A Survey of Reliable Multicast Protocols

Projet Planète; INRIA Rhône-Alpes
vincent.roca@inrialpes.fr
http://www.inrialpes.fr/planete/people/roca
December 13th, 2001

Outline of the presentation

- part 1- introduction
- part 2- reliable multicast and associated high level services
- part 3- selected bibliography

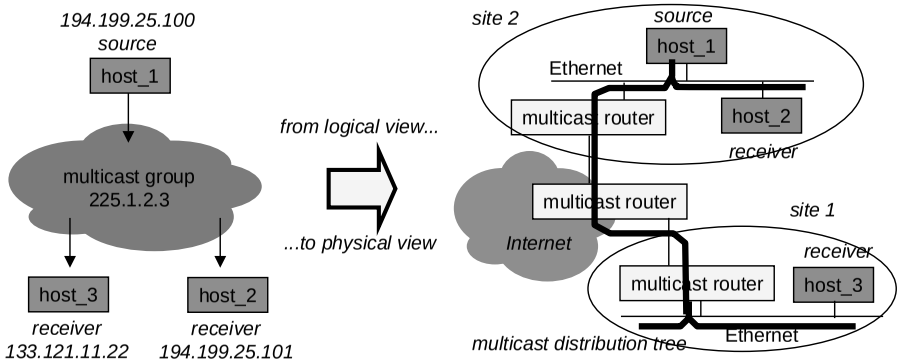
Part 1

Introduction

- Introduction: what is it and why/when should we use it?

The Internet group model

- multicast/group communications means...
  - **1 → n as well as n → m**
- a group is identified by a class D IP address (224.0.0.0 to 239.255.255.255)
  - **abstract notion that does not identify any host!**



The Internet group model... (cont')

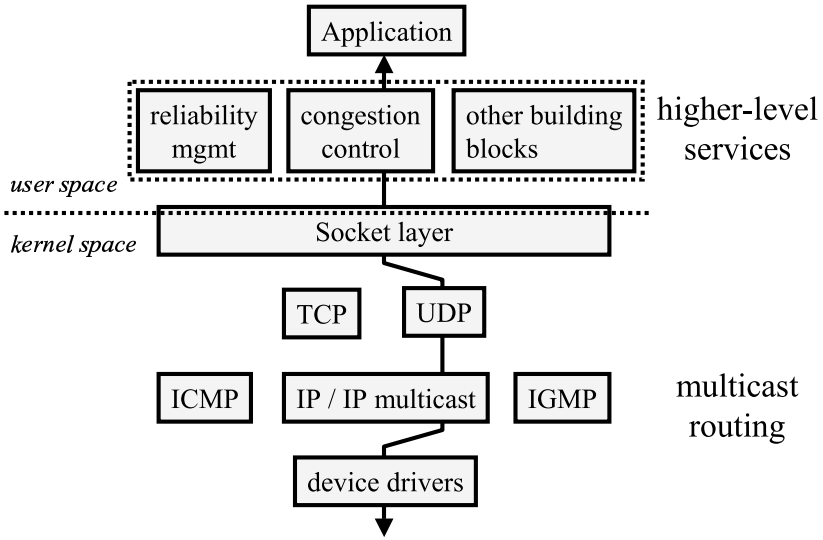
- the group model is an open model
  - anybody can belong to a multicast group
    - **no authorization is required**
  - a host can belong to many different groups
    - **no restriction**
  - a source can send to a group, no matter whether it belongs to the group or not
    - **membership not required**
  - the group is *dynamic*, a host can subscribe to or leave at any time
  - a host (source/receiver) *does not know the number/identity of members* of the group

The Internet group model... (cont')

- local-area multicast
  - **use the potential diffusion capabilities of the physical layer (e.g. Ethernet)**
  - **efficient and straightforward**
- wide-area multicast
  - **requires to go through multicast routers, use IGMP/multicast routing/...**

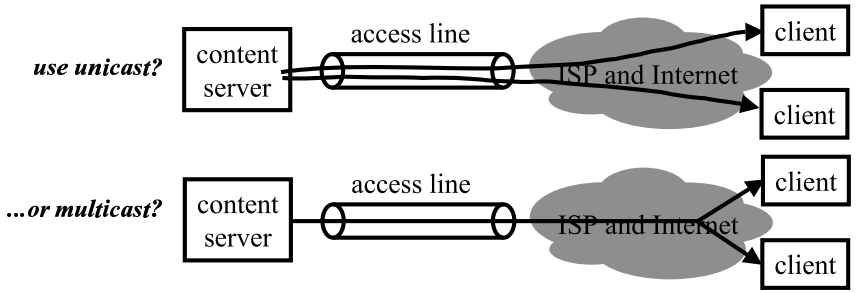
    (e.g. DVMRP, PIM-DM, PIM-SM, PIM-SSM, MSDP, MBGP, BGMP, etc.)
  - **routing in the same administrative domain is simple and efficient**
  - **inter-domain routing is complex, not fully operational**
- In this talk we won't consider mcast routing!

