UNIVERSITE DE RENNES 1

Multicast Transport Protocols

Bernard Cousin
Irisa - university of Rennes 1
Bernard.Cousin@irisa.fr

Introduction

- Transport layer is above network layer:
  - Network layer basically handles Routing
  - Transport layer is responsible in end-to-end tasks
    - Connection management,
    - Error detection and recovery
    - Congestion control
  - These tasks should be provided efficiently
- Multicasting introduces one additional constraint:
  - Large number of members
    => Scalability
    => Reduction of transmission cost: data and control

  => Usual unicast Transport mechanisms and algorithms have to be adapted to multicasting

Application Examples

- Examples of reliable multicasting
  - Broadcast of new version of a company catalog
  - Data transmitted from the financial market
  - Video conferencing
  - Multiplayer game
  - Distributed cache data
  - Etc.
- Various requirements
  - Audio or video streaming:
    - Semi-reliable
    - Real-time
  - Data broadcasting "push mode":
    - Periodic transmission (no-real time)
  - Data broadcasting "pull mode"
    - Source oriented
    - Totally reliable
  - Etc.

Requirements for Reliable Multicast Transport

- The application have numerous receivers, but
- Does the application have
  - One or several sources?
- Does the application need
  - To know that everyone receive the data?
  - To constrain difference between receivers?
  - To be totally reliable?
  - To scale to very large numbers of receivers?
  - Ordered data?
  - To provide low delay or time bounded delivery?

Requirements for Reliable Multicast Transport

- Two multicasting models :
  - ASM : "any (number of) source multicasting"
    - Original model of IP multicasting [Deering 91]
    - Very abstract service
    - High complexity of the deployment
  - SSM : "single source multicasting"
    - Adapted to a large number of application classes
    - Easy to implement
    - Supported by IGMP v3 and PIM, for instance.

Requirements for Reliable Multicast Transport

- Did everyone receive the data
  - Confirmation at the service (application data unit) level or at the packet level ?
  - Make senses when the ADUs are significantly larger than a single packet
  - Either strong requirement for confirmation that all the receivers got an ADU
  - Or if not, to be informed of which specific receivers failed to receive the ADU
  - Aggregation of (n)acknowledgments will help to the scalability of the solution
- Delivery Guarantees
  - A mechanism for receivers to inform the sender when data has been delivered
  - Packet Transport Confirmation is an aid in application data unit confirmation

Requirements for Reliable Multicast Transport

- Total versus semi-reliable ?
  - Many applications require delivery of application data to be totally reliable
    - If any data is missing, none of the received portion of data unit is useful
    - Example: file transfer
  - Some applications do not need total reliability
    - Example : audio broadcasting where missing packets reduce the quality of the audio but do not render it unusable
    - IP native reliability could be nevertheless insufficient

Requirements for Reliable Multicast Transport Protocol

- Ordering Guarantees
  - Source ordered (or unordered) delivery guarantees
  - Total ordering across multiple senders is not recommended (more easily implemented at a higher level)

Requirements for Reliable Multicast Transport

- Constraining differences
  - Some applications constrain differences between receivers so that data reception characteristics for all (or a group of) receivers falls within some range
    - Example: stock price feed where a receiver does not accept to suffer more delay than any other
  - Difficult to satisfy without harming performance
    - The worst receiver leads
    - Counter example : XTP offers a reliable multicast transport service which selects always the lowest bidder

Requirements for Reliable Multicast Transport

- Timed-bounded delivery
  - Many applications require data to be delivered as fast as possible
    - No absolute deadline
  - Some applications have hard time delivery constraints
    - If data does not arrive at the receiver by a certain time, there is no point in delivering it at all
    - Example : audio or video streaming with real–time constraints or where new data supersedes old one
    - Usually implies a semi-reliable protocol
- Real-Time Control
  - May provide some means for soft real-time feedback to be measured and returned to the sender

Performance Requirements for Reliable Multicast Transport Protocol

- Good performance mechanisms
- Congestion control and good throughput
  - Packet loss :
    - First symptom of congestion
    - Primary obstacle to good throughput
  - Measuring and reacting to packet loss is crucial
  - Main solutions are
    - Data packet acknowledgment
    - Negative ack. of missing packets
    - Redundancy allowing not all packets to be received

General Requirements for Reliable Multicast Transport Protocol

- Safe to deploy in the widespread Internet
- Adaptability/Scalability
  - Should able to work under a variety of conditions
    - Network topology
    - Link speed
    - Receiver capability
  - Any receiver set size : 1000 - 10^6
- Security
  - Data confidentiality, sender authentication, defenses against DoS, etc.

