Virtual Ring Routing

Network Routing Inspired by DHTs

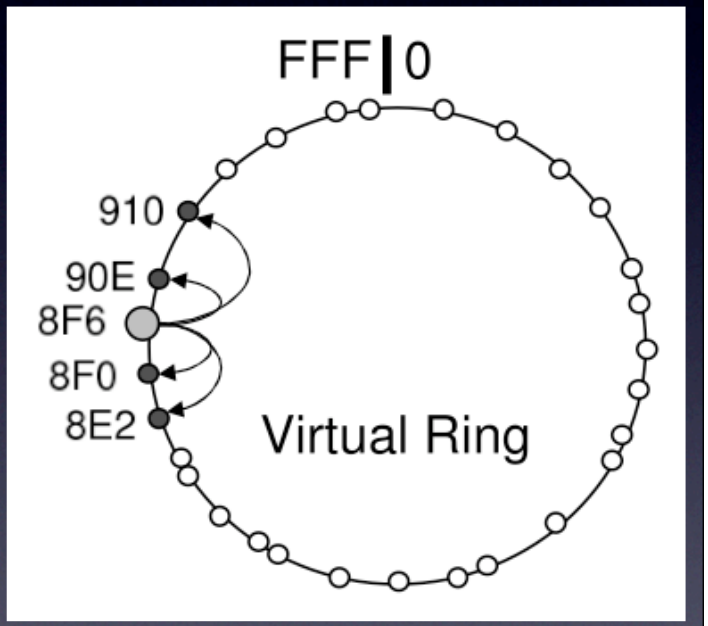
Jonathan Hunt, George Nikolaidis, Nikolas Stephan, Frank Vaisman

Introduction

- Virtual Ring Routing: routing protocol :-)
- Inspired by Distributed Hash Tables
- No flooding
- Location-independent identifiers
- Works on top of any link-layer technology
- Inherently scalable

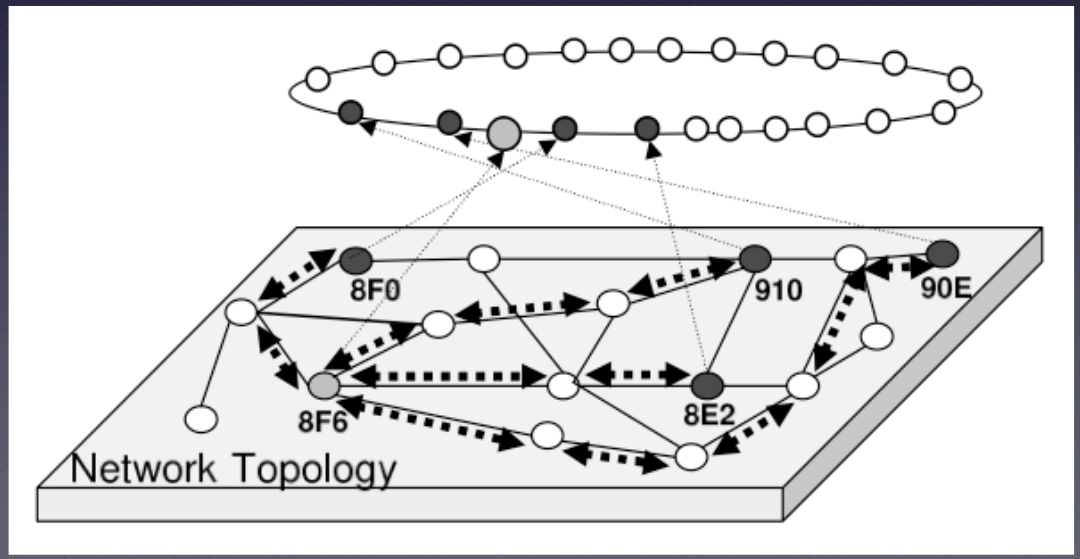
Overview

- Nodes are identified by random integers
  - Fixed, unique, location-independent
- Organised by node identifier within a virtual ring
- Each node maintains a *vset* and a *pset:*
  - Virtual neighbour set of its immediate neighbours within the virtual ring
  - Physical neighbour set of nodes within one-hop link-layer range



Overview

- Paths to virtual neighbours are usually multi-hop
- *vset-paths* are stored both at endpoints and nodes on the way
- Always route to the node whose identifier is closest to that of the destination
- Provides an inbuilt DHT API for P2P applications



