

OSPF WG meeting, IETF62

OSPF MANET Design Team update

March 8, 2005

Tom Henderson

{thomas.r.henderson@boeing.com}

A brief history

- MANET WG standardized a set of Experimental RFCs
- Initial problem statement drafted
 - draft-baker-manet-ospf-problem-statement-00 (expired)
- Initial drafts on an OLSR-like adaptation of OSPF, and database exchange optimizations
- WG decides to charter a design team (2004)
 - Meetings in San Diego and Washington, and design-team mailing list

Problem statement

1. Focus on OSPFv3 and not OSPFv2
2. Compatibility with non-wireless OSPFv3
3. Intra-area extensions only
4. Not focusing on transit network case, but should not be precluded
5. Scaling goal is 50-100 nodes on wireless channel
6. Leverage existing MANET work where possible
7. Use RFC 3668 guidance on dealing with IPR claims

Consensus reached so far

- Working on defining a new MANET **interface type** rather than a MANET **area type**
 - in parallel with existing OSPF interface types
- Focusing first on designing an **optimized flooding mechanism** for new LSA generation
 - using acknowledged (reliable) flooding
- Additional optimizations (more efficient Hellos, DB exchange) a lesser priority for the design team
- Focus on two active I-Ds
 - draft-chandra-ospf-manet-ext-02.txt
 - draft-ogier-manet-ospf-extension-03.txt

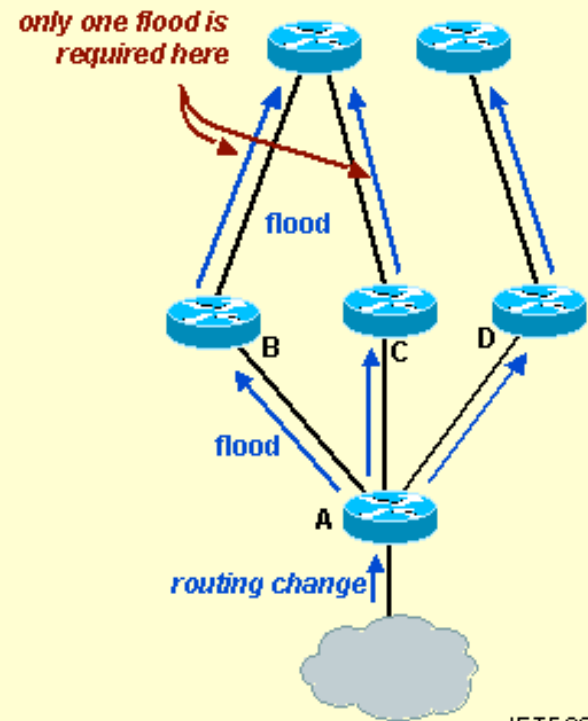
Draft overview

- Both drafts focus on selecting more efficient Relay Node Sets (RNS) for flooding
 - A “Connected Dominating Set” (CDS)
- Differences
 - Source Independent vs. Source Dependent CDS
 - Use of Hellos or LSAs for dissemination of two-hop neighborhood information
 - Differential (Incremental) Hello implementations
 - Ogier draft addresses the minimization of adjacencies formed in a dense network

Review of draft-chandra*

Optimized Flooding

- New reachability and topology change information is flooded to all adjacent neighbors
 - Several copies of an LSA may reach a neighbor two hops away
 - This is wasteful of available bandwidth and processing power
- *We need to optimize flooding*



draft-chandra-ospf-manet-ext-01.txt

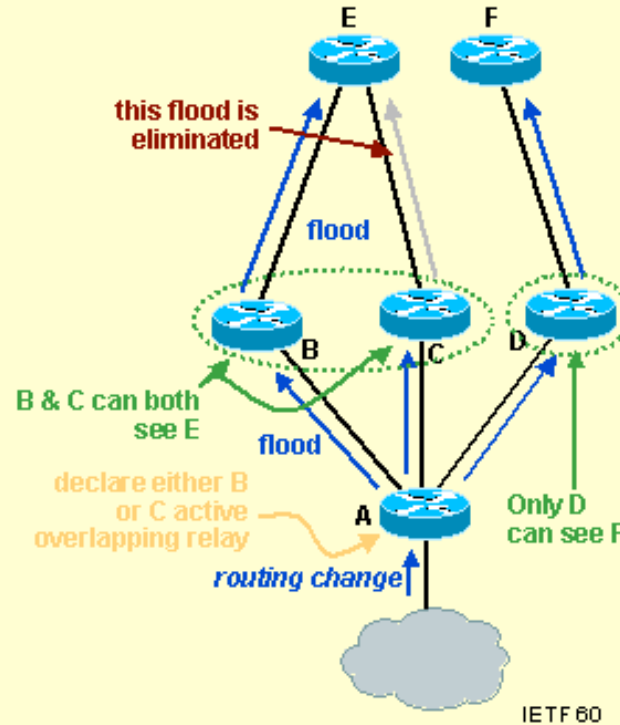
IETF 60

* from Proceedings of OSPF WG, IETF-60

Review of draft-chandra (2)*

Optimized Flooding

- Find common "two hop" neighbors
 - Group neighbors based on their neighbor sets
- Calculate minimum set of overlapping relays
 - Pick one neighbor from each group of neighbors with the same "two hop" neighbors
- Signal overlapping relays to flood LSAs
- Remaining neighbors do not reflood learned information (they backup the active overlapping relay)



