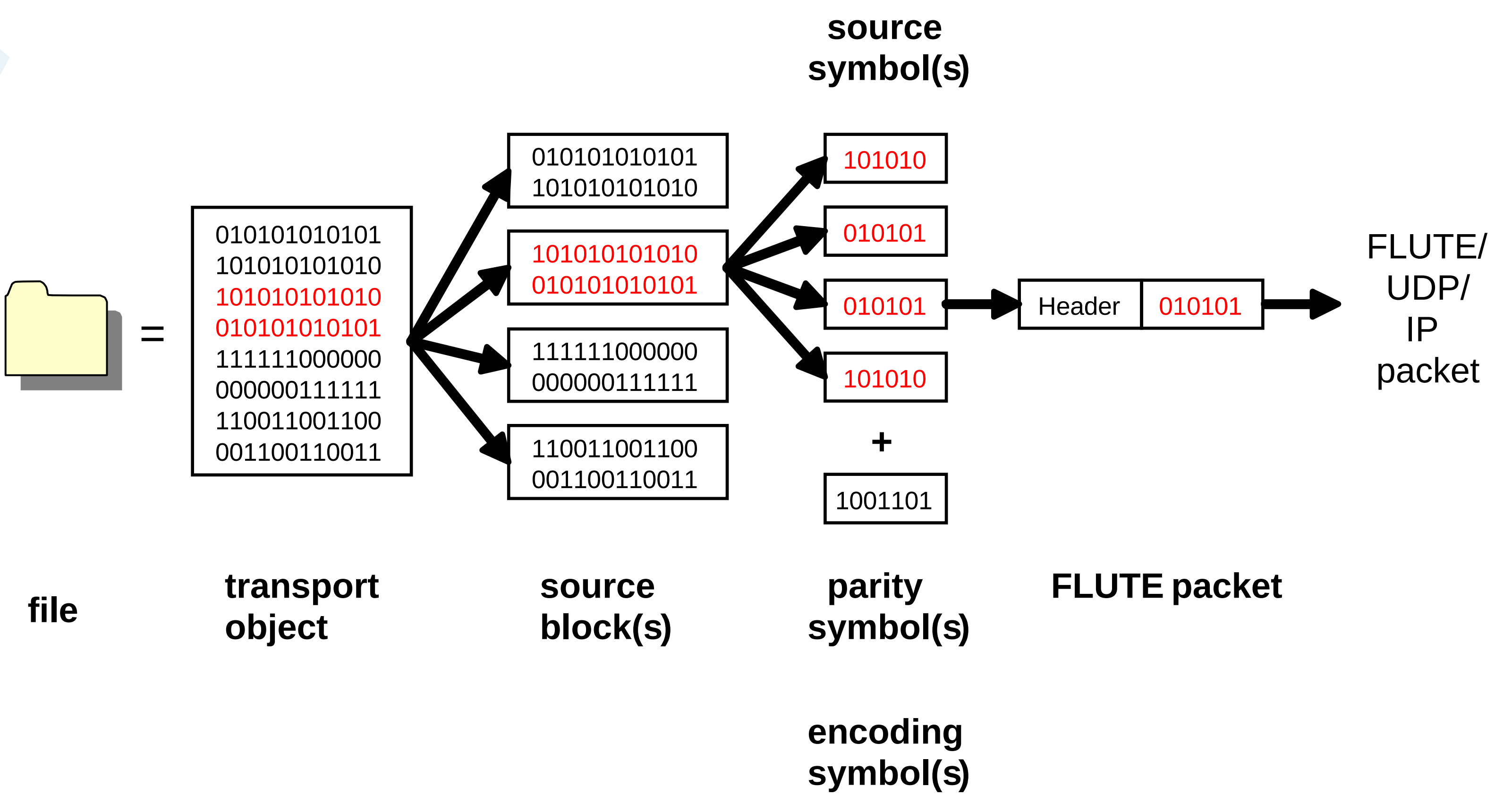
- **FLUTE builds on Asynchronous Layered Coding (ALC) Protocol Instantiation [2] of the Layered Coding Transport (LCT) Building Block [3]**
- **ALC combines the LCT Building Block, a Congestion Control (CC) Building Block and a Forward Error Correction (FEC) Building Block [4] to provide congestion controlled reliable asynchronous delivery**
- **LCT provides transport level support for reliable content delivery and stream delivery protocols**
- **The use of CC and FEC building blocks with FLUTE is optional**