Others requirements

- Group membership
  - Anonymous: the sender does not know the list of receivers
  - Fully distributed: the sender receives a count of the number of receivers and, optionally a list of a failures
- Group membership control
- Special Networks
  - Support for satellite networks is not required (including those with terrestrial return paths or no return paths at all)

Outline

- Introduction to Reliable Transport Multicast
- Requirements
- Main functionalities
  - Reliability
  - Congestion Control
- Internet Reliable Transport Multicast
- RTP
- XTP

Reliability Mechanisms

- ACK
- NACK
- Replication
- FEC
- Layered Coding

ACK-based Mechanisms

- Every receiver send an ACK packet for every data packet
  - Implosion of ACKs
- Blocking multiple ACKs into a single packet [RMWT98]
  - Allowing larger receiver groups
  - But feedback becomes too infrequent for sender-based congestion control

Tree-based ACK Mechanisms

- Arranging the receivers into a tree [MWB+98, KCW98]
  - Receivers generate ACKs to a parent node
  - Which aggregate those ACKs to its parent in turn, etc.
  - Data packets are multicast as normal
  - Failures affect a subset of receivers
  - With good ACK-tree formation, tree-based ACK mechanisms are potentially the most scalable RM solutions

Tree-based ACK Mechanisms (2)

- Tree formation and maintenance is the first issue
  - Automatic tree formation based on local information
- Subtree retransmission is the second issue
  - Intermediate tree nodes can retransmit missing data to the nodes below them (without relying on the original sender)
    - Reduced load on sender and higher nodes, fast detection and fast retransmit
    - Rely on a good correlation at the point of retransmission between the ACK tree and the actual multicast data tree
    - Use of administrative scoped multicast groups might provide a solution

Tree-based ACK Mechanisms (3)

- Nature of aggregation
  - Performed at the interior nodes on the ACK-tree
    1. Aggregate ACKs by sending a single ACK when all children have ACKed
    2. Aggregate ACKs by listing all the children that have ACKed
    3. Send an aggregated ACK with a NACK-like exception list

  1 is simple and efficient, but 2 or 3 are required when the sender needs to know exactly which receivers received the data

NACK-based mechanism

- Send a NACK for every data packet, they have discover, they did not receive
  - No needs to know how many receivers there are
  - Receivers are responsible for reliability :
    - simple fault-tolerance
  - Sender dos not need to keep track of the receivers state
    - Sender state reduced
  - A single NACK is needed to indicate a missing packet by any number of receivers, i.e. cumulative

NACK suppression

- The NACK must
  - Reach the sender (or any node that can resend the packet)
  - As soon as possible
    - ACK could be delayed, NACK should not
  - For only one copy of the missing data to be received by the nodes needing retransmission

Protocol Examples: SRM

- Scalable Reliable Multicast
  - "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing". Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, Lixia Zhang, IEEE\ACM Transactions on Networking (1995)
  - Uses random timers weighted by the round trip time
  - Between the sender and each node missing the data

Protocol Examples: NTE

- Network Text Editor (NTE)
  - "A scalable shared text editor for the MBone". Mark Handley and Jon Crowcroft, SIGCOMM (1997)
  - Sender-triggered mechanism based on random keys and sliding masks
  - No timers
  - Difficult to provide the constant low-level stream of feedback needed to perform congestion control

Protocol Examples: AAP

- *AAP*
  - *" Multicast Address Allocation Protocol". M. Handley, Internet Draft (1999)*
  - Exponentially distributed random timers
  - Without needing to compute the RTT to each receiver

Protocol Examples: PGM or LMS

- *PGM - LMS*
  - *"PGM reliable transport protocol specification". Farinacci, Speakman*. Internet draft (1998).
  - "An Error Control Scheme for Large-Scale Multicast Applications" Christos Papadopoulos, Guru Parulkar, George Varghese, Symposium on Principles of Distributed Computing (1998)
  - Routers suppress duplicate NACKs
  - In PGM router assistance supplements random timers and localize suppression

Timers

- Random timers
  - Reduce feedback delay
  - But are difficult to use when
    - All the RTTs are not known
    - Or the numbers of receivers is unknown
- Exponentially weighted random timers
  - Work well across a large range of session sizes
  - Good worst case delay
- Router assistance
  - Either form of timer mechanism can be supplemented by routers
  - Sender-triggered NACK mechanisms is not well appropriated

Replication

- Some applications do not need explicit reliability mechanisms.
  - For instance
    - A multicast game where the position of a moving object is multicast
    - Because a new position supersedes the old one before any retransmission could take place
  - In traditional ACK or NACK based protocol, the probability of any packet being received by all the receivers in a large group can be very low
    - leads to high retransmission rates
- Replication does not suffer from the size of the group and has minimal delay

Forward Error Correction

- FEC
  - Technique for protecting data against corruption
  - Based on redundancy
- Erasure codes
  - Allows generation of *n* encoding packets from *k* original data packets
  - The initial packet can be reproduced, if at least *k* of *n* encoding packets are received
- Dependency on which packets have been lost is removed
  - The amount of traffic required to repair spatially uncorrelated packet loss is lower than with retransmission mechanisms

