Mobile Ad-hoc Network (MANET) Multicast Routing Perspective

Brian Adamson
Internet Engineering Task Force 101
March 2018 - London
Outline

• MANET Routing Protocol Synopsis
• Multicast Routing for MANET
• Simplified Multicast Forwarding (SMF)
• Emerging Techniques
• Forwarding Plane Considerations
MANET WG Unicast Routing Protocols

• Original Experimental RFCs:
  • Optimized Link State Routing (OLSR) – RFC 3626
  • Ad-hoc On-Demand Distance Vector (AODV) – RFC 3561
  • Dynamic Source Routing (DSR) – RFC 4728
  • Topology-Dissemination Based on Reverse-Path Forwarding (TBRPF) – RFC 3684

• Proposed Standards:
  • Optimized Link State Routing version 2 (OLSRv2) – RFC 7181
    • Supported by NHDP (RFC 6130) and PacketBB (RFC 5444)

• Other related protocols
  • Routing over Low-Power/Lossy Networks (RPL)
  • BABEL routing protocol
  • OSPF MANET extensions
MANET Multicast Routing

• Challenges:
  • May forward packets via same interface as received
  • Highly dynamic topologies
  • Packet loss often higher with multicast than unicast (e.g., 802.11)
  • Often lower rate transmission than unicast (e.g., 802.11)

• Early Proposed Concepts:
  • Multicast AODV (MAODV)
  • Multicast OLSR (MOLSR)
  • On-Demand Multicast Routing Protocol (ODMRP)
  • Piggybacking plugin for olsr.org code reused routing dissemination forwarding

• Where the working group landed:
  • Simplified Multicast Forwarding (SMF) – RFC 6621
Simplified Multicast Forwarding (SMF)

• Floods multicast packets within a MANET routing area
  • Contention range of many wireless systems greater than communication range so flooding on a small diameter network often not much worse than group-based routing

• Uses duplicate packet detection (DPD) to safely flood
  • Identifier- and hash-based techniques specified for IPv4 and IPv6

• Compatible with efficient flooding techniques enabled by distributed, Connected Dominating Set (CDS) relay set selection algorithms
  • Only relies upon local neighborhood information

• Relay Set Selection Algorithms
  • Source-based Multipoint Relay (S-MPR)
  • Essential Connected Dominating Set (E-CDS)
  • Other variations
  • Enhancement have been explored:
    • Metric-based selection with redundancy
    • “Sticky” techniques for improved stability
SMF Implementations

- *nrlsmf* – cross-platform, user-space forwarding daemon
  - Linux, Windows, Mac OS X, Android, etc. supported
  - Runs stand-alone for classic flooding or can be controlled by NHDP daemon with relay-set algorithms

- *olsrd* – OLSR daemon with Basic Multicast Forwarding (BMF) plug-in
  - Open source at [https://www.olsr.org](https://www.olsr.org)
**nrlsmf’s Place in the World**

- *nrlsmf* provides a user-space forwarding function “playground” for development and experimentation
- Have been exploring forwarding plane considerations for MANET (Forwarding Information Base (FIB) constructs such as DPD state)
Recent MANET Multicast Concepts

• ODMRP-ASYM
  • Can work in topologies with non-reciprocal links
  • References:
    • draft-gerla-manet-odmrp-asym-01 (expired)

• Elastic Multicast
  • Flow-based multicast routing paradigm
  • Builds from flooding mesh provided by SMF using token bucket limited user traffic or "advertisement" control messages to announce flows
  • ACKs from "downstream" nodes (including group members) are propagated "upstream" towards source(s) to keep subset of flooding relays fully active
  • Gateway mechanism to/from enterprise multicast routing protocols (e.g. PIM) can be supported
  • References:
    • draft-adamson-elasticmcast-00 (expired)
Forwarding Plane Considerations

• SMF adds duplicate packet detection (DPD) processing and state as compared to conventional destination-based forwarding
  • *nrlsmf* caches DPD state on a **per-flow** basis to help reduce false duplicates
  • A limited **window** of DPD history is kept per flow

• Flow-based constructs like Elastic Multicast also have some additional state and control-plane interaction requirements
  • New flow detection, flow status updates, etc
  • On-demand protocols (e.g., AODV) also have similar needs