Forward Error Correction

- **Parity data to recover from packet losses**
- **The use of Forward Error Correction in reliable multicast is defined in RFC 3453 [5]**
- **With FLUTE the default FEC code is Compact No-Code FEC [6]**
  - No actual FEC encoding or decoding
  - File segmentation into source blocks
  - FLUTE's algorithm for computing source block structure
    - Computes a source block structure so that all source blocks are as close to being equal length as possible
    - First number of source blocks share same larger length
    - Remaining second number of source blocks share the same smaller length
  - Encoding symbols contains only the source symbols
- **Others: XOR, Reed-Solomon, LDPC, Digital Fountain**

Building up a FLUTE Packet



FLUTE Session

- **A FLUTE session (i.e. an ALC/LCT session) consists of one or more ALC/LCT channels**
  - Defined by the combination of a sender's IP address and an address associated with the channel by the sender
  - Congestion control and multiple rate delivery by using multiple channels
- **A receiver joins a channel to start receiving the data packets sent to the channel by the sender, and leaves the channel to stop receiving data packets from the channel**
- **File Delivery Table (FDT) Instances together with FLUTE header fields give the necessary parameters to identify, locate and restore the files at the receiver**
- **Sessions may be started without complete knowledge of their content**
  - Requires FDT Instances to gradually define file parameters during the session

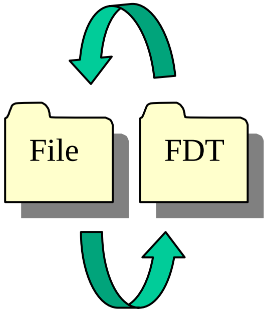
File Delivery Table Instance

- **One FDT Instance can describe all or part of the files for the FLUTE session**
- **An example FDT Instance:**

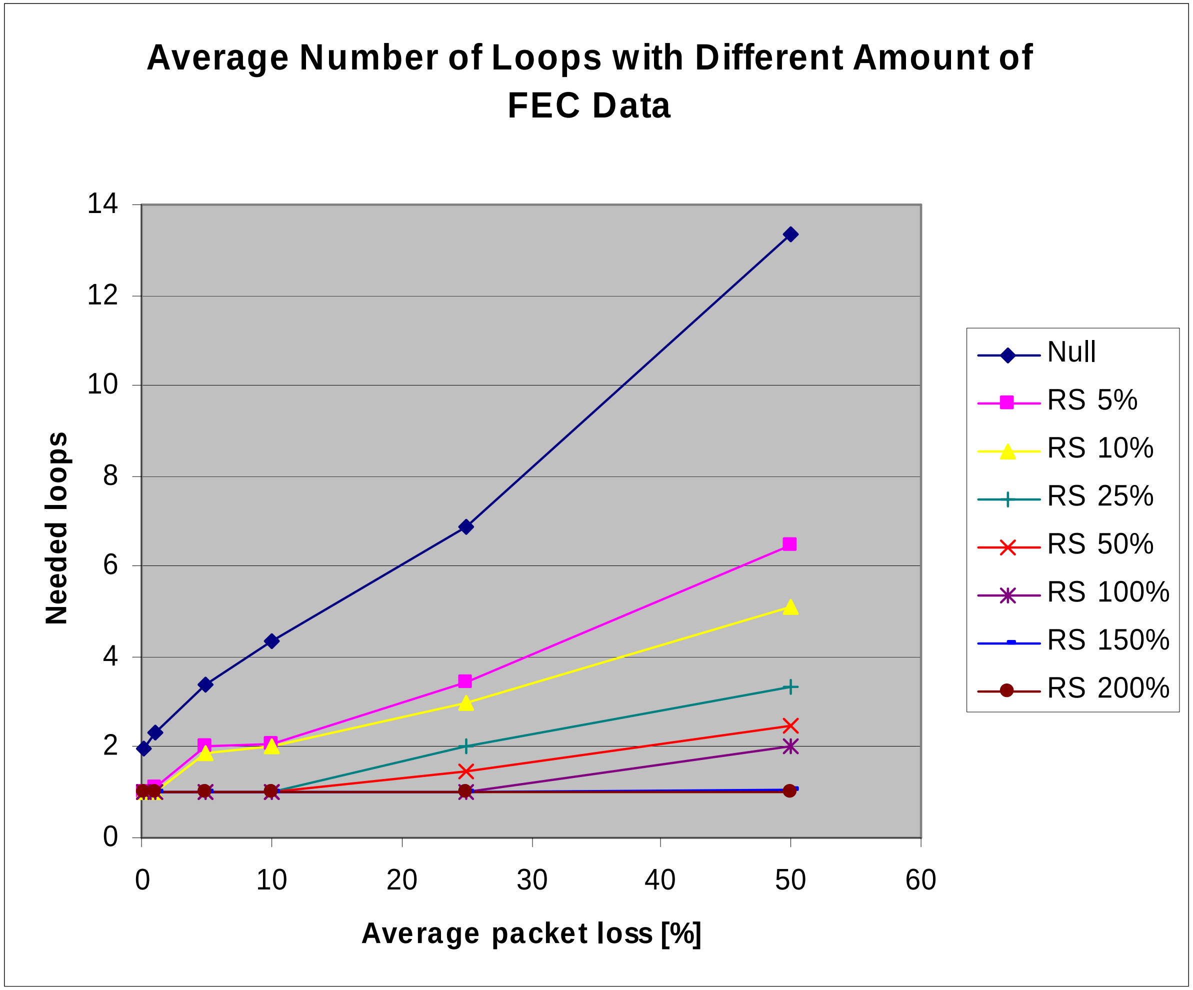
```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns:fl="http://www.example.com/flute"
              xsi:schemaLocation="http://www.example.com/flute-fdt.xsd"
              Expires="2890842807">
 <File Content-Location="www.example.com/menu/tracklist.html"
       TOI="1"
       Content-Type="text/html"/>
 <File Content-Location="www.example.com/tracks/track1.mp3"
       TOI="2"
       Content-Length="6100"
       Content-Type="audio/mp3"
       Content-MD5="Eth76GlkJU45sghK"
       Some-Private-Extension-Tag="abc123"/>
</FDT-Instance>
```