Multicast and the TCP/IP layered model



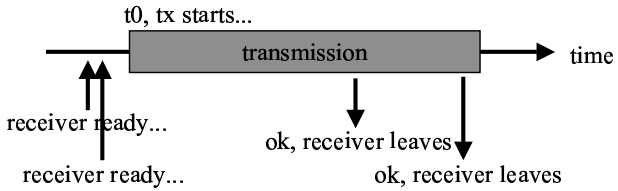
Why IP multicast?

- scalability...
  - scales to an unlimited number of users
- reduced costs...
  - cheaper equipment and access line
- increased speed...
  - increases the delivery speed



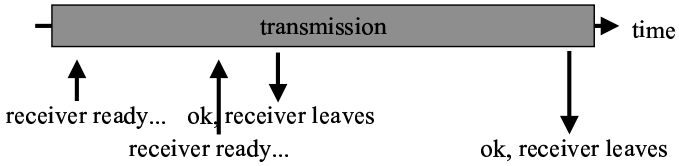
The three delivery models

- *Streaming* (e.g. for audio/video)
  - multimedia data requires efficiency due to its size
  - requires real-time, semi-reliable delivery
- *Push* delivery
  - synchronous model where delivery is started at t0
  - usually requires a fully reliable delivery, limited number of receivers



The three delivery models... (cont')

- *On-demand* delivery
  - popular content (video clip, software,update, etc.) is continuously distributed in multicast
  - users arrive at any time, download, and leave
  - possibility of millions of users, no real-time constraint



Part 2

Reliable multicast and associated high-level services

- State of the art of current research and standardization efforts

Outline of the section

- 2.1- challenges
- 2.2- use of FEC (forward error correction)
- 2.3- scalability in reliable multicast
- 2.4- IETF standardization work: the various classes of reliable protocols
  - NORM PI / TRACK PI / ALC PI
- 2.5- congestion control protocols

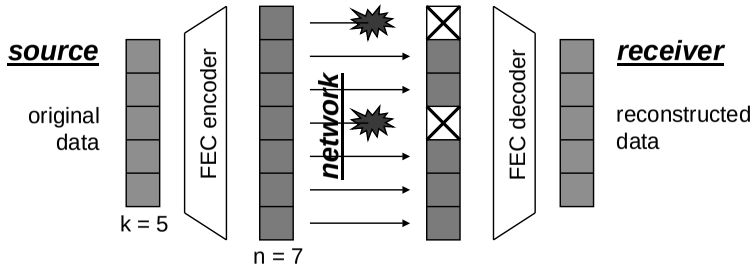
2.1- The challenges

- IETF requirements (RFC 2357)
  - *scalability* 10...000s members/sources
  - *congestion control* fair in some respect to TCP
  - *security* if possible... MSEC/SMUG working groups

- Other challenges
  - *many different application requirements*
    ⇒ "one size does not fit all"
  - *various group models*: closed (members known & fixed), semi-closed, open
    ⇒ reliability is more or less easy to provide
  - take into account the *heterogeneity of receivers*
  - be *easy to use, configure* (e.g. TRACK), *monitor*

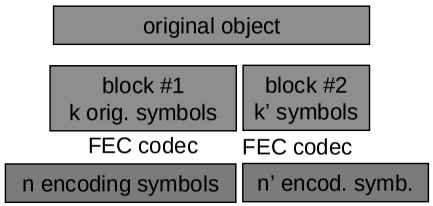
2.2- The use of FEC

- FEC (Forward Error Correction) [Rizzo97]
  - **Sender: uses FEC (k, n)**
    for k original data packets, add n-k FEC encoded redundant packets
    ⇒ total of n packets sent
  - **Receiver:**
    as soon as it receives *any* k packets out of the n, it reconstructs the original k packets



The use of FEC... (cont')

- several FEC codes exist...
- small-block FEC codes
  - e.g. Reed-Solomon codes
  - (k,n) with a k parameter limited to a few tens for computational reasons
    ⇒ **split large data objects into several blocks**
  - limited number of n-k FEC symbols created
    ⇒ **can lead to packet duplications**
  - open-source implem.
  - codec speed:
    10-80 Mbps / min(k, n-k)



The use of FEC... (cont')

- large-block FEC codes
  - e.g. Tornado codes
  - (k,n) with a **very large k**
  - but n is limited in practice for memory reasons (e.g. n = 2k)
  - high-speed encoding/decoding
  - codec speed: tens of Mbps (with k=tens of thousands, n=2k)
  - patents!!!