Proactive vs. reactive FEC

- Proactive FEC
  - Sender decides a priori what encoding level is used for each round of data packets
- Reactive FEC
  - The sender initially transmits only the original data packets
  - Feedbacks from the receivers inform the sender of the packet lost rate
  - The appropriated additional encoding packets are retransmitted
  - Receivers report via ACKs or NACKs
  - Only the receiver missing the most packets need sends a NACK
  - Used to weight the random timers
- Proactive and reactive can be combined efficiency
- FEC adds end-to-end latency
  - No problem for bulk-data applications but replication may be better for interactive applications

Layered coding

- Data is spread across several multicast groups, each one associated to one encoding layer
  - A receiver must join one or more of the multicast groups
- Generally the encoding is hierarchically organized
  - To be able to decode the data of the layer *N* the receiver should receive the data packets of the *N* first multicast groups
- Different receivers are allowed to receive the traffic at a different rates, according to the available capacity
- Scalable solution because it requires no feedback
  - However coordination from sender of receivers behind the same congested links should be required

Outline

- Introduction to Reliable Transport Multicast
- Requirements
- Main functionalities
  - Reliability
  - Congestion Control
- Internet Reliable Transport Multicast
- RTP
- XTP

Congestion Control Mechanism

- Delivery model of basic Internet
  - Best effort, no guarantee
  - End-systems are expected to be adaptive:
    - Reduction of their transmission rate at a level appropriate for the congestion state of the network
- Five classes of single-sender multicast congestion control
  - Sender, receiver or router based

Sender-controlled, one group

- A single multicast group
- Feedback from the receivers is used to control the rate
- Transmit at a rate dictated by the slowest receiver
  - Cf. XTP

Sender-controlled, multiple groups

- The initial multicast group is adaptively subdivided into multiple subgroups
  - Subgroups are centered on congestion points in the network
- Application-level relays
  - Buffer data from a group nearer the original sender
  - Retransmit data at a slower rate into a group further from the original sender
    - Different receivers can receive at different rates
  - Sender based congestion control between members of a subgroup and their relay

Receiver-controlled, one group

- A single multicast group
- If the receiver transmit too rapidly for the congestion state of the network, the receiver leaves the group

Receiver-controlled, layered

- Data is striped across multiple multicast groups simultaneously
  - Cf. ALC
- Receivers join and leave these layered groups depending of their measurement of the congestion state of the network
- Receivers should left and join in a coordinated fashion behind a bottleneck link
  - Cf. coordination done by RTP/RTCP

Router-based congestion control

- Functions added to multicast routers:
  - Conditional joins
    - Join is rejected if the specified loss rate is above the acceptable level
  - Traffic filtering
    - Exceeded traffic is discarded
  - Fair queuing scheme with end-to-end adaptation
    - Additional states are generally not acceptable to backbone routers

Reliability versus Congestion Control

- Reliability and Congestion Control should be considered simultaneously:
  - The same mechanism providing reliability will sometimes be used to provided congestion control
  - Receiver-based congestion and FEC are likely for achieving good throughput for bulk-data transfer:
    - no feedback in both solution

Outline

- Introduction to Reliable Transport Multicast
- Requirements
- Main functionalities
  - Reliability
  - Congestion Control
- Internet Reliable Transport Multicast
- RTP
- XTP

Internet Reliable Multicast Transport

- RMT Working group from IETF
  - See RFC 3450 to 3453 (experimental), RFC 3048, RFC 2357 (requirements)
- Three protocols
  - NORM, TRACK, ALC
- PGM from Cisco (Track implementation)
- FEC
- Congestion control :
  - PGMCC
  - RLC/FLID-SL/FLID-DL

Internet Reliable Transport Protocol

- To cope with heterogeneous application requirements and, with the specificities of multicast Transport control mechanisms (i.e. to achieve some efficiency)
- The Building Block approach
  - BB: Building Blocks
  - PI: Protocol Instantiation
- Three protocol classes:
  - NACK Oriented Multicast (NORM)
  - Tree based Acknowledgment (TRACK)
  - Asynchronous Layered Coding (ALC)

NORM Protocol Instantiation

- A negative acknowledgement is sent when a loss is detected
  - Adapted to small or medium size groups, with homogeneous receivers
  - If flow control is assured by PGMCC then the slowest receiver leads
- Main blocks:
  - Emission block
  - NACK management block (receiver side):
    - NACK suppression mechanism
  - NACK management block (sender side)
  - RTT estimation block
    - used by the NACK suppression mechanism, and the flow congestion control
  - Flow control block (for instance PGMCC)
  - Group size estimation block
  - FEC block (essential to achieve scalability)

An example of NORM instantiation: SRM

- Scalable Reliable Multicast [S. Floyd 95]
  - Used by the classical "wb" application
  - Several senders may exist
  - Nearby members are used to retransmit missing packets
  - Packet ordering is not guaranteed
  - Packets are identified (<sender IP @, packet number>)
  - Retransmission requests are randomly delayed:
    - One first receiver broadcasts a retransmission request
    - Others receivers detects the retransmission

