Multiparty Transport Overlay Control Protocol (MTOCP) draft-kellil-sam-mtocp-00.txt

Authors:
Mounir Kellil
Christophe Janneteau
Pierre Roux
CEA, LIST

77th IETF Meeting

Anaheim, CA, USA

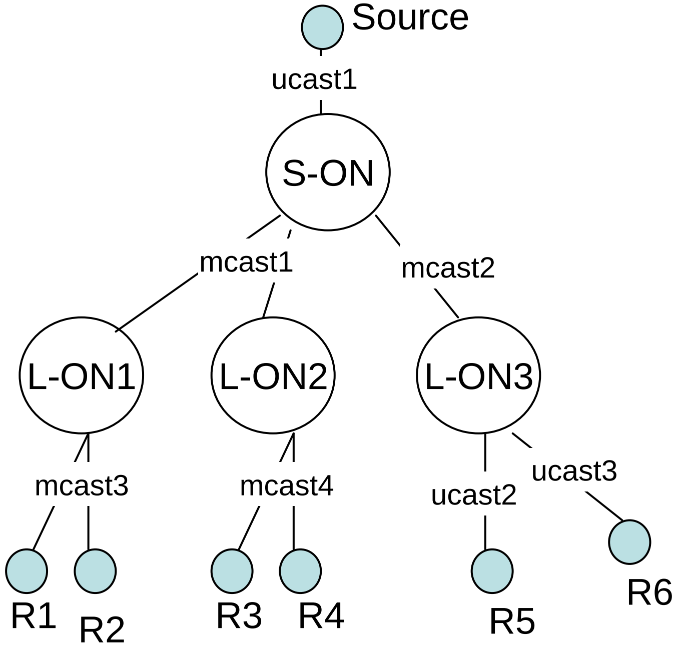
Objective

- Efficient support of group communications
  - Cope with the moderate deployment of IP multicasting in today's Internet.
  - Hide network heterogeneity to the application layer
    - IPv4 and IPv6.
    - Multicast and non-multicast networks.
  - Provide an application-agnostic group data delivery service.
  - Provide a multicast routing-agnostic delivery service.
  - Maximize the use of IP multicast available in the network.

- Solution: Multiparty Transport Overlay Control Protocol (MTOCP)
  - Idea:
    - Place the multiparty overlay paradigm at the transport layer: Multiparty Transport Overlay (MTO).
    - Dynamic addition and removal of MTO’s transport connections.

MTO Architecture Components (1/2)

- MTO Tree: a set of overlay nodes (ONs) interconnected through unicast or multicast transport connections.
  - Overlay Node (ON):
    - Forward data based on local forwarding table (packets received on a given "input" transport connection will be forwarded to one or more "output" transport connections).
    - No IP-in-IP tunneling.
    - 3 types but a common functionality: source overlay node (S-ON), on tree overlay node (O-ON), and leaf overlay node (L-ON).
  - Transport Connections of MTO tree
    - UDP.
    - Unicast or Multicast destination address
    - IPv4 or IPv6 addresses.
    - Transport Connection ID: <source address, source port, destination address, destination port>.



Example of an MTO Tree

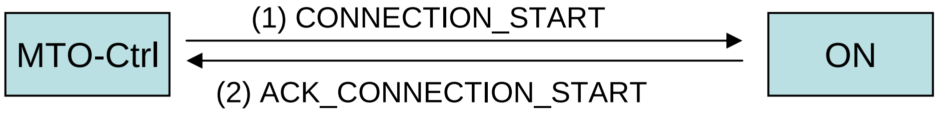
MTO = Multiparty Transport Overlay

MTO Architecture Components (2/2)

- MTO controller (MTO-Ctrl): control the creation, update, and removal of the MTO tree in the network.
  - Define an MTO tree through: an MTO tree ID, a list of ONs and associated transport connections.
  - Assign a couple of unique ports (source and destination ports) per MTO tree.
    - Ports SHOULD be used for all transport connections belonging to the same MTO tree.
    - MTO-Ctrl MAY choose to use a different destination port for a given connection of an MTO tree.
      - e.g., the destination port is the listening port of a unicast terminal (MTO-Ctrl MAY learn the destination port through an out-of-band mechanism).
  - Push to (and further update for) each ON its MTO Tree-specific forwarding table in the form of a list of input and output transport connections.
  - Remove/flush transport connections from ONs.