The use of FEC... (cont')

- Expandable FEC codes
  - no predefined limit to the n parameter
    consequence: FEC symbols can be produced on-demand, no symbol duplication
  - codec speed: 3-20 Mbps (with k=tens of thousands)
  - patents!!!

2.3- Reliable multicast scalability

- many problems arise with 10000 receivers...
  - Problem 1: *scalable control traffic*
    - **ACK each data packet (à la TCP)...**
      oops, 10000ACKs/pkt!
    - **NAK (negative ack) only if failure...**
      oops, if pkt is lost close to src,10000 NAKs!
  - Problem 2: *scalable retransmissions*
    - **if each receiver has 1% packets losses, each packet is sent several times... oops!**
  - Problem 3: *heterogeneity*
    - **send data reliably to everybody at the slowest receiver rate? High end receivers won't be happy!**

Reliable multicast scalability... (cont')

- Problem 1: scalable control traffic
  - *solution 1:* feedback suppression at the receivers
    - **each node picks a random backoff timer**
    - **send the NAK at timeout if loss not corrected**
  - *solution 2:* proactive FEC (forward error correction)
    - **send data plus additional FEC packets**
    - **any FEC packet can replace a lost data packet**
  - *solution 3:* use a tree of intelligent routers/servers
    - **use a tree for ACK aggregation and/or NAK suppression**
    - **see PGM**

Reliable multicast scalability... (cont')

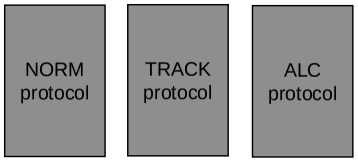
- Problem 2: scalable retransmissions
  - *solution 1:* use proactive/reactive FEC
    - **proactive ⇒ always send data + FEC**
    - **reactive ⇒ in case of retransmission, send FEC (can replace several diff. lost packets)**
  - *solution 2:* use a tree of retransmission servers
    - **a receiver can be a retransmission server if he has data requested**
- Problem 3: heterogeneity
  - *solution 1:* adjust tx rate to the slowest receiver without going below a given threshold
  - *solution 2:* use various homogeneous rx groups
  - *solution 3:* use multirate transmissions (ALC)

2.4- Current IETF standardization work

- "One size does not fit all"
  - *"requirements"* x *"conditions/problems"* matrix is too large for a single solution!!!
  - define Building Blocks (BB)
    - **logical, reusable component**
    - **used by the PI**
    - **example: Forward Error Correction (FEC)**
  - define several classes of protocols for reliable multicast: Protocol Instanciation (PI)
    - **non reusable**
    - **define protocol headers**

Current IETF standardization work... (cont')

- Flat NORM
  - for small to medium sized groups
  - simplicity, uses NAK
- Hierarchical TRACK
  - for medium sized to large groups
  - requires tree building (manual/automatic)
- Layered ALC
  - for all sizes of groups,unlimited scalability



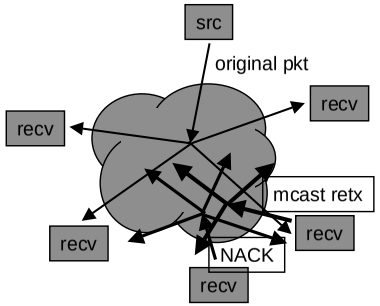
The NORM PI

- Negative Acknowledgment Oriented Reliable Multicast
- based on NAK transmissions in case of errors
- suited to small/medium size groups

- Building blocks required (or optionally used)
  - NACK (control the generation/suppression of NACK and responses)
  - FEC (for increased scalability)
  - CC (single layer, adjust tx rate to slowest rx)
  - security
  - ...