• A common Forwarding Information Base (FIB) specification might be specified for these types of protocols
Example: XORP `mrouted` Interface

```
193 struct mfc_cache {
194     struct mfc_cache *next;  /* Next entry on cache line */
195 #ifdef CONFIG_NET_NS
196     struct net *mfc_net;
197 #endif
198     __be32 mfc_mcastgrp;    /* Group the entry belongs to */
199     __be32 mfc_origin;    /* Source of packet */
200     vifi_t mfc_parent;    /* Source interface */
201     int mfc_flags;       /* Flags on line */
202
203     union {
204         struct {
205             unsigned long expires;
206             struct sk_buff_head unresolved; /* Unresolved buffers */
207         } unres;
208         struct {
209             unsigned long last_assert;
210             int minvif;
211             int maxvif;
212             unsigned long bytes;
213             unsigned long pkt;
214             unsigned long wrong_if;
215             unsigned char ttls[MAXVIFS]; /* TTL thresholds */
216         } res;
217     } mfc_un;
218 }
```

```
mroute.h
188     __be32 mfc_mcastgrp;
189     __be32 mfc_origin;
200     vifi_t mfc_parent;
201     int mfc_flags;
union {
    struct {
        unsigned long expires;
        struct sk_buff_head unresolved; /* Unresolved buffers */
    } unres;
    struct {
        unsigned long last_assert;
        int minvif;
        int maxvif;
        unsigned long bytes;
        unsigned long pkt;
        unsigned long wrong_if;
        unsigned char ttls[MAXVIFS]; /* TTL thresholds */
    } res;
} mfc_un;
```

**DPD?**

**Other stuff**

**SMF / EM**

**Multiple tables**
Example: \textit{nrlsmf} Forwarding Architecture

**Host operating system**

- **class SmfApp** (aka \textit{nrlsmf})
  - \texttt{class SmfApp : ElasticMulticastController}

- **class Smf** (aka \textit{ElasticMulticastForwarder})
  - \texttt{class Smf : ElasticMulticastForwarder}

- \texttt{Smf::Interface::Associate} defines forwarding relationship from an input interface to an output interface
  - An \texttt{Smf::Interface} may "self-associate" for forwarding on a MANET interface
- Forwarding decision is conducted in the \texttt{Smf::ProcessPacket()} method
  - \texttt{Smf::ProcessPacket()} code enhanced to conduct Elastic Multicast forwarding (principally at the post-DPD stage)
  - Detected flows are aged / pruned in the \texttt{Smf::OnPruneTimeout()} method
- Elastic Multicast controller functionality embedded in \texttt{class SmfApp} code
Controller/Forwarder Relationship

class ElasticMulticastController

Control Plane
(e.g., user space)

Routing Control
(e.g. olsrd, nhdp, babel, rpl, etc)

“Membership” Table

class ElasticMulticastForwarder

Forwarding Plane
(e.g., kernel)

Forwarding Engine
(e.g., SMF)

Forwarding Information Base (FIB)

Control Interface API
(e.g., netlink, protobufs, etc)
Conclusion

• MANET multicast routing concepts have some added considerations as compared to conventional multicast routing specifications.

• A common FIB and supporting control plane interface specifications could be developed to support these protocols as well as some unicast MANET/ROLL/BABEL protocols with special needs.

• MANET is chartered to develop this FIB looking for input on interests and best approaches to provide the largest benefit.
Backup Slides
1. An entry may be dual-hatted as “managed” and “dynamic”
2. “Managed” entries are removed only by Controller
3. “Dynamic” entries inherit policy/parameters from matching “managed” entry
4. Non-matched dynamic entries are not acked or forwarded and follow default policy (unicast support will likely require explicit default policy entries to be made)
The Key: FlowDescription

- **class MulticastFIB::FlowDescription**
- Key is a tuple of `dst:src:class:protocol:ifaceIndex`
- Fields may be wildcarded
  - “Best match” search and prefix-based / matching iterators are provided using ProtoTree data structure (Patricia trie radix tree)
  - Could be extended to include port numbers for application-specific policies
- Controller uses FlowDescription with interface index as key for membership table entries
  - Can support ASM and SSM memberships and unicast flows for Adaptive Routing
- Forwarder uses FlowDescription as key for forwarding table entries (does not use interface index field)
- FlowDescriptions and interface indices are used as reference for controller<->forwarder interaction
  - E.g., a protobufs or netlink structure could be defined
- Prefix mask lengths allow prefix subnet entries for `dst` and/or `src` addresses
  - `dstLength` or `srcLength == 0` is “wildcard” `dst` or `src` address
    (otherwise length of associated address field in bytes)
  - `dstMask` and `srcMask` are prefix mask lengths (in bits) for `dstAddr` and `srcAddr`
  - `class == 0x03` is “wildcard” traffic class (i.e., ECN bits)
  - `protocol == RESERVED (255)` is “wildcard” protocol type
  - `index == 0` is “wildcard” interface index