MTOCP Protocol - Operations

- MTOCP: operates between the MTO-Ctrl and ON to manage the MTO tree.
- Message exchange is initiated by MTO-Ctrl.
- Use of TCP for MTOCP message transport.
- Three types of exchanges
  - Connection Addition
    - Add one or multiple connections of a given MTO tree



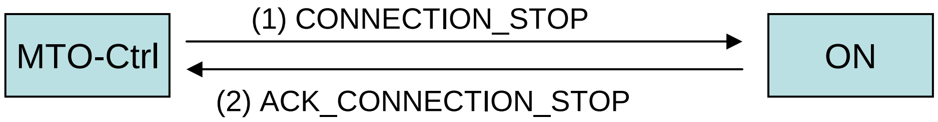
MTO-Ctrl

(1) CONNECTION_START

(2) ACK_CONNECTION_START

ON

  - Connection Removal
    - Remove one or multiple connections of a given MTO tree



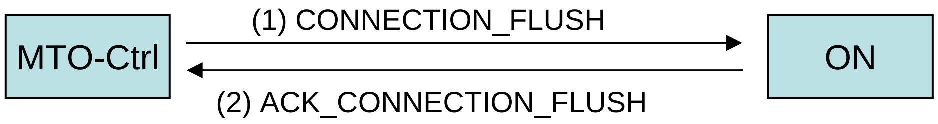
MTO-Ctrl

(1) CONNECTION_STOP

(2) ACK_CONNECTION_STOP

ON

  - Connection Flush
    - Remove all the connections of a given MTO tree