The NORM PI... (cont')

- An old example: SRM (Scalable Reliable Multicast)
  - no hierarchy
  - multicast NACK with limited scope (scalability)
  - FEC possible for improved scalability
  - automatic configuration
  - used by wb (libsrm)
  - many limitations:
    - **many-to-many multicast**
    - **RTT evaluations**
    - **moderate scalability**



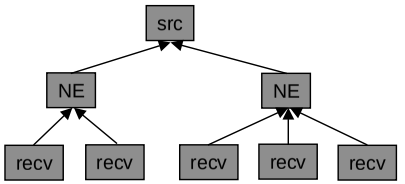
The TRACK PI

- Tree Based Acknowledgment
- a tree offers assistance services for NAK suppr., ACK aggr., retransmissions (or a subset of them)
- for medium to large groups

- Building blocks required (or optionally used) by the TRACK PI
  - like the NACK PI (NACK, FEC, CC, security)
  - plus GRA (Generic Router Assistance) for tree management

The TRACK PI

- CISCO's PGM (pragmatic multicast):
  - build a tree of NE (Network Elements) (server or router) that perform:
    - **ACK aggregation along the tree**
    - **NACK suppression along the tree**
    - **localised retransmission in a subset of the tree**
    - **retransmission (if data is cached)**
  - FEC possible for increased scalability/lower latency



The ALC PI

- Asynchronous Layered Coding
- based on multi-rate transm. + proactive FEC
- entirely ``receiver-oriented'' for maximum scalability (several millions...)
- ALC targets multicast file transfert...
- ...but a varient can easily handle hierarchical video coding for real-time streaming, etc.

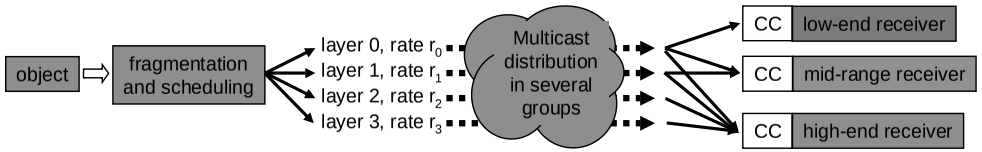
- Building blocks required by the ALC PI
  - LCT (glue between BBs + header definition)
  - FEC
  - layered CC
  - security

The ALC PI... (cont')

- Sessions
  - characterized by a set of {groups/port numbers}
- Objects
  - information carried by a session
  - example:
    - **a file <=> an object**
    - **a jpeg <=> an object**
    - **a file slice <=> an object**
  - can be one object per session
    - **e.g. transmission of a tar archive**
  - can be several objects per session
    - **e.g. transmission of a stripped archive file**

The ALC PI... (cont')

- How does it work?
  - multi-rate tx address the *receiver heterogeneity*
  - the congestion control BB (e.g. RLC) tells a receiver when to add or drop a layer

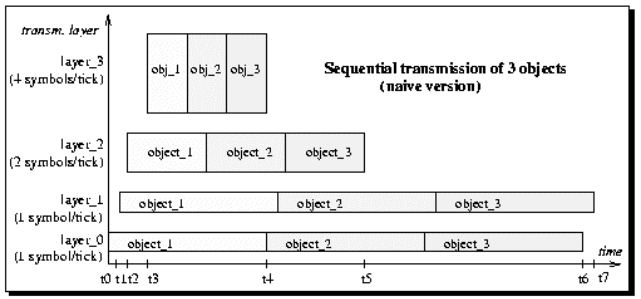


The ALC PI... (cont')

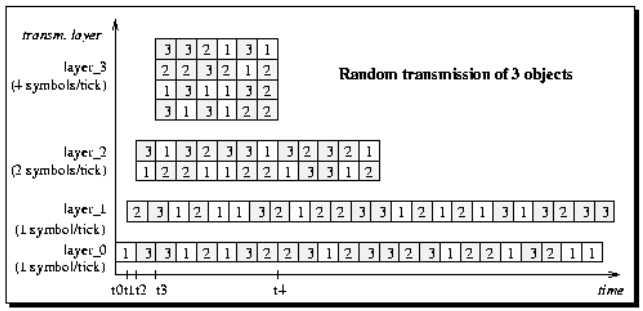
- How does it work... (cont')
  - mix in a (more or less) random manner all the data+FEC packets and send them on the various layers
  - required to counter the random losses and random layer addition/removal

The ALC PI... (cont')

sequential ordering of objects/packets



random ordering of packets



2.5- The Congestion Control BB

- general goals of CC
  - be ***fair*** with other data flows (be "TCP friendly")
    - **should a multicast transfer use as much resource as a TCP connection or n times as much ?**
    - **no single definition**
    - **be responsive to network conditions**
  - be ***stable***, i.e. avoid oscillations
  - utilize network resources ***efficiently***
    - **if only one flow, then use all the available bandwidth**