Performance Test Setup

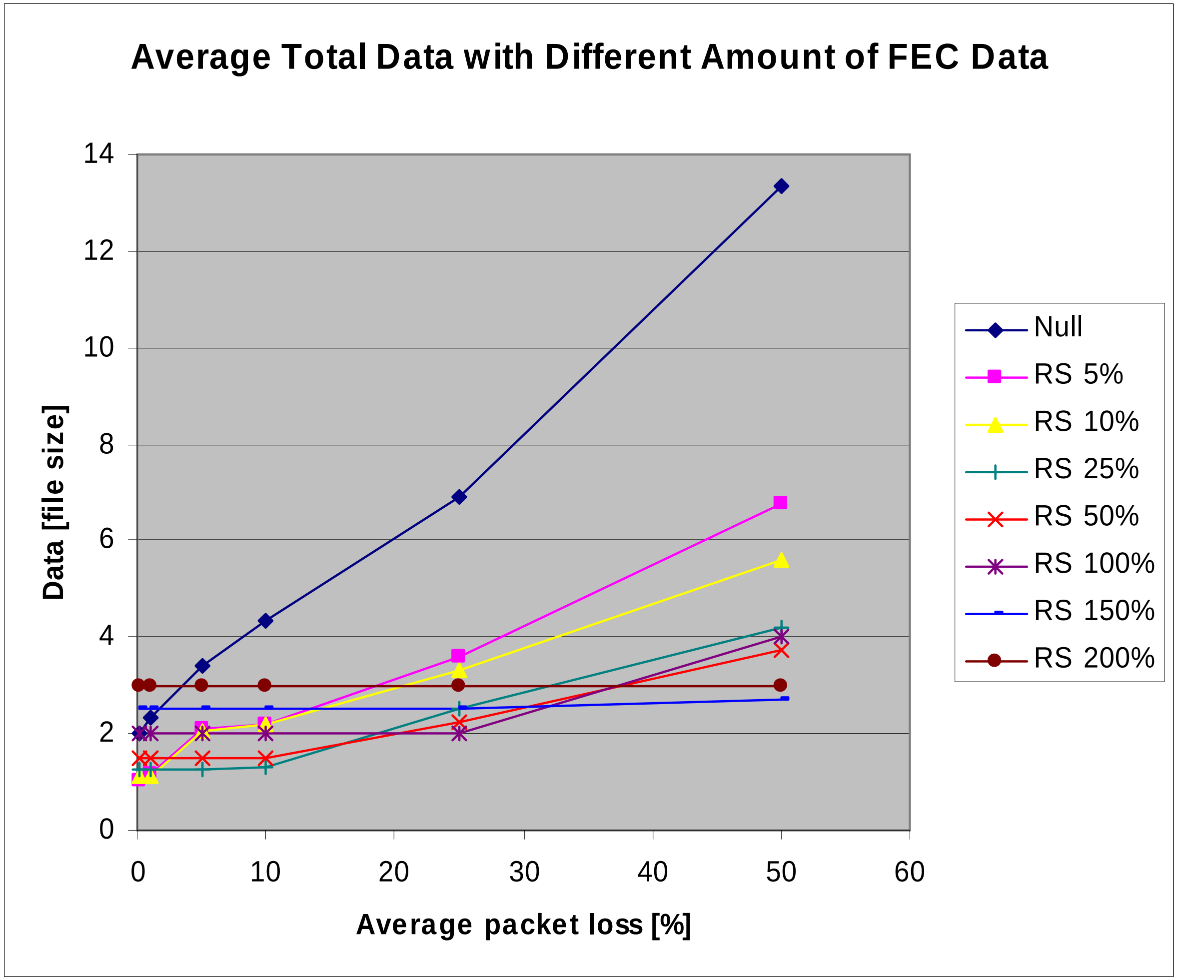
- **Single file, file size 5,2 MB**
- **Multicast UDP delivery**
- **Artificial packet loss generator**
- **Compact No-Code FEC**
- **Reed-Solomon FEC**
  - Non-interleaved transmission
- **Intention to study how many transmissions are needed to complete the reception with different amount of FEC data**
- **No feedback from the receiver to the sender**



