MobileMAN Project
Building Campus-Wide MANETs through Cross-Layering

Andrea Passarella

University of Cambridge
Computer Laboratory

`andrea.passarella@cl.cam.ac.uk`

UNIVERSITY OF CAMBRIDGE

4th Cost 290 Management Committee Meeting

Wuerzburg, Oct 2005

Outline

- MobileMAN Project
  - quick overview
- MobileMAN view
- Selected results
- Efficient p2p systems
- Follow-ups and ongoing works

MobileMAN

- Mobile Metropolitan Ad hoc Network
  - where “metropolitan” stands for “campus-wide” (see later)
- http://cnd.iit.cnr.it/mobileMAN/
- 3 year EC FET-IST Project (5th framework programme)
  - ... in its final stages
- Partners
  - IIT – CNR, Pisa, Italy – Marco Conti
  - Computer Lab – UCAM, UK – Jon Crowcroft
  - Eurecom, France – Refik Molva
  - SUPSI, Switzerland – Silvia Giordano
  - HUT, Finland – Raimo Kantola
  - NetiKos, Italy – Piergiorgio Cremonese

MobileMAN view

- After about 10 years of research, ad hoc networks are still just a research topic – no industry products are coming up
- Lack of efforts on practical/experimental considerations
  - Mainly analytical- and simulation-based research
  - Focus on low-level issues – MAC, routing, transport
  - How MANET behaves in the real-world?
  - Which applications could be valuable to end users?
- Commitments
  - strong realistic approach
  - experimentation & prototyping whenever possible
  - realistic approximations when simulation is needed
  - focus on real (possible) applications

MobileMAN focus

- Large, flat, ad hoc networks are rather unlikely
  - theoretical bounds (throughput drops as 1/n for n-hop connection)
  - confirmed in experimental studies

- "Ad hoc horizon" - Christian Tschudin
  - 10 to 20 nodes
  - 2 to 3 hops

- Real MANETs within the ad hoc horizon – i.e., campus-wide networks
  - build prototypes, i.e., implement full stacks
  - evaluate current solutions
  - propose new, improved, architectures

(Selected) results

- Enhanced 802.11-compliant MAC
  - prototype of the enhanced MAC card
- Prototype-based comparison of different full-stack architectures
  - Using off-the-shelf current solutions
  - Using improved, cross-layer, components
- Real usage scenarios for ad hoc networks – economic analysis e.g., the taxi scenario
  - could ad hoc networks be used as a taxi radio dispatch system?
  - now ... basically free radio dispatch system
  - in future ... provide Internet access through mesh networks

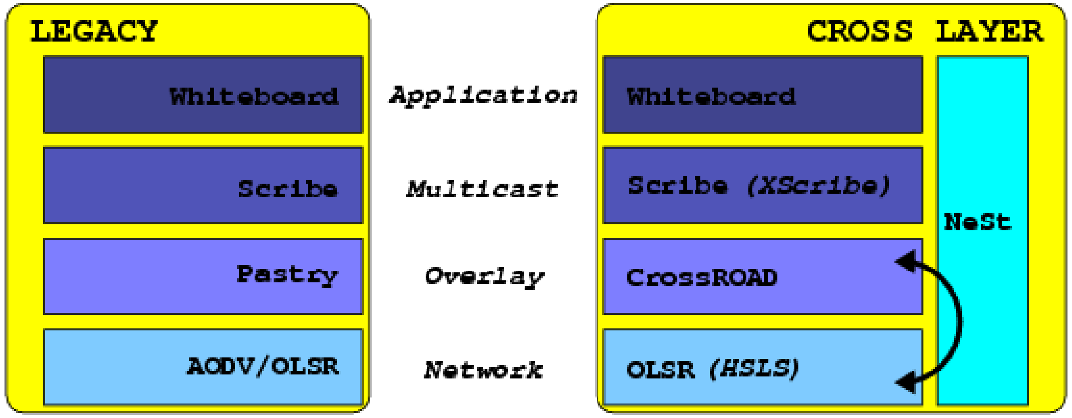
(Selected) results

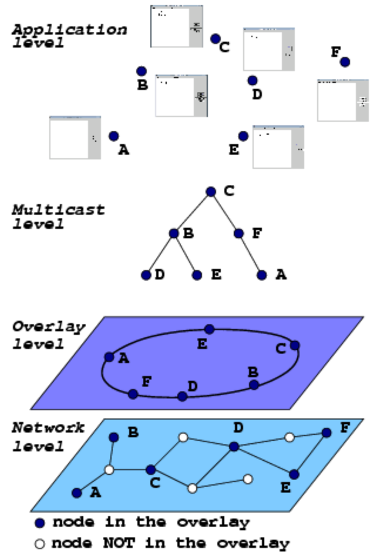
- Enhanced 802.11-compliant MAC
  - prototype of the enhanced MAC card

- Prototype-based comparison of different full-stack architectures
  - Using off-the-shelf current solutions
  - Using improved, cross-layer, components

- Real usage scenarios for ad hoc networks – economic analysis e.g., the taxi scenario
  - could ad hoc networks be used as a taxi radio dispatch system?
  - now ... basically free radio dispatch system
  - in future ... provide Internet access through mesh networks

Architectures: Legacy and Cross-Layer

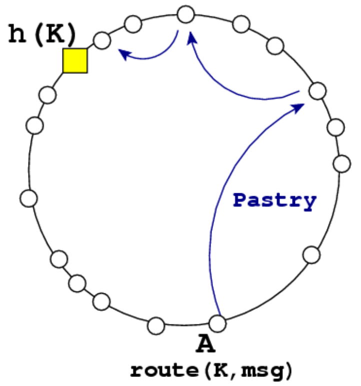




- Use "off-the shelf" components
  - often designed for wired networks
- Overlay network: *Pastry*
- Multicast: *Scribe*
- Do legacy P2P system work well on MANETs ???