Forwarding

- Each *vset-path* is identified uniquely by the pair {path id, endpoint$_A$}
- Includes paths where the node is an endpoint, as well as those for which it is on the route
- If multiple paths are available, one-hop paths are preferred

endpoint$_A$	endpoint$_B$	next$_A$	next$_B$	path id
8F0	8F6	20E	null	03
8E2	8F6	F01	null	2F
8F6	90E	null	7E2	1E
910	8F6	F01	null	2F
35F	37A	20E	7E2	12
A01	A10	F01	FC1	F0
8F6	20E	null	20E	FF
8F6	F01	null	F01	FF
8F6	7E2	null	7E2	FF
8F6	FC1	null	FC1	FF

Routing table for node 8F6

Node joins

- A node populates its *pset* by listening for "*hello*" messages broadcast by other nodes
- It uses its *pset* to send a *setup_req* to itself
- A virtual neighbour receives this *setup_req* and replies with a *setup* message
  - *setup* message includes that node's *vset*
- The node leverages this received *vset* to route *setup_req* messages to its other virtual neighbours

Node and link failures

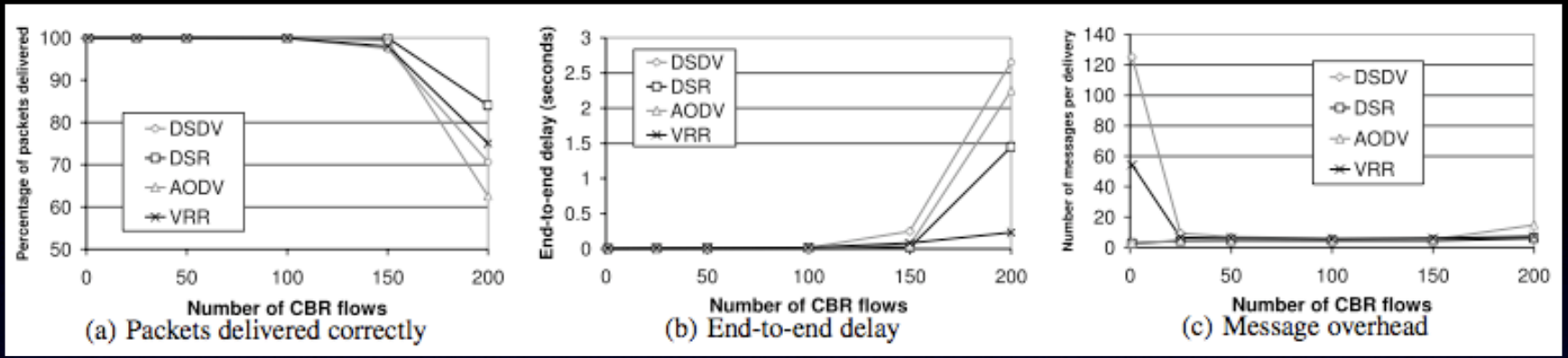
- Proactive failure detection
- Symmetric failure detection
  - Both nodes keep the same state for a link
  - Achieved by means of *hello* messages
- Failure repair
  - Tear down a broken *vset-path* along its entire length
  - Attempt to establish a new *vset-path*