Performance 1/2



Performance 2/2



Conclusions

- **FLUTE has good performance when some amount of parity data is added into the data carousel**
  - For example, simulations showed that it is possible to protect against 1% average packet loss by adding 10% Reed-Solomon parity data
  - Two to four loops are needed to recover missing packets in the same case without the parity data
  - With higher packet loss ratios it is even more beneficial to use parity data
  - Other FEC techniques might perform even better compared to Reed-Solomon
- **Because sender does not know anything about the receiving status of the receiver(s), the results gives some hints how to use the FLUTE sender so that the FLUTE receiver(s) gets the file(s) with optimal amount of data transmitted in a network**
- **Another option to the carousel type of packet loss recovery is to use some kind of point-to-point or point-to-multipoint file repair technique**
  - If some file repair technique is supported, the FLUTE sender could carousel the file for example the average number of loops
  - In other cases it might be best to use the worst case values to enable reliable delivery

References

[1] Paila, T., Luby, M., Lehtonen, R., Roca, V. and R. Walsh: FLUTE - File Delivery over Unidirectional Transport. RFC 3926, October 2004

[2] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L. and J. Crowcroft: Asynchronous Layered Coding (ALC) Protocol Instantiation. RFC 3450, December 2002

[3] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M. and J. Crowcroft: Layered Coding Transport (LCT) Building Block. RFC 3451, December 2002

[4] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M. and J. Crowcroft: Forward Error Correction (FEC) Building Block. RFC 3452, December 2002