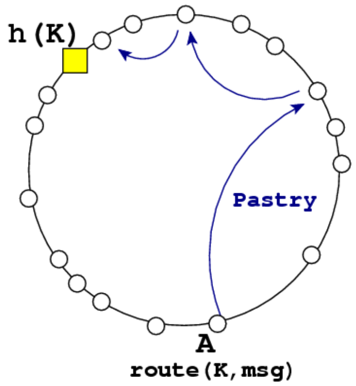
- MANET optimised components
  - cross-layering
- Overlay Network: CrossROAD
  - Pastry optimisation through XL interactions with proactive routing
- Network Status (NeSt)
  - standardises XL interactions
  - joins XL advantages and standard architectures' flexibility

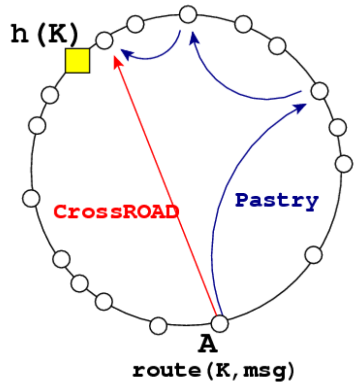
Pastry vs. CrossROAD

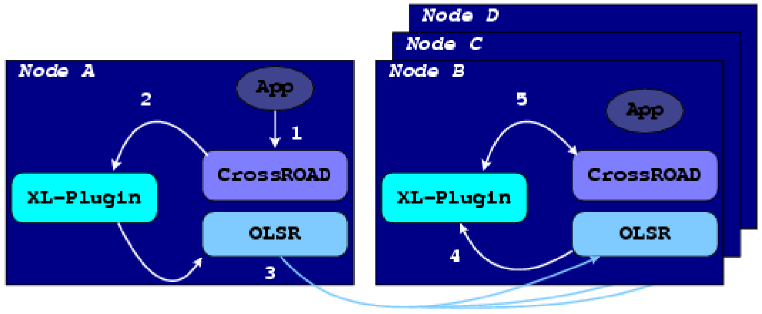


- Logical node id
  - by hashing the IP and port: **`A=h(IP, port)`**
- destinations are specified via some key **`k`** (e.g, “Andrea Mailbox”)
- **`msg`** is routed to the closest node to **`h(K)`**
- build topology anew at the overlay level
  - high management traffic
- store just a subset of node in a local tables
  - to scale on large networks
- multi-hop at the overlay level
- Initial bootstrap

Pastry vs. CrossROAD



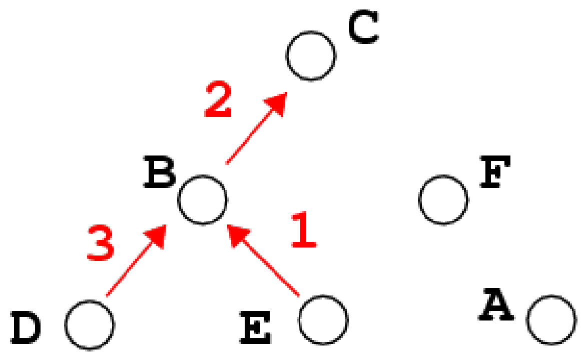


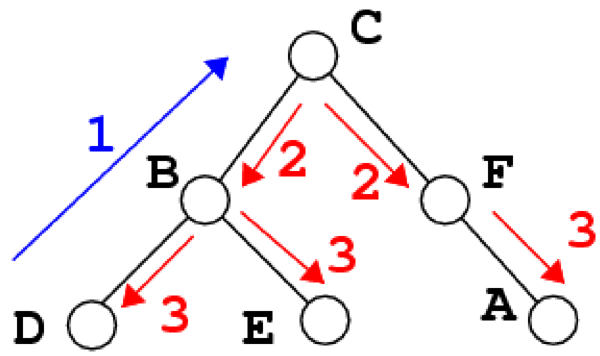


- Logical node id
  - by hashing the IP and port: **`A=h(IP, port)`**
- destinations are specified via some key **`k`** (e.g, “Andrea Mailbox”)
- **`msg`** is routed to the closest node to **`h(K)`**
- build topology anew at the overlay level
  - high management traffic
- store just a subset of node in a local tables
  - to scale on large networks
- multi-hop at the overlay level
- Initial bootstrap

- Same Pastry functionality and API
- Exploit topological information already available at the routing level
- Embed service availability in routing LSU packets (1, 2, 3)
- Get complete view of the overlay topology at all nodes (4, 5)
  - without any additional traffic
- Single-hop at the overlay level
  - multi-hop at the routing level
- Reduced management traffic
- Quick adaptation to topology changes
  - converges as quick as routing
- No initial bootstrap

Scribe





Building the tree

- Multicast group defined by topic **`t`**
- Step 1 ⇨ E:**`route(t,subs)`**, B is next hop in the overlay
- Step 2 ⇨ B:**`route(t,subs)`**, register E as child

  C: I'm the root (the closest to **`h(t)`**)
- Step 3 ⇨ D:**`route(t,subs)`**, B is next hop in the overlay