TRACK

- Tree based Acknowledgment [Whetten 2003]
  - Assumption : automatic tree configuration
  - Main design considerations:
    - Application-level confirming delivery
    - Aggregation of control traffic and sender statistics
    - Local recovery
    - Enhanced flow and congestion control

Major Elements

- Session
  - *End of Stream* condition
  - Session id
  - Session tree
  - Source, members and intermediary nodes
- Repair Head:
  - A node within the tree which receives and retransmits data
  - Aggregates and forwards control information toward the sender
  - The sender is the root Repair Head

TRACK algorithms

- Timing Algorithm
  - to control the speed at which TRACK messages are sent
- Statistics Request
  - A sender may prompt receivers to generate and report a set of statistics back to the sender
- TRACK Aggregation
  - Interior tree nodes provide aggregation of control traffic flowing up the tree. The aggregated feedback information includes that used for end-to-end confirmed delivery, flow control, congestion control, and group membership monitoring and management

TRACK Pros & Cons

- The tree enhances the scalability
  - NACK suppression
  - ACKs aggregation
  - Local retransmission

But

  - Increasing of the complexity : tree management
  - Repair nodes have to
    - be identified,
    - maintain one state for every group,
    - memorize packets

TRACK blocks

- Same blocks than NORM

+ Generic Router Assist
  - Repair Head functionalities

PGM

- Pragmatic Generic Multicast
  - Proposed by Cisco
  - Similar to TRACK/GRA
  - Based on "Network Elements"
    - Either multicast routers or servers
- NE functions:
  - NACK suppression
  - Router broadcasts missing packets on the appropriated subtree
  - Server sends missing packets
- NEs can be incrementally added as the group size increases. For small size groups, one solution is to use no NE

ALC

- Asynchronous Layered Coding
  - Receiver-oriented protocol
  - No feedback
    => Maximal scalability (million of receivers)
- Based on the layer notion
  - With the packets which belongs to the *k* first layers a receiver can produce a quality level *k* data flow.
  - Each receiver joins to one or several layers
    - The receiving data rate (and the quality) is selected by the receivers
    - Heterogeneity of the receiver (of the path toward) is taken into account

Functioning principles

- Layered congestion control (LCC) determines the receiver to join or leave a layer
  - When the error rate is low the receiver join the next upper layer
  - When the error rate is high the receiver leave the higher layer
- Determination of the number of layers and the rate associated to each layer
  - Is application dependant
  - Not too many layers: delay can be very high to reach the highest layers

Application field

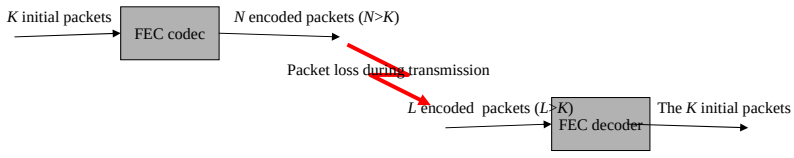
- Streaming application
  - The best solution for
    - Push mode with repetitive continuous transmission (à la videotext)
    - Pull mode when the size of the group and its heterogeneity are high
  - For instance : MPEG video transmission
    - I frame = layer 1
    - P frame = layer 2
    - B frame = layer 3

ALC Blocks

- ALC is built over the following blocks:
  - Layered Coding Transport: the core block. Definition of the general header format and link with the next blocks
  - LCC block: congestion control
  - FEC block: mandatory because layered transmission is very sensitive (prone) to error. Lower layers must be protected
  - Security block: source authentication and data integrity

FEC

- Forward Error Correction
  - Usual FEC protects against bit error
  - Here, it should protect against packet loss



FEC Codes

- Restricted Bloc Codes:
  - For instance: Reed Solomon code
  - *K* <= *N* <= 256
    - Generally, *K*=32, packet size is 1024 bytes (file<32KB !)
  - Most frequently used FEC codes, most dense codes
- Large Bloc Codes:
  - For instance: Tornado code
  - K is larger (N<2048), coding and decoding times are shorter but L > K.(1+a) with a=[5%-10%]
- Extensible code
  - Appropriate to very large sequence of packets (N>>)

Congestion Control Protocol

- Multicast communications are potentially dangerous
  - Base on UDP: no congestion control !
  - Some usual solutions:
    - Fixed and very low throughput (some few kbps)
    - Adaptative approach using RTCP
  - Fairness amongst the data flows and efficiency:
    - TCP-friendly method
    - No waste of resource
    - Stability
- Two approaches
  - PGMCC (NORM or TRACK compatible
  - LCC (ALC compatible)