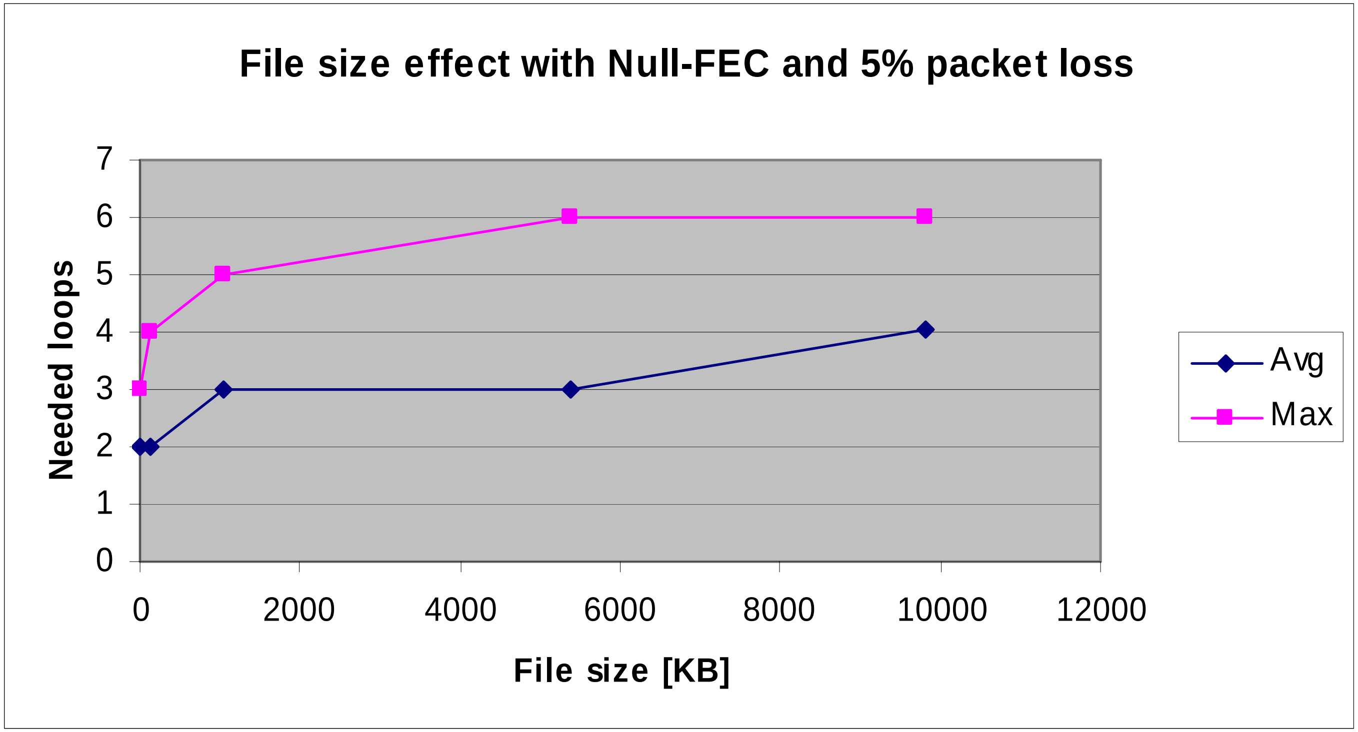
[5] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M. and J. Crowcroft: The Use of Forward Error Correction (FEC) in Reliable Multicast. RFC 3453, December 2002

[6] Luby, M. and L. Vicisano: Compact Forward Error Correction (FEC) Schemes. RFC 3695, February 2004

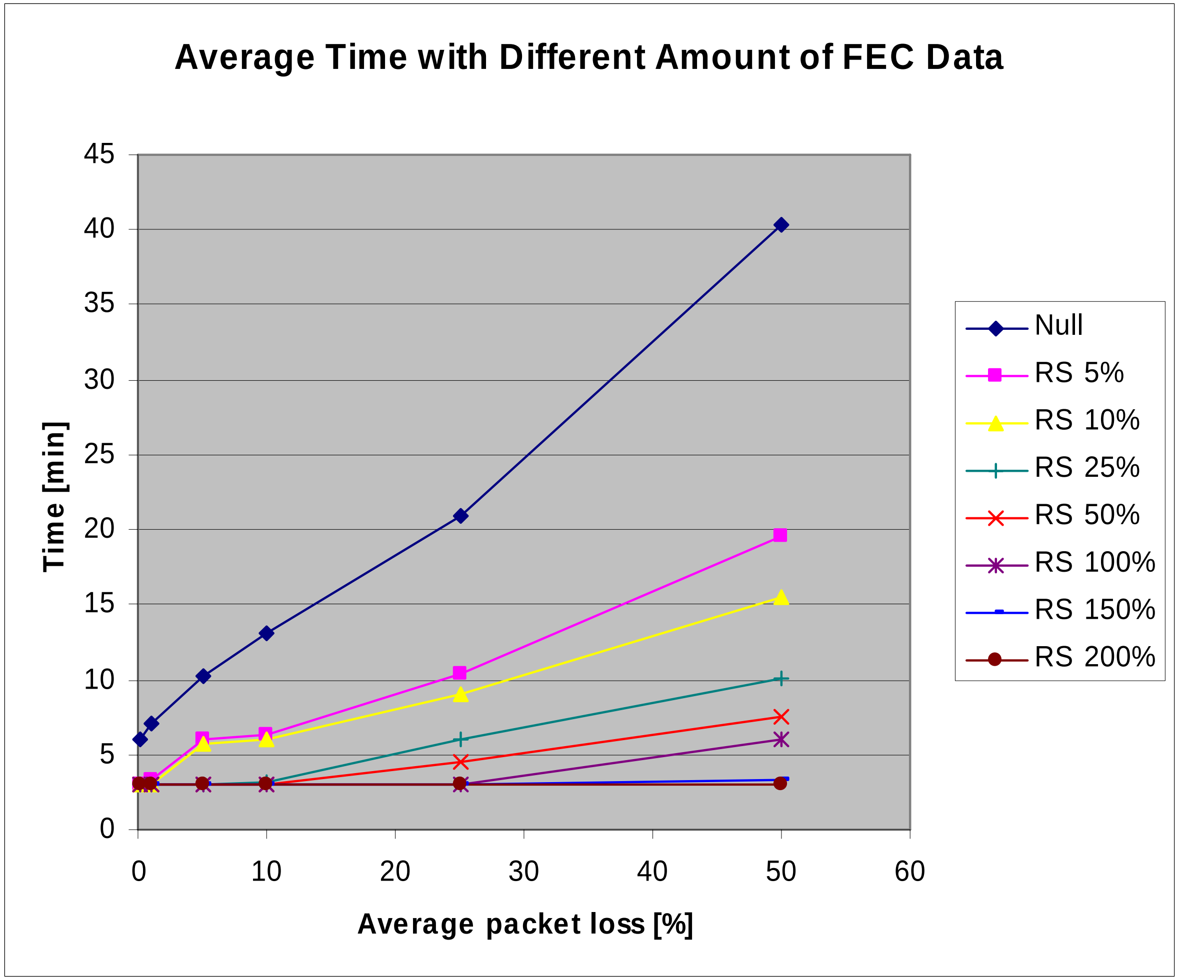
Thank you!