  B: register D as child, discard subs msg

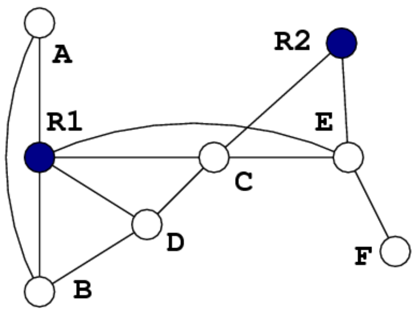
Data Delivery

- Step 1 ⇨ D:**`route(t,msg)`**, reaches the root (C)
- Step 2 ⇨ C:**`route(<children>,msg)`**, reaches B and F
  - one distinct message for each child
- Step 3 ⇨ B,F:**`route(<children>,msg)`**, reaches D,E and A

Pro and Cons

- Very easy to implement on structured overlay
- C might be a bottleneck
- Data delivery is not optimised

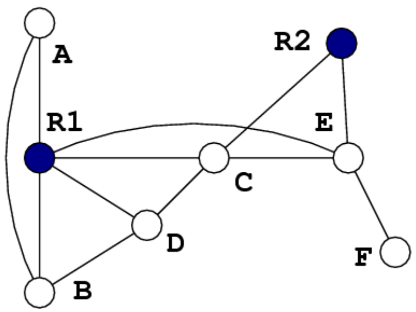
Experiment set-up



Set-up

- 6 nodes in the overlay (running WB)
- 2 more nodes as routers
- C is the intended Scribe root
- small-size MANET, within the ad-hoc horizon

Experiment set-up

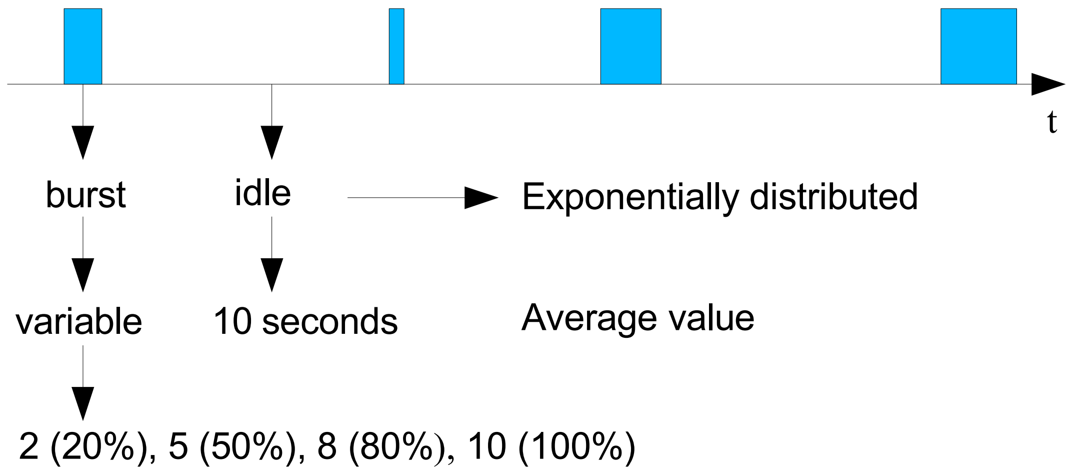


Set-up

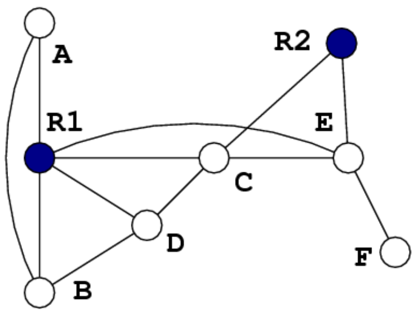
- 6 nodes in the overlay (running WB)
- 2 more nodes as routers
- C is the intended Scribe root
- small-size MANET, within the ad-hoc horizon

Human users

- ON/OFF traffic
- Random burst sizes and idle time duration
- Variable average burst size
  - 100% means 1 stroke/sec



Experiment set-up

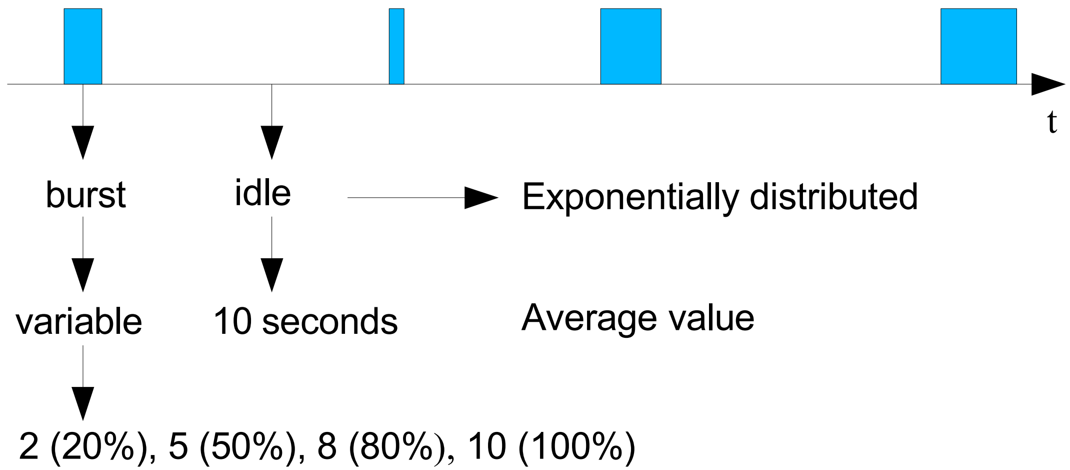


Set-up

- 6 nodes in the overlay (running WB)
- 2 more nodes as routers
- C is the intended Scribe root
- small-size MANET, within the ad-hoc horizon

Human users

- ON/OFF traffic
- Random burst sizes and idle time duration
- Variable average burst size
  - 100% means 1 stroke/sec