Node and link failures

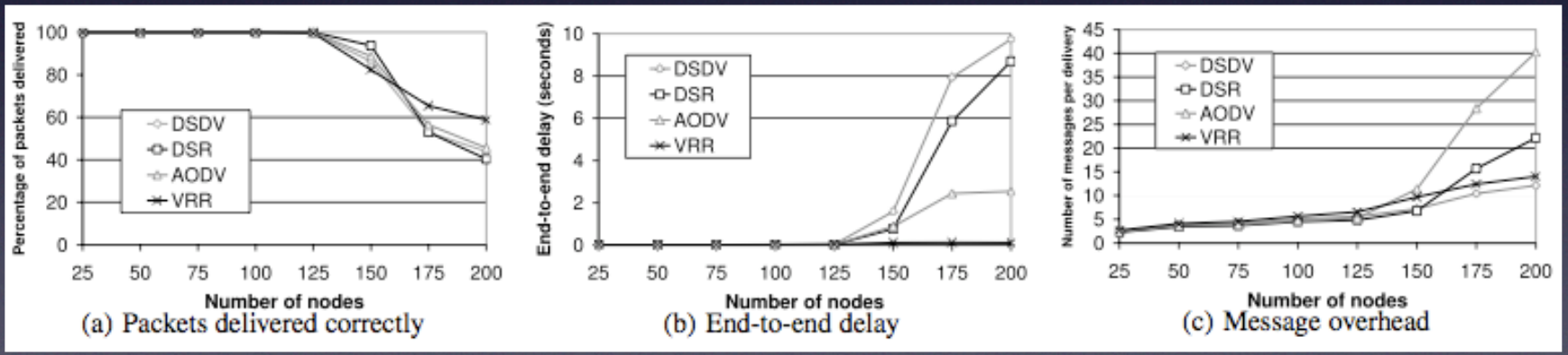
- Local *vset-path* repair
  - Try to substitute a node along a physical path with a neighbour
  - If that fails, tear down the path
- Partitioning
  - Independent rings choose a representative
  - The representative's identity propagates and, if such a physical link exists, it traverses between adjacent rings

Evaluation

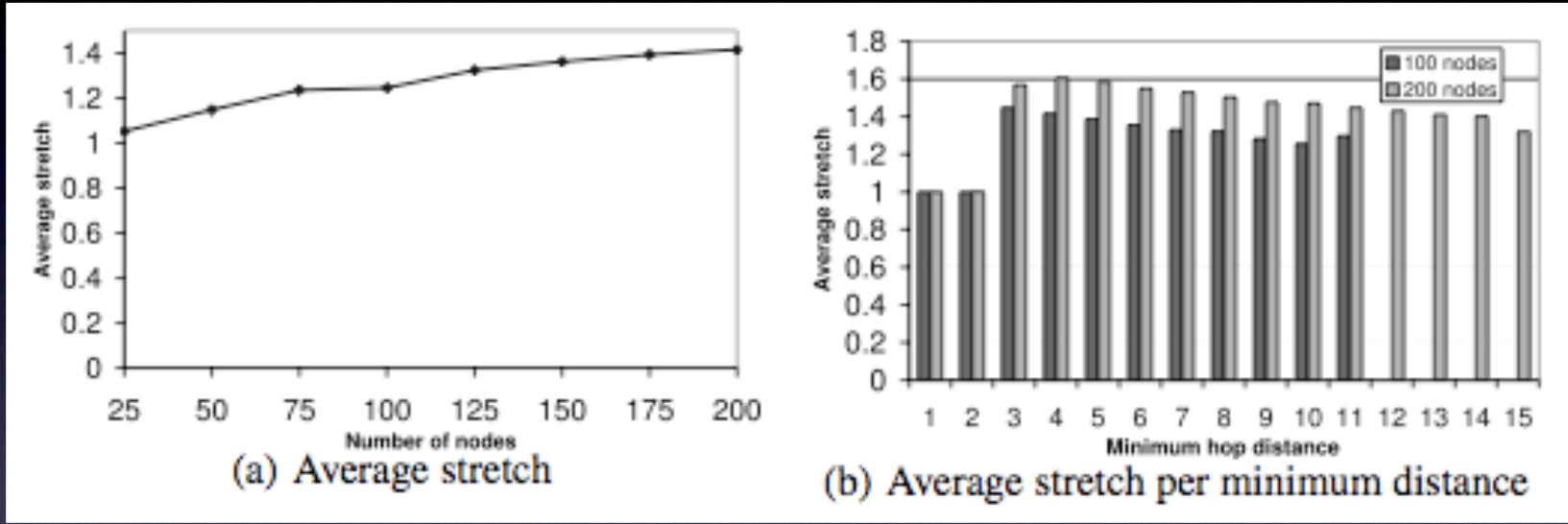
- Simulations with ns-2
  - Comparisons with DSR, AODV and DSDV
  - With and without mobility
  - Varying number of flows and nodes
  - Case of short lived flows tested