MTO-Ctrl

(1) CONNECTION_FLUSH

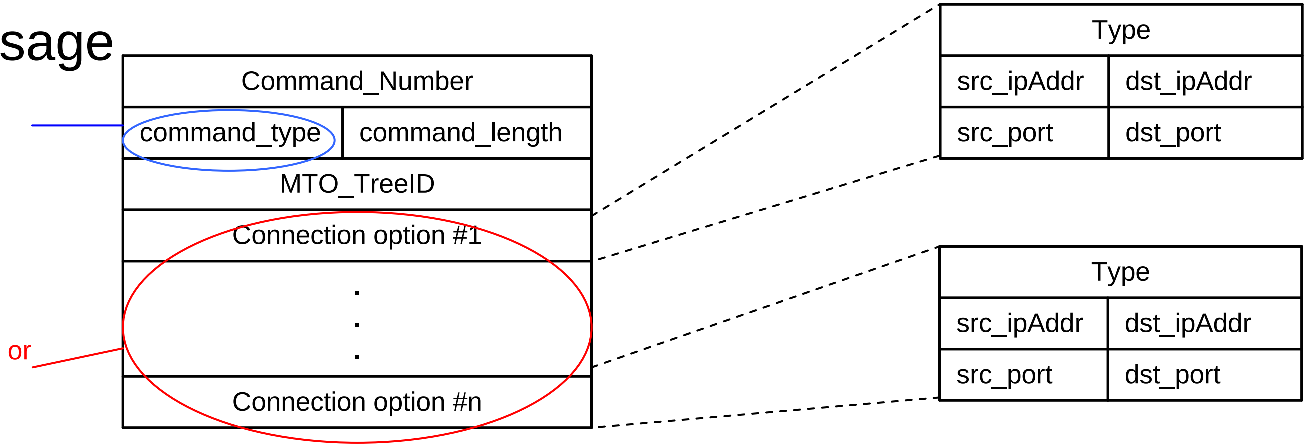
(2) ACK_CONNECTION_FLUSH

ON

MTOCP Protocol – Message Structure

- Command Message

CONNECTION_START,
CONNECTION_STOP, or
CONNECTION_FLUSH



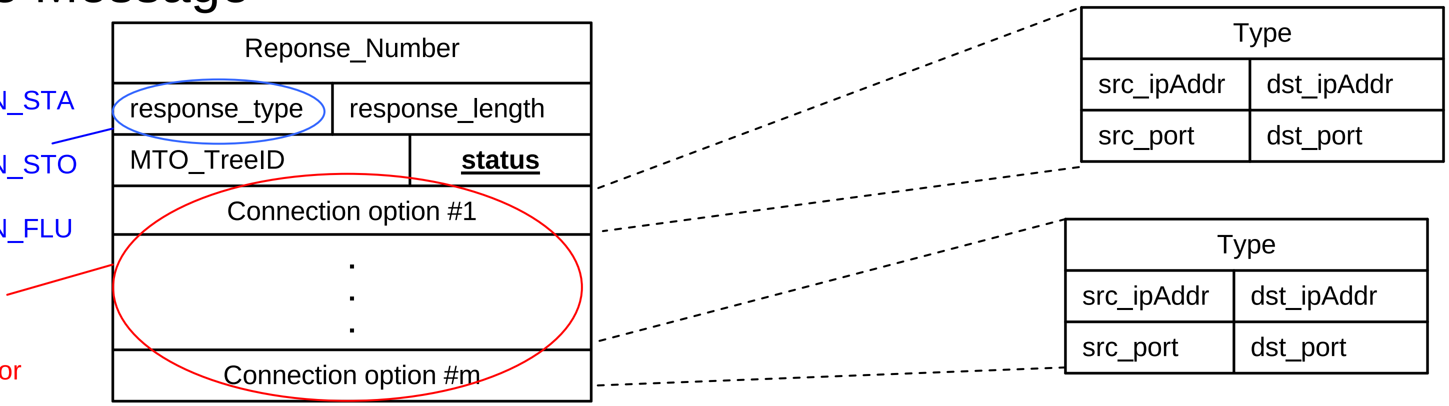
Command_Number	
command_type	command_length
MTO_TreeID	
Connection option #1	
. . .	
Connection option #n	

Type	
src_ipAddr	dst_ipAddr
src_port	dst_port

Type	
src_ipAddr	dst_ipAddr
src_port	dst_port

Case of
CONNECTION_START or
CONNECTION_STOP

- Response Message



Reponse_Number	
response_type	response_length
MTO_TreeID	**status**
Connection option #1	
. . .	
Connection option #m	