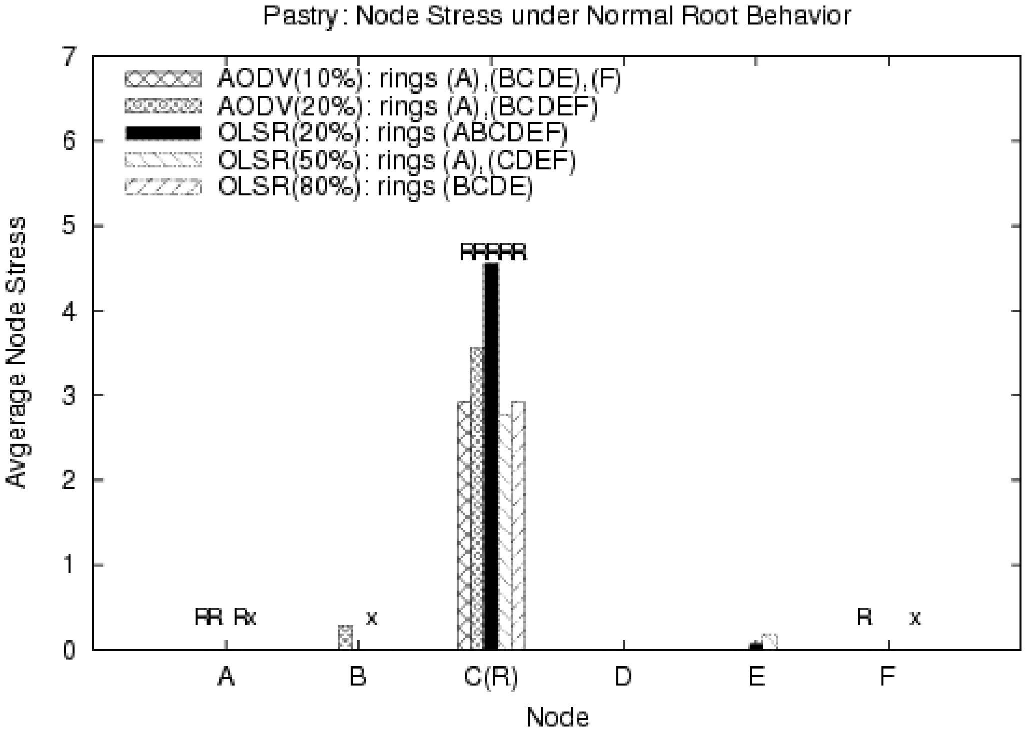
User Satisfaction

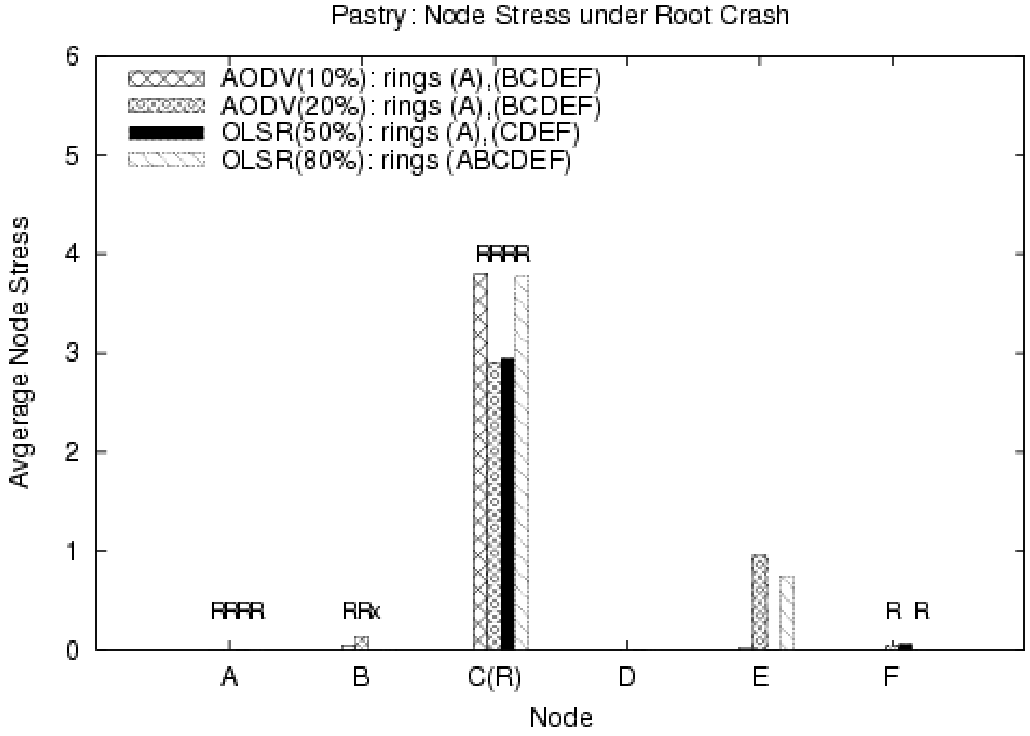
- Message loss rate
  - WRT to the whole bunch of messages sent during the experiment
- Average delay

Multicast Quality

- Average Node Stress
  - number of children for each node
- Number of re-subscription
  - because the previous apparently parent failed