The Congestion Control BB... (cont')

- single layer versus layered transmissions
  - completely different schemes
  - single layer
    - ***sender oriented***
    - **based on ACK / NACK feedbacks**
  - layered
    - ***receiver oriented***
    - **based on losses experienced**

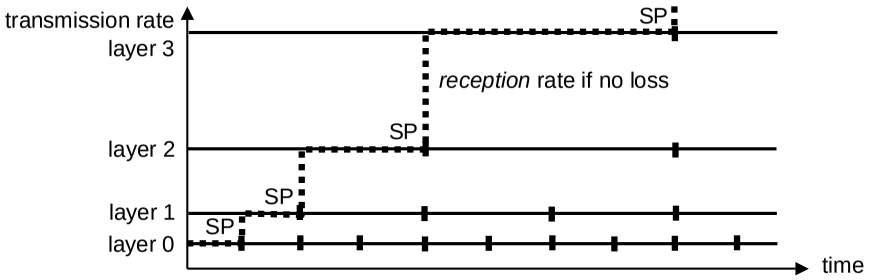
Single rate congestion control

- Example PGMCC (PGM Congestion Control)
  - used with single-rate (i.e. layer) protocols like NORM, TRACK
  - relies on a window based transmission
    - **mimics TCP**
    - **evolves according to the ACKs sent by the ``Acker''**
  - relies on an ``Acker'' selection process
    - **the ``Acker'' is the receiver with the lowest equivalent TCP throughput**

      *equivTCPthroughput = α / (RTT * sqrt(loss_rate))*
    - **the ``Acker'' changes dynamically**

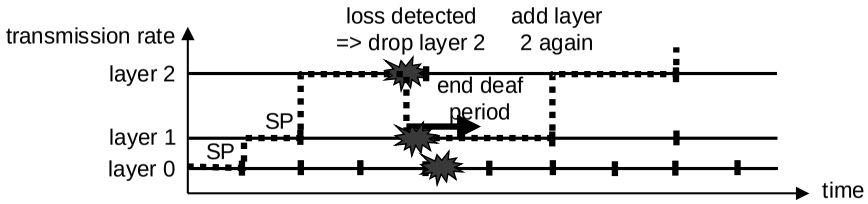
The Layered Congestion Control BB

- Example: RLC (Receiver Driven Layered Congestion Control)
  - add synchronization points (SP) / probes
    - **adding a layer is only possible at a SP if no loss has been experienced**
    - **before a SP, the source artificially increases its transmission rate to simulate the consequences of subscribing to an additional group**



The layered congestion control BB... (cont')

- RLC... (cont')
  - because of IGMP leave latency/multicast tree update latency, after dropping a layer, wait some time before measuring packet loss again

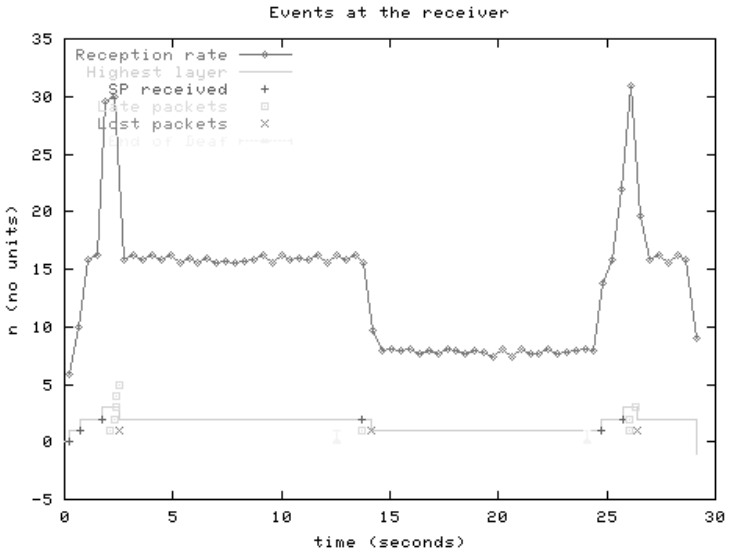


  - Limitations:
    - **limited by IGMP leave latency (a few seconds)**
    - **probing has limitations (which size?)**
    - **only adapts to packet loss, not to RTT**

      different from TCP where: rate ~1/(RTT*sqrt(p))