PGMCC

- PGM Congestion Control [Rizzo, SIGCOM'00]
  - A window (TCP congestion window like)
    - The window size limits the data rate
  - The ACKer sends ACKs on which the window size is modulated
  - The receiver having the lower rate is selected as the ACKer
  - Equivalent TCP rate is modeled by:
    - `Rate = constant/(RTT * sqrt(loss-rate))`
    - `RTT` and `loss-rate` are transmitted by the receivers into control messages
  - Not a reliability function

LCC

- Layered Congestion Control
- General principle
  - If there is no loss during a time period, the receiver may join the next upper layer, when some appropriate signal is sent by the sender
  - If there is some lost packets, the receiver leaves immediately the higher layer. The receiver enters a frozen state, which gives time to prune the overloaded tree branch.
- Three main LCC protocols exists:
  - RLC, FLID-SL, FLID-DL

Layered Congestion Control Protocols

- Receiver-driven Layered CC
  - Group Join or Leave are synchronized by Synchronization Point placed into some packets
  - The simplest CC protocol
- Fair Layered Increase/Decrease-Static Layer
  - Similar to RLC
- Fair Layered Increase/Decrease-Dynamic Layer
  - The layer rate periodically is decreased
    - Receivers should add a new layer periodically, to maintain their data rate
    - When a congestion occurs no notification is required
    - Complexity of this protocol is high

Internet Reliable Transport Protocol

Summary

- No one solution fits all
  - Most solutions provided single sender multicast service
- NORM
  - Full reliability, some reasonable number of receivers
- TRACK
  - Fully reliable, higher number of receivers
- ALC
  - Huge receiver number, but incomplete reliability (no retransmission),
  - No retransmission delay but coding and decoding delays

=> The best protocol is to be chosen by the application

Outline

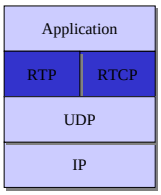
- Introduction to Reliable Transport Multicast
- Requirements
- Main functionalities
  - Reliability
  - Congestion Control
- Internet Reliable Transport Multicast
- RTP
- XTP

Real Time Protocol

- Real Time Transport Protocol
  - Designed to support multiparty multimedia conferences
  - Used by many multimedia applications
  - Real-time applications:
    - Transport of audio and video streams
  - RTP is a framework
    - Application can add specific functions
    - Application Level Framing concept
    - No functions for error control, retransmission, flow or congestion control.
    - No quality-of-service guarantee, no resource reservation

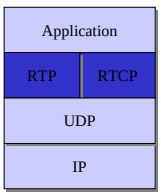
RTP Architecture

- RTP consists of two parts :
  - Real-time Transport Protocol
    - Transmission of real-time data
  - Real-time Control Transport Protocol
    - Feedback about transmission quality and information about members of a session
    - Could be used over any



RTP Transport Level Protocol

- RTP could use any Transport protocol :
  - UDP (for multicasting), but also TCP or ST-II.
  - Port number specifies the Internet service
    - RTP port = n (even)
    - RTCP port = n+1 (odd)

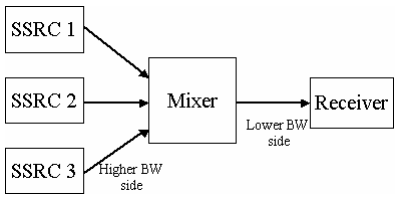


Connection/Membership management

- No tight control of group membership
  - Implicit member join
    - Users implicitly join a group by sending RTCP data units to the group
    - Others members are made aware when they receive these data units
  - Implicit member leave
    - Departure of a member from a group is recognized when RTCP data units stop arriving from this member
    - RTP group is (loosely) monitoring through timer
- RTP data transfer is unreliable
  - Because of the lack of group management functions
  - Furthermore RTP may be used without RTCP !

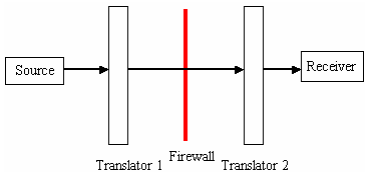
RTP Mixer

- For instance:
  - The mixer resynchronizes incoming audio packets to reconstruct the constant 20 ms spacing generated by the sender, mixes these reconstructed audio streams into a single stream, translates the audio encoding to a lower-bandwidth one and forwards the lower-bandwidth packet stream across the low-speed link
  - The mixer puts its own identification as the source (SSRC) of the packet and puts the contributing sources in CSRC fields



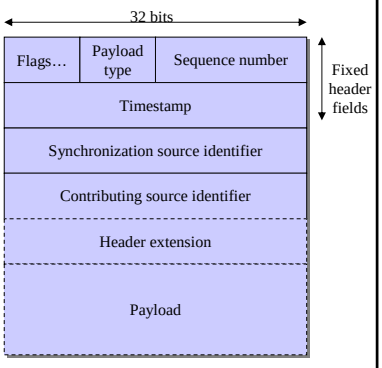
RTP Translator

- A problem occurs:
  - one or more participants of a conference are behind a firewall which won't allow an IP packet containing the RTP message to pass.
- Two translators are installed, one on either side of the firewall,
  - the outside one tunneling all multicast packets received through a secure connection to the translator inside the firewall. The translator inside the firewall sends them again as multicast packets to a multicast group restricted to the site's internal network
- Translator do not change SSRC or CSRC fields



