

The Multicast Application Ports

Nate Karstens

Stuart Cheshire

Mike McBride

draft-karstens-**pim**-multicast-application-ports

Working Group: intarea

Demultiplexing for Unicast Traffic

- In IP transport protocols, port numbers are used to demultiplex traffic destined to different applications on a host (RFC 7605 §5)
- Traditionally, each application protocol has been assigned a unique port from the IANA registry
- Port assignments are a relatively scarce resource
- When running two copies of the same application on a host static port assignment struggles
- Using a dynamic port and advertising it using DNS-SD has reduced the need for static port assignments

Demultiplexing for Multicast Traffic

- Multicast is more complicated because all hosts in the multicast group must have the same port available
- Hard to coordinate without global static port assignment
- But...
 - Multicast applications *don't need ports* for demultiplexing!
 - The multicast group destination address *already* identifies the receiving application
- For multicast traffic the destination port field is redundant
 - If a multicast packet arrives at a host, the host should use it!

Multicast Addresses

- Fixed addresses
 - IANA IPv4 & IPv6 Multicast Address Space Registries
- Dynamic addresses
 - RFC 2730 (MADCAP)
 - draft-ietf-pim-ipv6-zeroconf-assignment
 - draft-ietf-pim-gaap

Proposed Solution

- Assign UDP ports specifically for use with multicast applications
 - 49150, 49151 (0xBFFE, 0xBFFF, last two ports of the “user” port range)
 - Consecutive ports to support RTP/RTCP (see RFC 3550 §11)
- Conformant applications set `SO_REUSEADDR` or `SO_REUSEPORT` socket options to share the port with other applications
 - Therefore, host stacks do not need to be updated, though it would be beneficial
- Conformant hosts always act as if these ports are shared

socat Demo (See IETF 118 pim meeting)

- Transmitter, Terminal 1:

```
socat STDIN UDP4-DATAGRAM:239.0.0.1:49151,ip-multicast-if=172.16.6.100
```

Output interface address

- Transmitter, Terminal 2:

```
socat STDIN UDP4-DATAGRAM:239.0.0.2:49151,ip-multicast-if=172.16.6.100
```

- Receiver, Terminal 1:

```
socat UDP4-RECVFROM:49151,bind=239.0.0.1,ip-add-membership=239.0.0.1:172.16.1.132,fork STDOUT
```

Bind to multicast address

Interface address

- Receiver, Terminal 2:

```
socat UDP4-RECVFROM:49151,bind=239.0.0.2,ip-add-membership=239.0.0.2:172.16.1.132,fork STDOUT
```

Receive multiple messages without terminating



239.0.0.1 →

239.0.0.2 →



Demo Code

- <https://github.com/ietf-wg-pim/draft-karstens-pim-multicast-application-ports>
 - `mcast_app_ports_posix.c` (Linux & macOS)
 - Bind to multicast address
 - Use `IP_ADD_MEMBERSHIP` socket option to join group
 - Use `SO_REUSEADDR` to support more than local one application using group
 - `mcast_app_ports_windows.c`
 - Bind to interface address
 - Use `IP_ADD_MEMBERSHIP` socket option to join group
 - Use `SO_REUSEADDR` to support more than local one application using port

To Discuss

- Security policy
- If two ports are necessary (RFC 3551 assigns ports 5004 and 5005)
- If port 49151 should remain reserved