NRL Simplified Multicast Forwarding (*nrlsmf*) Implementation *Approach & Experiences*

Brian Adamson

4 Aug 2005

63rd IETF - Paris, France

Overview

- NRL has created an open-source SMF implementation entitled "nrlsmf"
- *nrlsmf* provides a forwarding module that:
  - Works in conjunction with NRL's OLSR implementation "*nrlolsrd*", but
  - Provides an open control interface such that alternative process could control its operation, or
  - Can perform "classical flooding" in a stand-alone mode.
- Supports IPv4 and IPv6 forwarding.
- Code base supports Linux, MacOS/BSD, Win32, and WinCE platforms.
- NRL has also extended its *ns-2* "ProtoManetKernel" to support SMF forwarding. (OPNET implementation in progress)
- Code available via <http://smf.pf.itd.nrl.navy.mil>

Approach

- *nrlsmf* leverages NRL's cross-platform "Protolib" C++ toolkit to create a "user space" forwarding engine.
- Uses "ProtoCap" class to capture and forwarding incoming packets
  - "ProtoCap" is a wrapper around Linux PF_PACKET socket, Berkeley Packet Filter (bpf), libpcap (or WinPcap), or RawEther™ depending upon platform.
- Uses "ProtoDetour" class to intercept outbound multicast packets to re-sequence IPv4 ID field or apply extension header to IPv6
  - "ProtoDetour" is a wrapper around Linux "ip_queue/ip6_queue", BSD "divert" socket, or Win32 NDIS intermediate driver depending upon platform
- An alternate approach using a virtual interface mechanism (e.g., TAP driver) may be considered for future.

Duplicate Packet Detection

- *nrlsmf* uses a Patricia tree structure (ProtoTree) to manage duplicate packet detection entries.
- Each entry maintains a configurable bitmask marking "recent" packet receptions/forwards within a sliding sequence-based window.
- A timeout is used to prune stale entries
  - stale == no recent, non-duplicate packets
  - This also mitigates issues related to source restart, etc
- Some additional state needed for multiple interface support for S-MPR approach.

Limitations/Issues

- Current code snapshot doesn't perform MPR forwarding quite correctly (will be remedied soon)
- Current code snapshot only does "per source" sequencing spaces and IPSec not yet supported. (also to be fixed soon)
- *nrlolsrd* support for E-CDS algorithm computation not fully tested, but preliminary results are consistent with expected results.
- "ip6_queue" only recently was enabled by default in Linux 2.6 kernels
- BSD "divert" socket support for IPv6 not yet tested.
- IPv6 currently not supported by *nrlsmf* on WinCE.
  - *Win32 OS support for IPv6 forwarding limited to only certain platforms anyway.*
- "ProtoDetour" wrapper around Win32 NDIS intermediate driver not yet completed.

Some Experiences

- Have successfully demonstrated multicast VoIP, reliable multicast and other apps on multihop 802.11 networks.
- Have active projects to integrate this technology with some other radio technologies
- 802.11 multihop multicast performance quite fragile (perhaps some per-hop retransmission strategy would be useful?)
- SMF (and perhaps MANET routing in general) might benefit if stable, quality routes were sometimes favored over shortest hop?
  - Some environments (indoor, handheld devices, urban) have extremely dynamic short term link characteristics
- Aside: *Promiscuous reception could be helpful for unicast flows in highly dynamic conditions.*

Conclusions

- "Native" Layer 3 forwarding of multicast in wireless can be accomplished without custom encapsulation.
- Some more study needs to be done of quality of resultant service (I.e., multihop, multicast reliability)
- Kernel/operating system support of SMF forwarding heuristics would be nice.