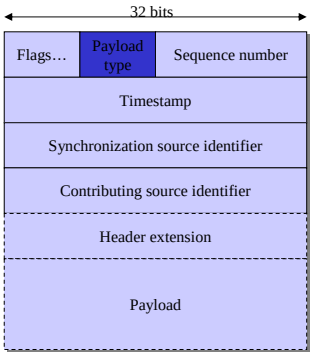
RTP packet format

- RTP packet
  - Fixed header part
  - Header extensions (optional)
  - Payload



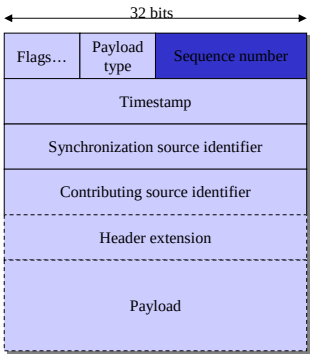
Profile

- The significance of every fields is not defined by RTP, but by profiles
  - Payload format and RTP header extensions are application dependant
- Profile
  - RTP profile is determined by Payload Type field
    - Payload format
    - Required header extensions and their format
  - For instance
    - MPEG profile
    - H.261 profile



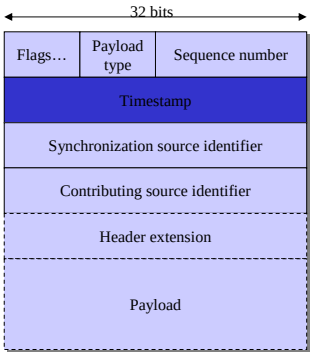
Sequence number

- A (unique) sequence number is assigned by the sender to each RTP packet
  - Loss detection and reordering
  - Not interpreted by RTP



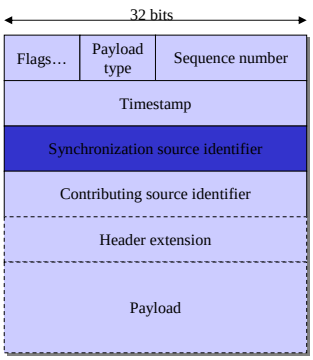
Timestamp

- The timestamp is incremented for each sample.
  - Application can use the timestamp to synchronize the samples of a stream or between different streams
    - Cf. Mixers



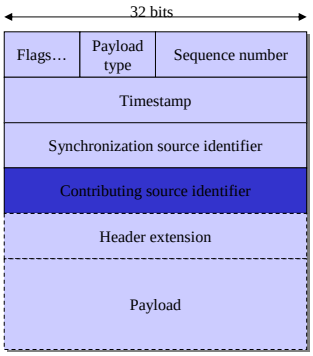
SSRC

- Synchronization source identifier
  - Identifies the source of a data stream
  - Must be unique
    - Selected randomly
    - In case of collision, a participant must chose another SSRC and send a RTCP BYE message
  - Sequence numbers and timestamps apply to each stream with the same SSRC
- For instance a sender must use two different SSRCs when sending an audio and a video stream



CSRC

- Contributing source identifier
  - Identifies the orign of a data stream
- For instance a mixer must use two different CSRCs after having merged into one single stream, one music stream and one voice stream
- Many others extensions
  - Begin of synchronization unit
  - Reverse-path option
  - Security option
  - Application specific option
  - Etc.



RTCP

- Use to exchange information between users
  - Feedback information about receiving quality
    - For instance: RR
  - Information about sent data
    - For instance: SR
  - Information about session participants
    - For instance: Source Descriptor (SDES)
      - Which maps source identifier to one or several more general identifiers : EMAIL, CNAME, TXT, etc.
- Transmission interval of RTCP packet depends on
  - Group size
  - Available bandwidth

SR: Sender Report

- Each source periodically issues a sender report
  - Timestamp to estimate the RTT when associated with RR
  - Report how many RTP packets and bytes has been sent so far

RR: Resource Report

- Resource Report
  - The feedback information is sent periodically by each receiver using RR data units
    - Loss rate
    - Number of lost RTP packets
    - Highest sequence number received
    - Jitter
    - NTP timestamp for the last sender report received
    - Time between receipt of the last sender report and transmission of the receiver report
  - Used by adaptive applications

Other RTCP messages

- SDES: Source description items, including CNAME
- BYE: Indicates end of participation
- APP: Application specific functions

Secure RTP

- SRTP:
  - the secure profile of RTP,
  - recommended for applications that need privacy or authentication.
- SRTP is not a separate protocol but a profile of RTP.
  - SRTP's SAVP profile encapsulates RTP packets, encrypts the RTP payload, optionally adds a message authentication tag (strongly recommended) and optionally adds an MKI (Master Key Id. identifies a key within an SRTP cryptographic context).
  - SRTP's SAVP profile accepts all of the RTP AVP profile's payload types.
  - As with any RTP system, there can be an SRTP intermediate system that intercepts RTP packets and converts them to SRTP packets, or vice-versa.