Multicast Quality: Node Stress





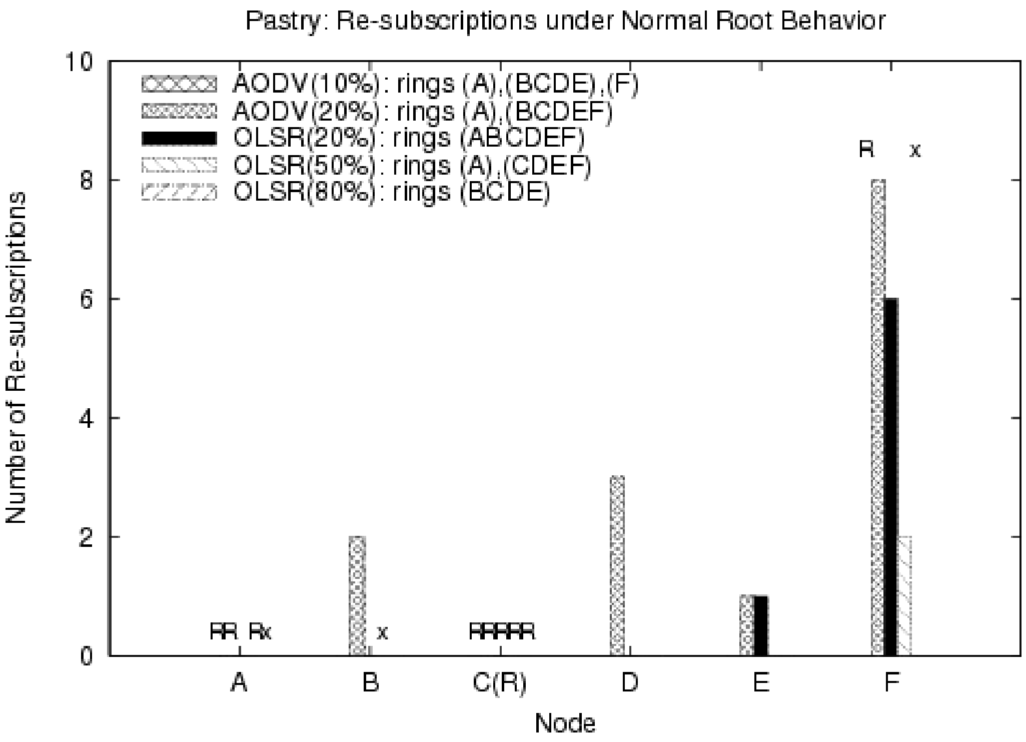
"Best" case

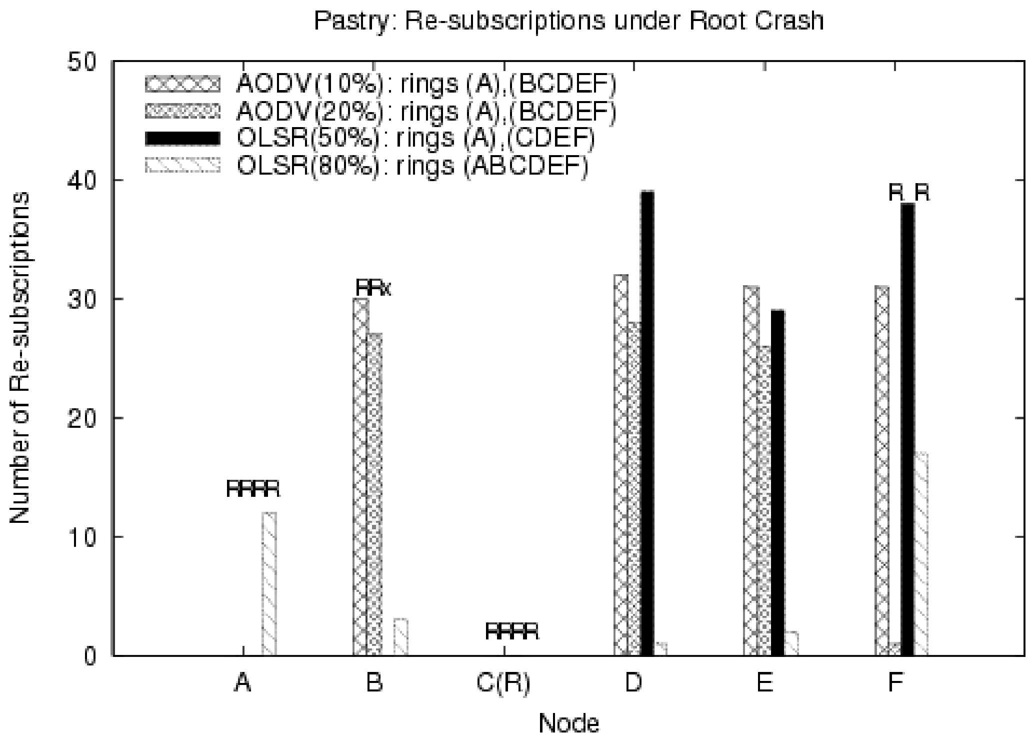
- Very unbalanced tree
- C is one-hop away (at the overlay level) from other nodes
- One of the reasons for the Root Crash
  - C has to generate one distinct message for each child when delivering

Root (C) "crashes"

- Several partitions of the Scribe tree due to congestion !!!
  - node B in AODV cases
  - node A and F in the OLSR 80% case
- Partitions due to:
  - Pastry bootstrap
  - Pastry congestion