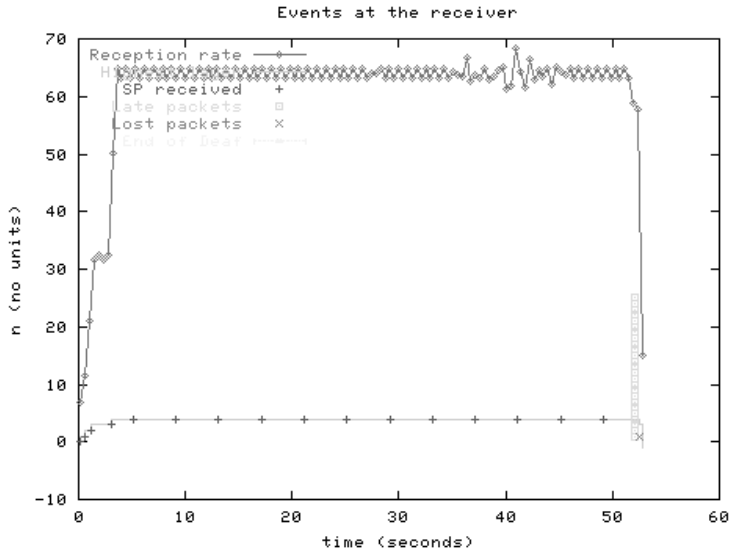
Layered congestion control : an example

- ALC session, receiver events, with losses



Layered cong. control : an example... (cont')

- ALC session, receiver events, no loss



Part 3

Short bibliography

The MCL and FCAST tools

- If you're interested in ALC/LCT... try our open source/GNU GPL implementation

http://www.inrialpes.fr/planete/people/roca/mcl/

- full featured ALC/LCT/RLC implementation
- Linux, Solaris, Win2000
- includes FCAST (multicast file transfer)
- supports « on-demand » and « push » sessions
- achieves up to 13.6 Mbps on a 100Mbps LAN
- file size limited by memory size of the source in case of high-speed tx !

Short Bibliography

- ALC, LCC, FEC documents

[ALC01] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, ``Asynchronous Layered Coding (ALC): a massively scalable content delivery transport'', RMT Working Group, draft-ietf-rmt-pi-alc-04.txt, November 2001.

[LCT01] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft, M. Handley, ``Layered Coding Transport (LCT): a massively scalable content delivery transport'', RMT Working Group, draft-ietf-rmt-bb-lct-03.txt, November 2001.

[LCC01] M. Luby, L. Vicisano, A. Haken, ``Reliable Multicast Transport Building Block: Layered Congestion Control'', RMT Working Group, draft-ietf-rmt-bb-lcc-00.txt, November 2000

FEC00] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft, ``RMT BB Forward Error Correction codes'', RMT Working Group, draft-ietf-rmt-bb-fec-05.txt, November 2001.

[FEC00] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, J. Crowcroft, ``The use of Forward Error Correction in Reliable Multicast'', RMT Working Group, draft-ietf-rmt-info-fec-00.txt, November 2000.

Short bibliography... (cont')

- NORM documents

[NORM01]B. Adamson, C. Bormann, M. Handley, J. Macker, ``NACK-oriented reliable multicast protocol (NORM)'', RMT Working Group, draft-ietf-rmt-pi-norm-02.txt, July 2001.

[NORM01b]B. Adamson, C. Bormann, M. Handley, J. Macker, ``NACK-oriented reliable multicast (NORM) protocol building blocks'', RMT Working Group, draft-ietf-rmt-bb-norm-02.txt, July 2001.

- TRACK documents

[TRACK01] B. Whetten, D.M. Chiu, M. Kadansky, G. Taskale, ``Reliable multicast transport building block for TRACK'', RMT Working Group, draft-ietf-rmt-bb-track-01.txt, March 2001.

[GRA01] K. Calvert, C. Papadopoulos, T. Speakman, D. Towsley, S. Yelamanchi, ``Generic Router Assist, functional specification'', RMT Working Group, draft-ietf-rmt-gra-fspec-00.txt, July 2001.

- Reliable multicast

[IETF RMT] Reliable Multicast Transport (RMT) charter, http://www.ietf.org/html.charters/rmt-charter.html

[Roca01] V. Roca, ``Un état de l'art sur les techniques de transmission multipoint fiables'', 4èmes Journées Réseaux (JRES01), December 2001.

[MCL] V. Roca, http://www.inrialpes.fr/planete/people/roca/mcl/