Questions?

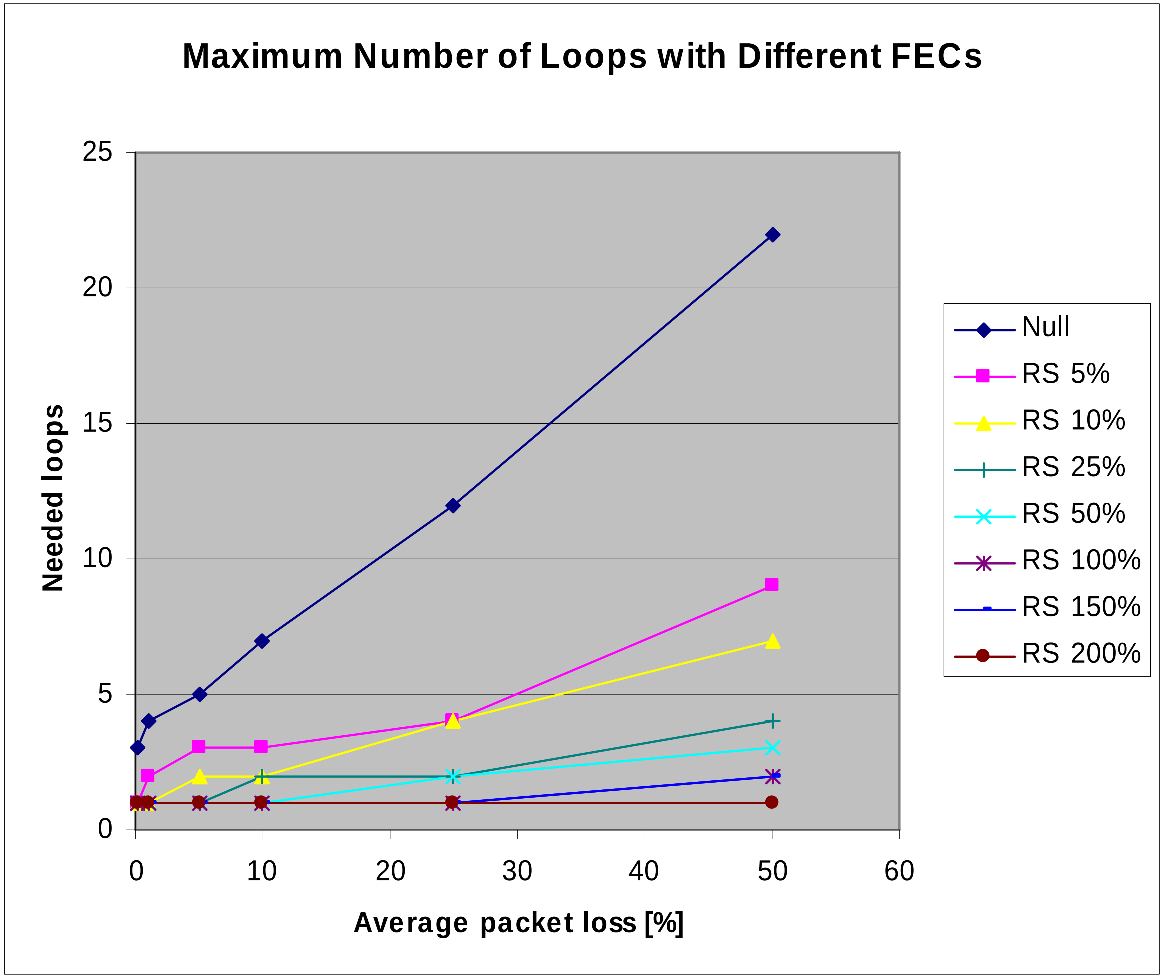
File Size Effect



Delay with 256 Kbit/s Tx



Maximum Number of Loops



Congestion Control

- **Congestion Control (CC) building block(s) needed to enable co-existence of FLUTE and TCP traffic on the Internet**

  “FLUTE is applicable for both Internet use, with a suitable congestion control building block, and provisioned/controlled systems, such as delivery over wireless broadcast radio systems.”

- **Possible CC BB from IETF RMT WG**
  - Wave and Equation Based Rate Control (WEBRC) Building Block
    - RFC3738 [7]
    - Receiver-driven
      - No feedback from receiver to sender

[7] Luby, M. and V. Goyal: Wave and Equation Based Rate Control (WEBRC) Building Block. RFC 3738, April 2004.

SDP Descriptors for FLUTE

```
v=0
o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24
s=File delivery session example
i=More information
t=2873397496 2873404696
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
a=flute-tsi:1
a=flute-ch:2
a=FEC-declaration:0 encoding-id=0
a=FEC-declaration:1 encoding-id=128; instance-id=0
a=content-desc:http://www.example.com/flute-sessions/session001
m=application 12345 FLUTE/UDP *