Multicast Quality: Re-subscriptions





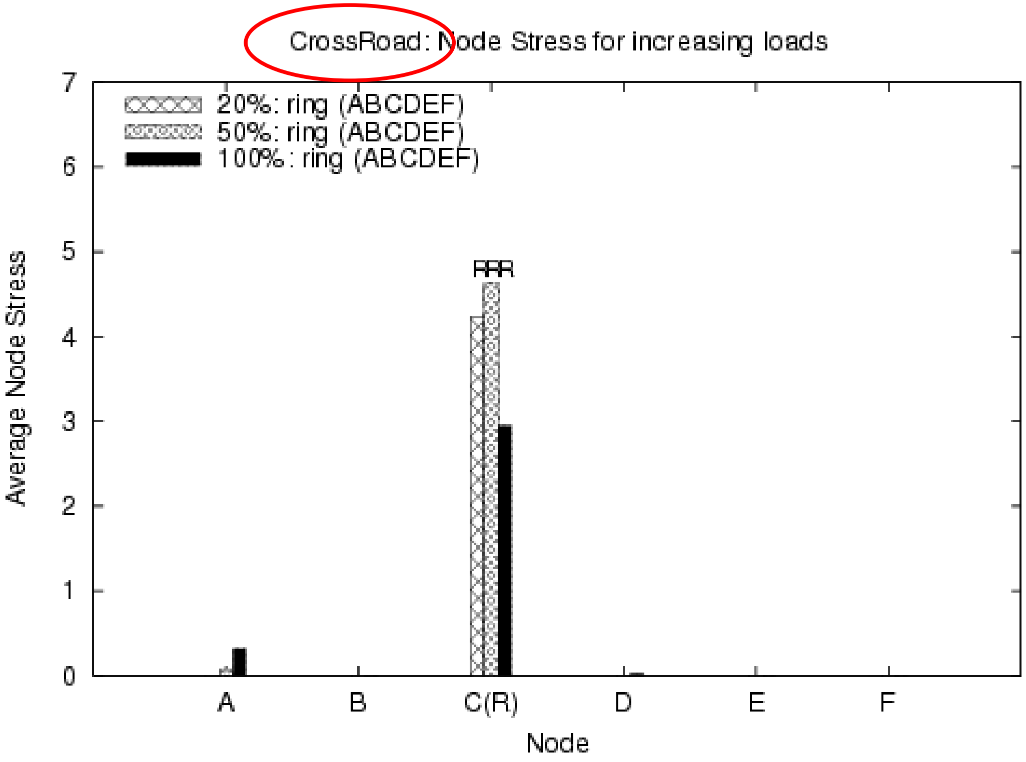
"Best" case

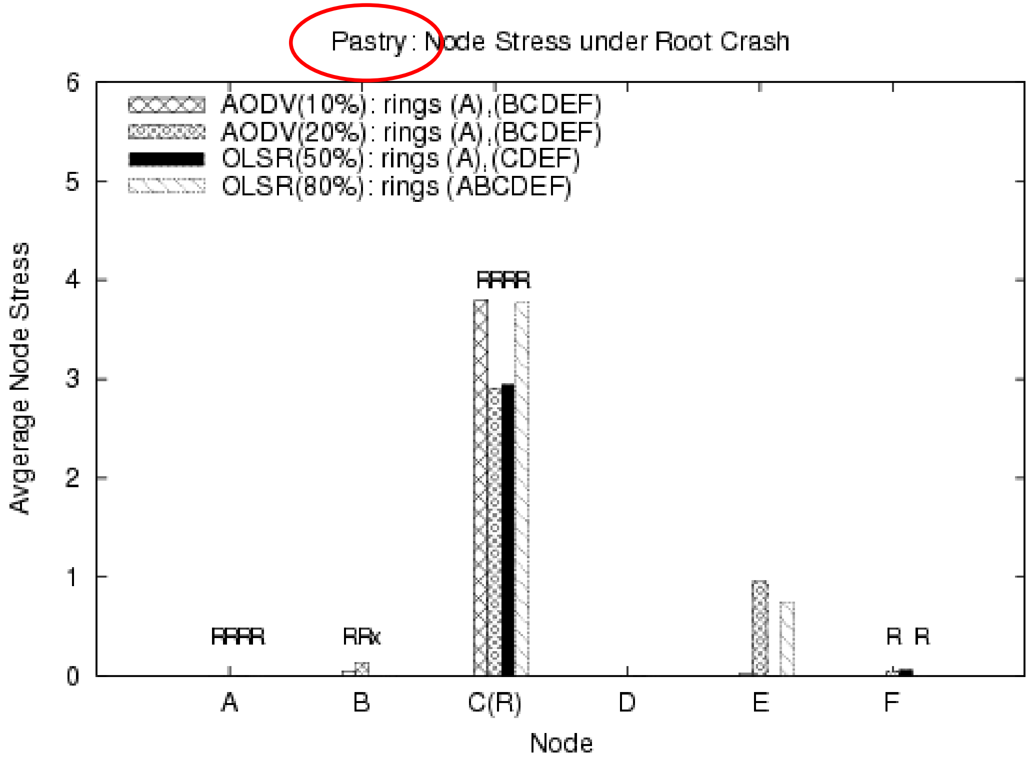
- Not a huge number
- Nodes at the edges of the network (F) tend to loose connectivity

Root (C) "crashes"

- Pastry positive-feedback loop:
  - congestion
  - higher overhead (new subscriptions)
  - more congestion