Performance with increasing number of CBR flows in the static scenario



Performance with increasing network size in the static scenario

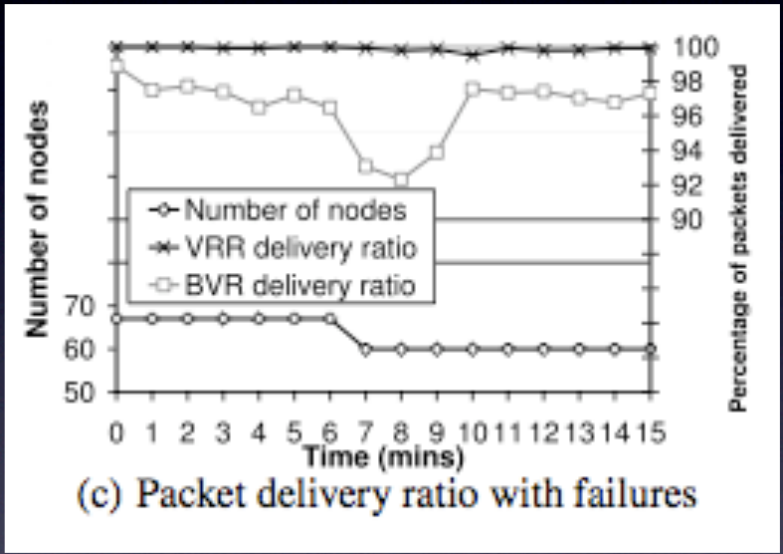


Stretch with short-lived CBR flows in the static scenario

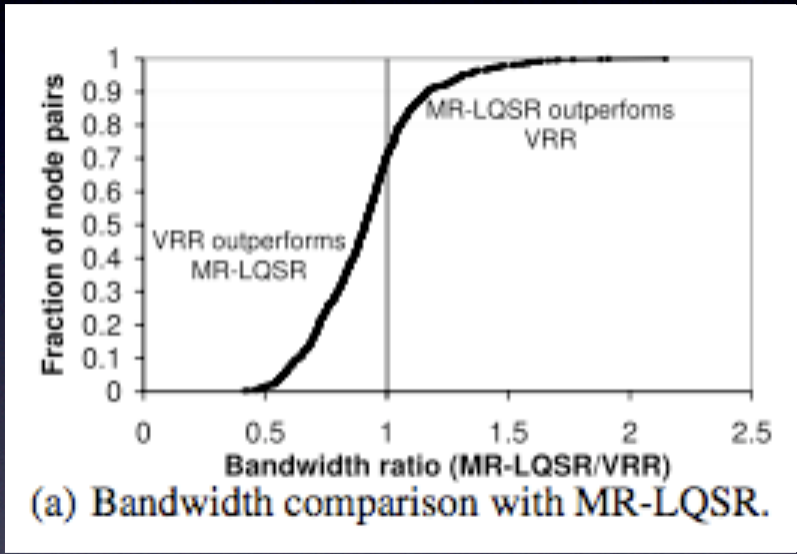
Evaluation

- Sensor network testbed (using BVR)
  - Mica2dot motes
  - Implementation on TinyOS
- 802.11a testbed (using MR-LQSR)
  - Windows PCs
  - VRR under TCP/IP stack

Testbed Metrics



(c) Packet delivery ratio with failures



(a) Bandwidth comparison with MR-LQSR.

Related work

- Two major alternative classes of ad-hoc routing protocols
- Flooding-based protocols
  - Reactive (e.g. AODV, DSR, TORA)
  - Proactive (e.g. DSDV, OLSR, WRP)
- Location-based protocols (e.g. GPSR, BVR)

Critique

- Shows a DHT being used at layer-3, rather than the typical layer-4 use
- Amount of state held on each node covered only very briefly, but seems promisingly low
- Paper is focused more on providing a detailed comparison with other routing protocols rather than providing specific explanations for VRR's test performances

Critique

- Certain sections of the paper provide insufficient detail
  - Stretch not measured against that of other protocols
  - Only a brief mention of end-to-end delays and no detailed justification for the figures produced
  - Reasons given for claimed scalability are only very brief
- Comparisons to other protocols often seem more like attempts to show that VRR "always wins"

Conclusions

- Implementations exist for Windows and Linux

Advantages

- Location-independent identifiers
- No flooding
- Can run above any link-layer technology

Disadvantages

- Provides no guarantees about finding the shortest path

References

- *Zero Servers with Zero Broadcasts,* M. Castro, G. O'Shea, A. Rowstron
- *ROFL: Routing on Flat Labels,* M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, S. Shenker
- *VRR Public Review,* B. Karp
- *Algorithms for Wireless Sensor Networks,* J. Gao