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
a=FEC:0
m=application 12346 FLUTE/UDP *
c=IN IP6 FF1E:03AD::7F2E:172A:1E25
a=FEC:1
```

- Mehta, H., Walsh, R., Curcio, I., Peltotalo, J. and S. Peltotalo: SDP Descriptors for FLUTE. IETF, draft-mehta-rmt-flute-sdp-03.txt (Work in Progress), June 2005

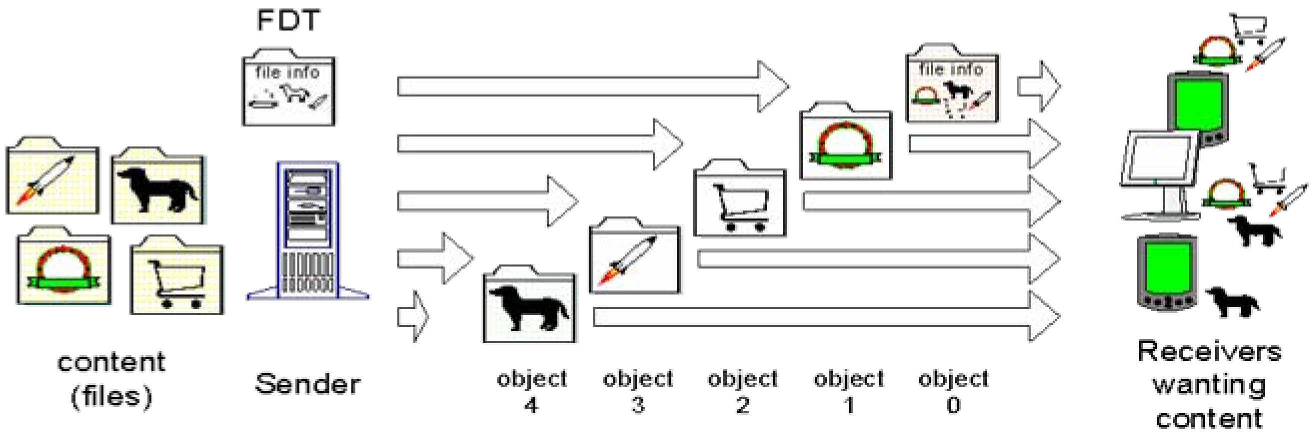
FLUTE Packet

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   V   | C | r |S| O |H|T|R|A|B|   HDR_LEN     | Codepoint (CP)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Congestion Control Information (CCI, length = 32*(C+1) bits)  |
|                          ...                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Transport Session Identifier (TSI, length = 32*S+16*H bits)  |
|                          ...                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Transport Object Identifier (TOI, length = 32*O+16*H bits)  |
|                          ...                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Sender Current Time (SCT, if T = 1)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Expected Residual Time (ERT, if R = 1)          |