Références

- Perkins, Colin. *RTP: Audio and Video for the Internet* (1st ed.) Addison-Wesley. 2003.
- H. Schulzrinne , S. Casner, R. Frederick, V. Jacobson. *RTP : A Transport Protocol for Real-Time Applications*. RFC 3550 (Standard 64). 2003.

Outline

- Introduction to Reliable Transport Multicast
- Requirements
- Main functionalities
  - Reliability
  - Congestion Control
- Internet Reliable Transport Multicast
- RTP
- XTP

XTP

- Express Transport Protocol
  - Strayer, Dempsy, Weaver, "XTP : The Xpress Transfer Protocol", Addison-Wesley, 1992.
  - High performance transport protocol
    - Hardware based implementation
    - Tentative integration of Network and Transport layers
      - Version 4.0 is IP compatible, but prior versions are not
    - 8 or 4 bytes alignment

XTP innovations

- Main innovations :
  - First systematic attempt to trim a protocol for performance
  - Offers a variety of advanced protocol mechanisms
    - Many protocol mechanisms can be selected separately
    - Strict separation between control data and user date
      - No piggybacking: user data units do not contain acknowledgment
    - Easy adaptation to meet the requirement of different applications
    - Multicast functionality has been added to support group communication
      - Reliable multicast service
      - Sender oriented-basis
      - Risk of acknowledgment implosion (no suitable for large group)

Data units

- Information data units
  - DATA : user data unit
  - FIRST : connection setup data unit
  - DIAG: data unit for error notification
- Control data units
  - CNTL: general control data unit
  - ECNTL: error control data unit
  - TCNTL: traffic control data unit
  - JCNTL: control data units for the join process to a multicast group

Connection Control

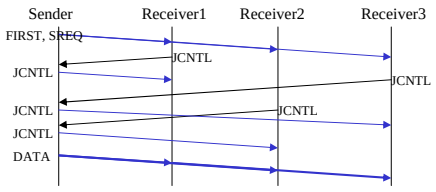
- Flags in the header of each XTP data unit:
  - RCLOSE
    - The sender is not accepting any more user data
    - Receiving direction is closed
  - WCLOSE
    - The sender will not sending any more user data (but it can continue to receive data)
  - END
    - Termination of a connection

Connection Setup

- Sender initiates connection setup
  - A FIRST data unit is broadcast with sender address, multicast group address, traffic parameters, MULTI bit set, RCLOSE bit set (single sender multicast connection)
  - Support known or unknown groups:
    - If SREQ bit of the FIRST data unit is set by the source
      - The joining members have to respond
      - Else receivers join silently

Known Connection Setup

- If a listening hosts receives a FIRST unit data
- The listening host responds with a JCNTL data unit
  - Unicast directly to the sender with traffic parameters
    - Other members are not aware of the group membership
    - Traffic parameters can be adapted to and by the receivers



Known Connection Setup (…)

- If the sender accepts the connection, it unicasts a JCTNL data unit to the respective receiver
- Hosts may not accept the connection due to traffic parameter
  - It responds with a DIAG data unit instead of a JCNTL
    - Contains a reason for the rejection
- Late join :
  - The new member broadcasts a JCNTL data unit to the multicast group
  - The sender responds, with a JCNTL data unit, directly to the receiver

Connection Release

- Explicit connection release is required :
  - Because XTP multicast service relies on receiver lists
- Three possible scenarios:
  - A single receiver leaving a group
  - Orderly release by the sender
  - End of connection

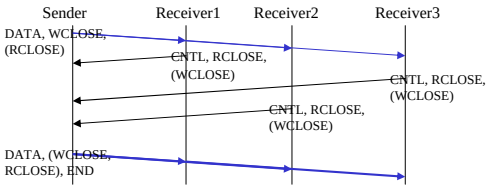
Leaving a Connection

- To leave a group
  - A receiver send a CNTL data unit with an END bit set in the header
  - The multicast sender remove the receiver from the list of active receivers
  - The receive move into a waiting state
    - For a certain period of time
    - It responds only to sender's data unit explicitly unicast to it
    - CNTL data unit can be retransmitted until
      - the sender has acknowledged the request to leave the group, or
      - the number of repeats exceeded a certain threshold

Graceful Connection Release

- Orderly way:
  - Each receiver has correctly received all transmitted data
- Procedure
  - The sender initiates connection release by setting a WCLOSE bit in the header of the data unit (RCLOSE bit is set already)
  - The receiver transmits a CNTL data unit with a set RCLOSE bit
  - When the multicast sender has received the corresponding acknowledgement from all active receivers, it issues a data unit with a set END bit.
  - The sender moves into the waiting state for a certain period of time
    - Incoming data units can still be processed, and response packets generated

Graceful Connection Release



Connection termination

- No data delivery guarantee
  - The multicast sender sends a data unit with an END bit set.
  - It then moves to the waiting state for a specific period of time.
  - Receivers after it receives this data unit, behaves in a similar manner
  - Data unit is not acknowledged and the context is frozen, irrespective of any outstanding data.

