



Elevating RTP to Protocol

Jonathan Rosenberg
Henning Schulzrinne
Bernard Aboba

Bell Labs/Columbia/Microsoft

41st IETF



Problem Definition

- RTP Packets are hard to classify inside the network!
 - RTP is not a well known port or protocol
 - No stateless way to find an RTP packet
- Many key emerging applications which require network to know contents of packets
 - Differentiated Services - network periphery must classify
 - RTP Header Compression
 - Firewalls
 - Static router queue and scheduler management based on 5-tuple



Solution

- Make RTP its own protocol
 - Protocol field in IP header now has a new value indicating “RTP”
 - RTP packets still keep UDP header, overall format unchanged except for new protocol field value
 - Now RTP packets clearly identified inside network



Implications

- End systems must now support an RTP socket API
 - New protocol type (IPPROTO_RTP) in Berkeley sockets socket() call - that's it
 - Applications must still do RTP header
 - OS processes data written to RTP socket just like UDP, except change in IP protocol field value - simple



Implications

- Routers, firewalls, NAT boxes
 - Must be able to recognize new value in IP protocol field
 - Shouldn't be a problem for routers which take numerical arguments for the protocol field classifiers
 - Firewalls will likely drop all RTP packets, as they won't recognize the value



Backwards Compatibility

- “old” RTP and “new” RTP wish to communicate
 - new RTP must determine that old RTP can’t receive the packets
- Unicast Case
 - New RTP implementations must also be prepared to send packets old way
 - First, try to send packet with RTP protocol identifier
 - If you receive an ICMP protocol unreachable error, switch to old UDP protocol identifier
- Multicast Case
 - Very hard
 - SDP will need to contain an RTP version identifier
 - If sender uses new RTP, and some receivers only can receive old - nothing you can do



Backwards Compatibility

- Application Layer
 - Can also have each application negotiate which one will be used
 - Easily supported in H.323 as a new capability
 - Easily supported in SDP/SIP



Other Approaches

- Assign RTP a range of ports
 - Can be severely restrictive for machines trying to run many real time applications at once
 - But, almost no change in hosts or routers
 - Lots of misclassifies from existing non RTP applications
- Define a DS value to mean RTP
 - Not the real meaning of DS byte - being RTP implies nothing about the service it should receive
 - May change inside network - but packet is still RTP



Conclusions

- Classification of RTP is key
 - Differentiated services, RTP compression, firewalls, etc.
- Make RTP its own protocol
- Requires many changes, each of which are very small
- Backwards compatibility is an issue