CrossROAD Multicast Quality

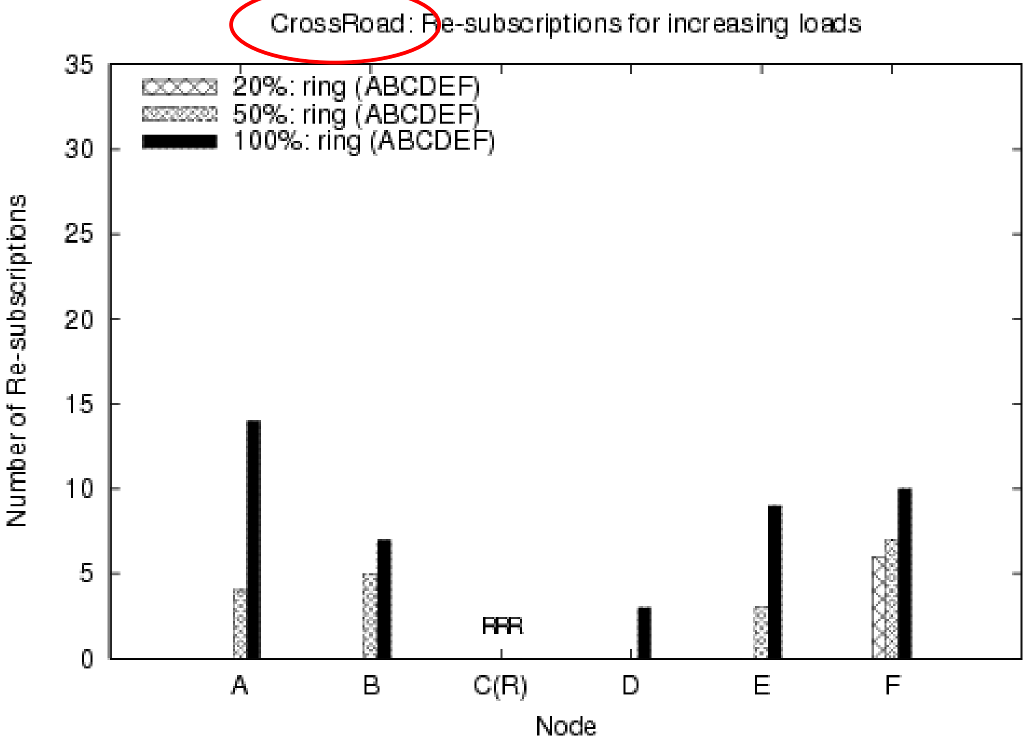


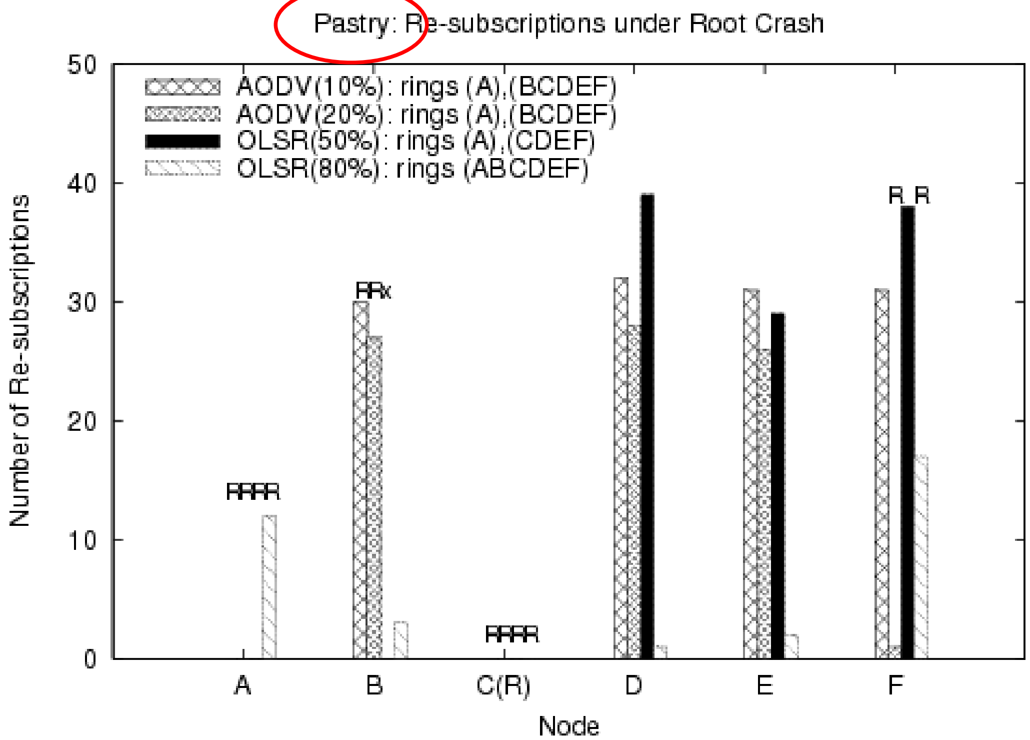


Node stress

- Very unbalanced tree
    - Intrinsic Scribe limitation

CrossROAD Multicast Quality

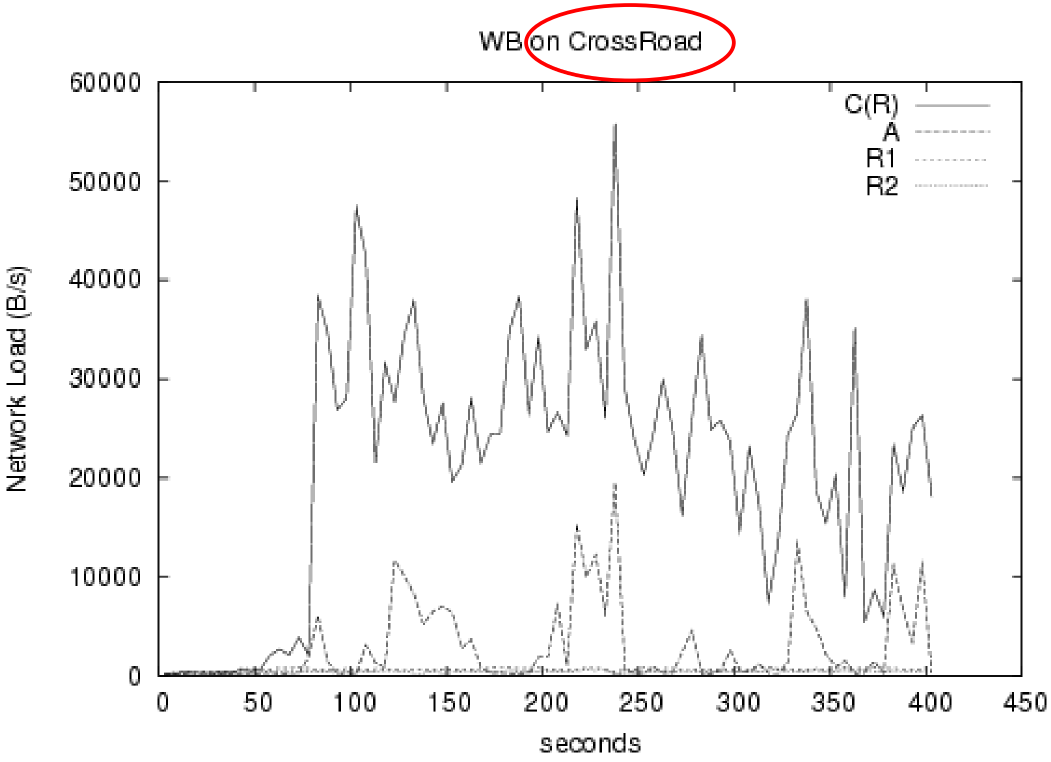


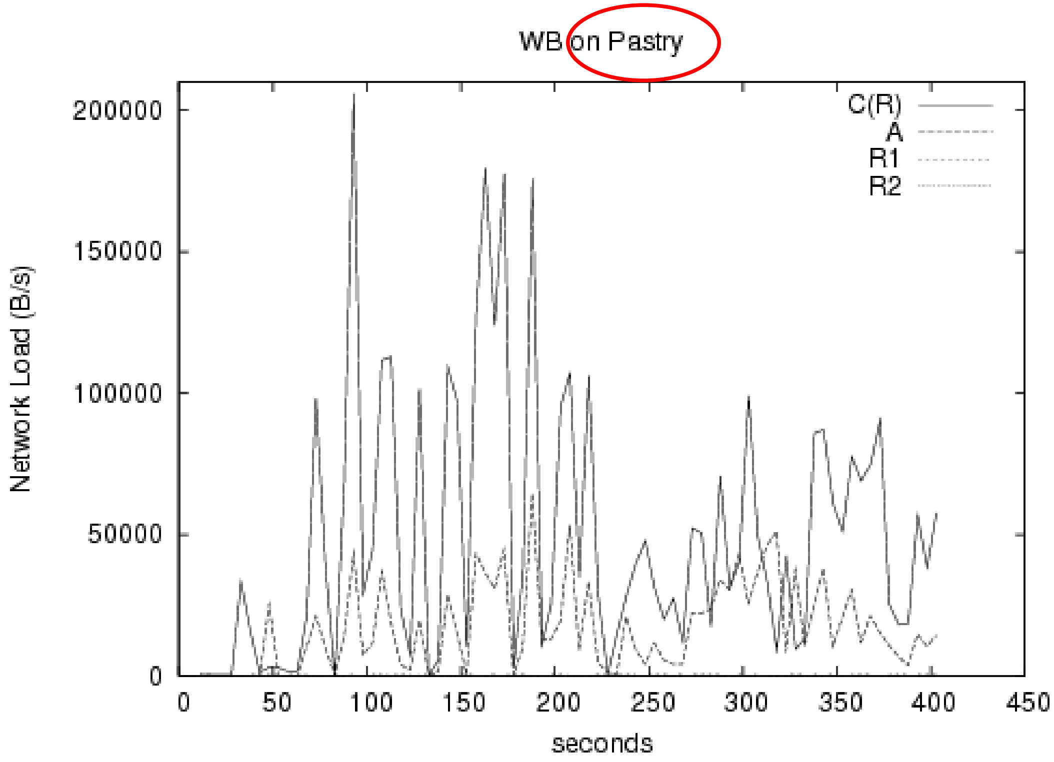


Re-subscriptions

- No bootstrap, no initial partitions
- Only one single tree, also for high traffic load
- Higher stability of the tree

CrossROAD vs. Pastry overheads





Performance

- Load on the Root node is about 60% less when using CrossROAD
- Load on other WB nodes is about 75% less when using CrossROAD
- Often, the additional traffic WRT routing is negligible

Because...

- CrossROAD nicely interacts with a proactive routing protocol
  - No need for bootstrap phase
  - Relies on routing for handling topology changes
  - The aggregated load (overlay + routing) is kept very low

“Take home”

- Current P2P systems need to be tailored to the MANET environment to be actually used

- Pastry is not suitable for so dynamic networks:
  - bootstrap phase and
  - overlay management (positive-feedback loop)

  result in congestion and network partitions (nodes isolation)
  - (even worse performance is expected if adding mobility !!!)

- In moderate-size networks, Scribe makes one node to become a bottleneck
  - tree building algorithm
  - data delivery policy

- CrossROAD fixes the Pastry problems by nicely interacting with a proactive routing protocol
  - No need for a bootstrap phase
  - Relies on routing for handling topology changes
  - The aggregated load (overlay + routing) is kept very low

Future Works

- Optimising Scribe to work on MANETs
- Data delivery
  - Why all messages need to be sent to the root of the tree?
- Tree building
  - Exploit topology information available at the routing level
- What about using unstructured overlay networks?

Thank you !!!