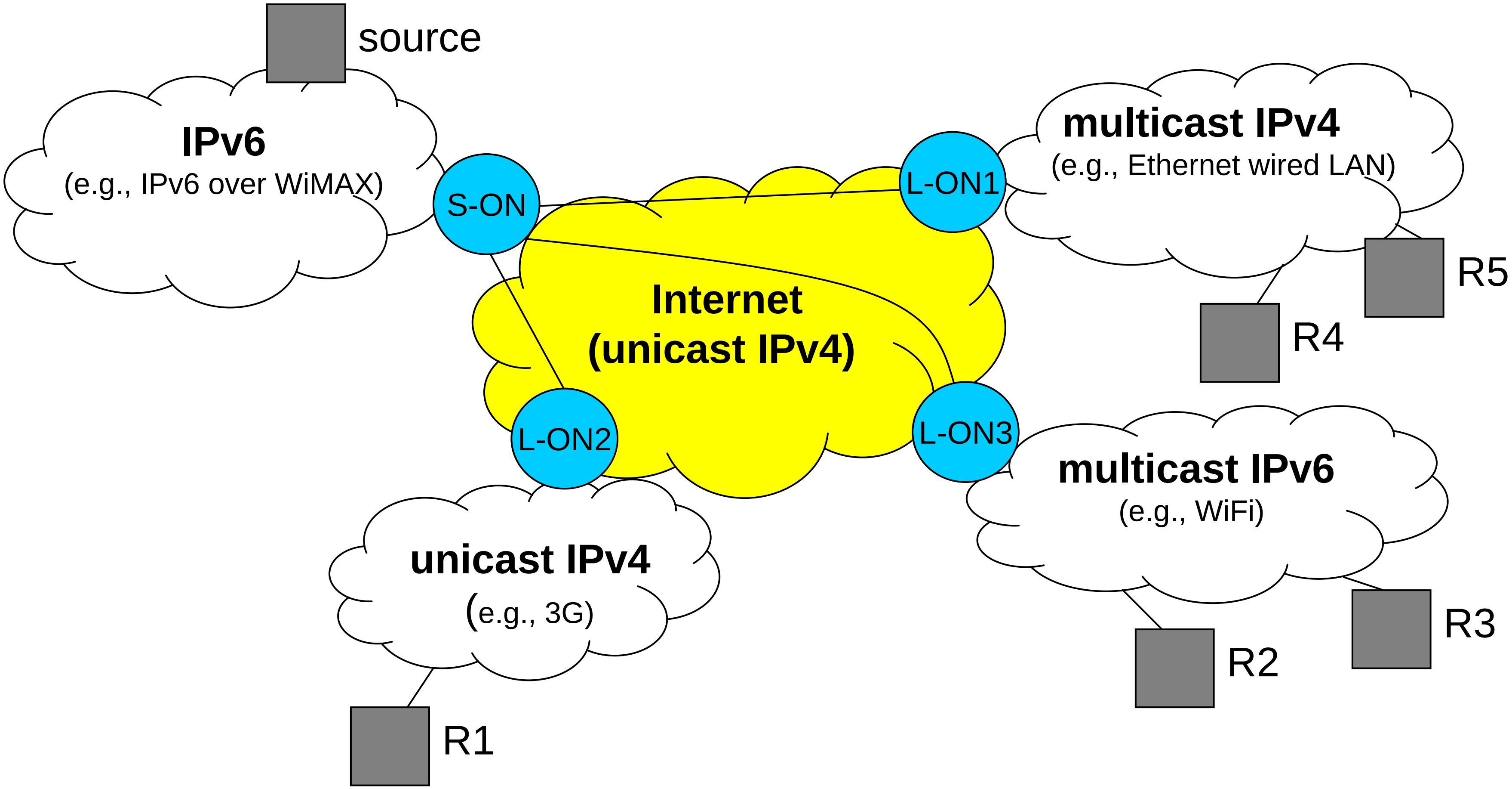
Type	
src_ipAddr	dst_ipAddr
src_port	dst_port

Type	
src_ipAddr	dst_ipAddr
src_port	dst_port

ACK_CONNECTION_START,
ACK_CONNECTION_STOP, or
ACK_CONNECTION_FLUSH

Case of operation failure
related to
CONNECTION_START or
CONNECTION_STOP

Example of MTO Tree Deployment in Today’s Internet



MTOCP Implementation Status

- IPv4 support: done.
- IPv6 support: in progress.
- MTOCP implementation integrated in a full context-aware multiparty transport platform (EU FP7 Project C-CAST):
  - MTOCP, and
  - Network context-aware service
    - Network capability detection (multicast, IPv4, IPv6, etc).
    - Dynamic creation and push of ONs and unicast terminal lists to MTO-Ctrl.
  - Leverage of standard protocols
    - SIP protocol for multiparty session initiation (session invitation to terminals).
    - IGMP/MLD for multicast terminals to subscribe to multicast groups.
    - PIM-SM for multicast routing.