draft-chandra-ospf-manet-ext-01.txt

IETF 60

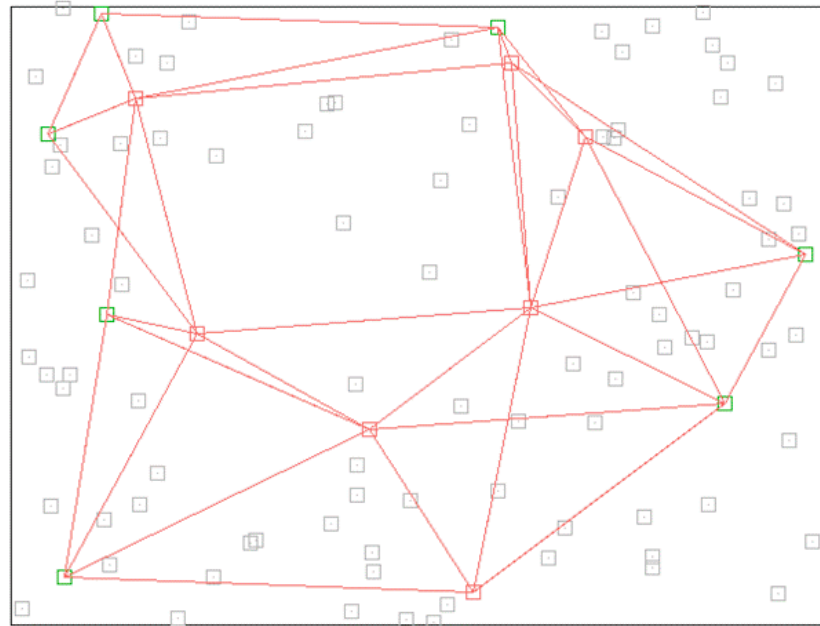
* from Proceedings of OSPF WG, IETF-60

Preview of draft-ogier*

Simulation in Mobile Networks (cont.)

Biconnected backbone consisting of DRs (red) BDRs (green), and adjacencies between them (red lines).

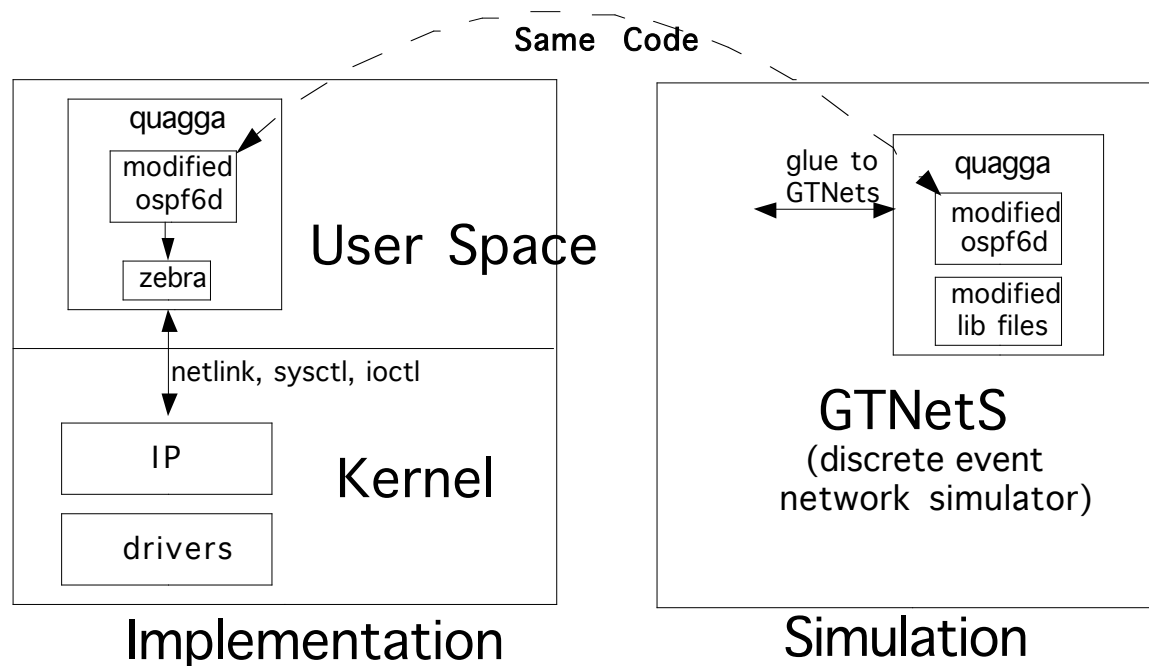
Computed by Essential CDS algorithm.



* from Richard Ogier's presentation (to follow this presentation)

Evaluation software

- Based on quagga open source OSPFv3 routing daemon
 - <http://www.quagga.net>
- Runs as Unix implementation, or as GTNetS simulation (same quagga code)
 - <http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/>
- Implements draft-chandra-ospf-manet-ext-02.txt (so far)



Preliminary simulation results

- Criteria for evaluation include:
 - Stability of relay-node-set selection
 - Overhead due to flooding
 - Robustness of routing
 - Stretch factor for data and overhead packets
 - Run-time complexity of algorithm
- Initial simulation results indicate
 - most overhead is due to LSA flooding/dissemination
 1. flooding must be made more efficient
 2. must minimize the number of LSAs generated (topology control)
 - draft-chandra reduces overhead by focusing on efficiency

Next steps

- Initial goal of draft output to WG (March) now delayed until summer IETF
 - Design Team to report out to the OSPF WG
- Implementation of Richard Ogier's draft for simulator/implementation is underway
- Consider whether to define additional wireless optimizations for this phase
 - or else defer to a later design cycle