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           LCT header extensions (EXT_FDT, EXT_FTI, etc.)      |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       FEC Payload ID                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Encoding Symbol(s)                      |
|                           ...                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

FLUTE Session Example

Four files described by one FDT Instance, and that FDT Instance is delivered before the files (Figure used with permission from Rod Walsh, published in "Advances in Mass Media Delivery to Mobiles", MIPS2004 Tutorial, http://mips2004.imag.fr/tutorials.php#tutorial2)



FLUTE in DVB

- **IP Datacasting over DVB-H (DVB-H / IPDC)**
  - FLUTE recommended transport protocol for announcement files in Service Discovery Channel
  - FLUTE proposed transport protocol for push file delivery

  - DVB A080, IP Datacast Baseline Specification: Specification of Interface I_MT, April 2004

FLUTE in 3GPP

- **3GPP Multimedia Broadcast Multicast Service (MBMS)**
  - FLUTE selected for MBMS download delivery method
  - Additional file repair procedure (requires return channel)
    - HTTP file repair request
    - Point-to-point repair
      - The file repair response message consists of HTTP header and file repair response payload (HTTP payload)
    - Point-to-multipoint repair
      - The file repair response message consists of HTTP header
        - » informs that point-to-multipoint repair is used instead of point-to-point repair
      - Actual file repair response payload using broadcast/multicast

- 3GPP TS 26.346 V6.1.0 (2005-06), 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (Release 6)

Other Multicast Data Transfer Protocols

- **FCAST**
  - Vincent Roca's reliable multicast file transfer application
- **NORM**
  - Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol
  - Negative-Acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Building Blocks
- **Multicast Dissemination Protocol version 2 (MDPv2)**
  - Macker J.P., R.B. Adamson, "The Multicast Dissemination Protocol (MDP) Toolkit", Proc. IEEE MILCOM 99, Nov. 1999.
- **Pragmatic General Multicast (PGM)**
  - J. Gemmell, T. Montgomery, et.al., "The PGM Reliable Multicast Protocol", IEEE Network, Vol. 17 No. 1, January/February 2003.
  - T. Speakman, et.al., "PGM Reliable Transport Protocol Specification", IETF RFC 3208, December 2001.
- **RMTP**
  - K. Lin, and S. Paul, "RMTP: A Reliable Multicast Transport Protocol," IEEE INFOCOM 1996, March 1996, pp. 1414-1424.
  - S. Paul, K.K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3, April 1997.
- **RMTP-II**
  - B. Whetten, M. Basavaiah, S. Paul, T. Montgomery, N. Rastogi, J. Conlan, and T. Yeh, "THE RMTP-II PROTOCOL", April 1998.
  - B. Whetten, and G. Taskale, "Overview of the Reliable Multicast Transport Protocol II (RMTP-II)", IEEE Networking, Special Issue on Multicast, February 2000.

Earlier Work

- **There has been done many studies concerning reliable multicast in packet erasure channels**
- **Almeroth, Ammar and Fei have examined the possibility to use best effort cyclic multicast to deliver Web pages**
  - Almeroth, K. C., Ammar, M. H. and Z. Fei: Scalable Delivery of Web Pages Using Cyclic Best-Effort Multicast. IEEE INFOCOM 1998 - The Conference on Computer Communications, no. 1, pp. 1214 - 1221, April 1998.
  - Both mathematical analysis and simulations
- **Rodriguez and Biersack used mathematical analysis to figure out the amount of parity data (per one transmission cycle) needed for succesfull delivery**
  - Rodriguez, P. and E. W. Biersack: Continuous Multicast Push of Web Documents over the Internet. IEEE Network Magazine, 12, 2:18–31, March-April 1998.
- **Nonnenmacher, Biersack and Towsley studied parity based schemes for loss recovery to achieve reliable multicast delivery**
  - Nonnenmacher, J., Biersack, E. W. and D. Towsley: Parity-Based Loss Recovery for Reliable Multicast Transmission. IEEE/ACM Transactions on Networking, 6(4):349–361, August 1998.
- **Jung, Nonnenmacher and Biersack have done mathematical analysis about reliable multicast delivery via satellite networks**
  - Jung, M., Nonnenmacher, J. and E. W. Biersack: Reliable Multicast via Satellite: Uni-directional vs. Bi-directional Communication. In Proceedings of KiVS'99, Darmstadt, Germany, March 1999.