Data transfer

- XTP provides a number of protocol functions for data transfer, these functions can be used for multicast communication
  - Flow Control:
    - Prevention of receiver overloading
  - Rate Control:
    - Prevention of congestion in the network
  - Error Control and Reliability

Flow Control

- Flow control mechanism :
  - Sliding window

A receiver grants the sender a transmission credit:
  - *rseq*: sequence number (in bytes)
  - *alloc*: window size (in bytes)
  - Apply to pure user data
- Transmission credit for an entire group:
  - The lowest value of all received credits
  - The slowest member determines the credit granted
- The sender may set a flag to indicate to receivers that it is ignoring flow control

Rate Control

- Rate control mechanism :
  - Every receiver specifies a rate that the sender may transmit during this unit of time
- XTP rate parameters:
  - *rate*: maximum data rate (in bytes per second)
  - *burst*: maximum number of bytes that may be sent within a time interval
  - *Rtimer* = *burst/rate*: length of the time interval
- Rate parameters for an entire group:
  - The lowest value of all received bursts and rates
  - The weakest member determines the rate for all others

Error control

- XTP uses checksums and sequence numbers
- Corrupted data unit is detected through checksum
  - Corrupted data units are discarded, with no further action
  - If a specific flag of the header is set :
    - Only the header of the data unit is considered in the checksum calculation
    - Otherwise it is calculated over the entire data unit

Error control

- Receivers use sequence number to detect:
  - Lost packets
  - Misordering
  - Duplicate packets

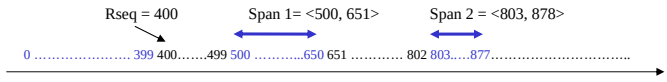
  Note : not afforded to control data unit

Reliability

- XTP acknowledgement process is sender-controlled:
  - Use of selective, negative acknowledgments
  - Receivers send acknowledgment if the sender has requested them to do so.
    - By setting a flag in the header of the data unit: SREQ
- If the group is unknown the SREQ flag may be set
  - Some level of error detection and recovery is possible
  - But reliable reception can not be guaranteed
- With the FASTACK flag,
  - the sender signals receivers to send a negative acknowledgment
  - even if no previous request have been received

Acknowledgments

- An acknowledgement is an error control data unit (ECNTL)
  - Data still missing is identified by spans : pairs of sequence number
    - Bottom limit of the range
    - Top limit of the range
  - Multicast sender merges spans from all receivers
    - Retransmits all requested data
  - Risk of acknowledgement implosion



XTP summary

- XTP v4 provides support for reliable multicast
  - Rate control is used to prevent receiver overrun without feedback
  - Reliable transport service:
    - If group membership is known
    - Through cumulative, selective negative acks and sender retransmissions
    - ACKs are triggered by sender => NAK implosion
- XTP does not scale very well for large and heterogeneous groups:
  - Synchronized receivers are bothered with unwanted retransmissions and bandwidth may be wasted if the number of unsynchronized receivers is small

Conclusion

- Single Sender Multicasting
  - Synchronization of senders is very complex
- Full reliability, some reasonable number of receivers
  - Acknowledgments, retransmission
- Huge receiver number, but partial reliability,
  - Layered coding and FEC
- Choice is made by the application
  - ALC

Bibliography

- Sanjoy Paul. "Multicasting on the Internet and its Applications". Kluwer academic, 1998.
- Katia Obraczka. "Multicast Transport Protocols: A Survey and Taxonomy". IEEE Communications magazine, vol. 36, n°1, 1998.
- B. Whetten, Dah Ming Chiu, M. Kadansky, Seok Joo Koh, G. Taskale, "Tree-Based ACK (TRACK) Building Block for Reliable Multicast Transport", December 2003 .
- Ralph Wittmann, Martia Zitterbart. "Multicast Communication : Protocols and Applications". Academic Press, 2001.
- Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", IEEE/ACM Transactions on Networking, December 1997.
- W. Strayer, B. Dempsey, A. Weaver. "XTP : The Express Transfer Protocol", Addison–Wesley. 1992.

Some implementations

- http://www.irisa.fr/planete/people/roca/mcl
  - Flute [RFC 3926]
  - NORM
  - ALC

Multicast transport and security

- Real issue is receiver-set scaling
  - Authentication of the sender and data integrity
- Data encryption, key distribution (in particular re-keying)
  - Perfect forward privacy is difficult to achieve