imec

TRINITY COLLEGE DUBLIN

cnit

berlin

nCENTRIC

RUTGERS

UFRJ

SEOUL NATIONAL UNIVERSITY

WiSHFUL

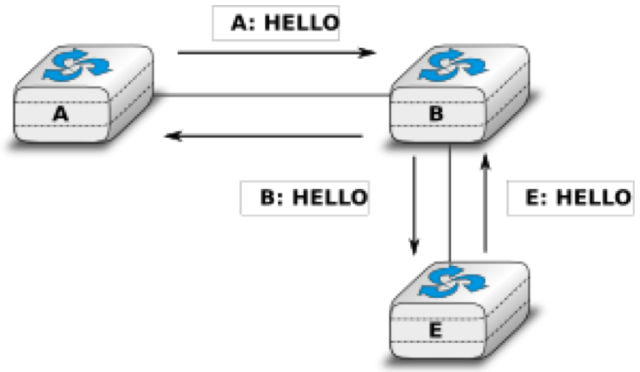
POPROW
POP ROUTING OVER WISHFUL

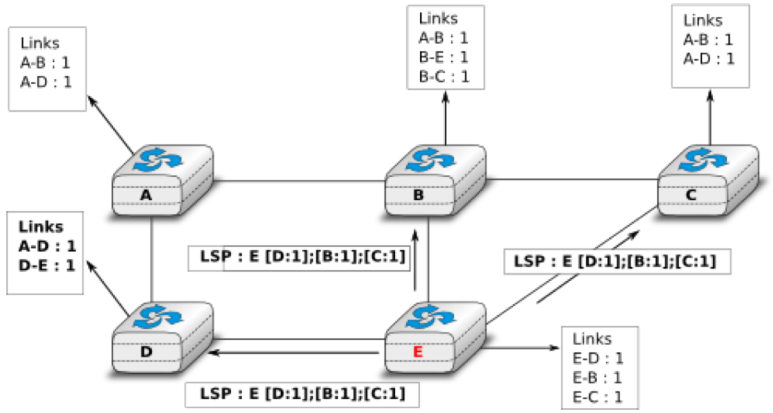
Michele Segata, Nicolò Facchi, Leonardo Maccari,
Renato Lo Cigno – University of Trento

The research leading to these results has received funding from the European Horizon 2020 Programme under grant agreement n° 645274 (WiSHFUL project).

Routing with OLSR

- OLSR: Optimized Link State Routing protocol
- Maintains a table of neighbors
  - Periodically sends HELLO messages every t_H seconds
- Has a complete knowledge of the network topology
  - Broadcasts (and floods) neighbors table through TC messages every t_{TC} seconds
  - Uses Dijkstra on the topology to build the routing table





Routing performance

- Overhead and convergence depend on t_H and t_{TC}
  - Low values: high overhead AND fast convergence
  - High values: low overhead AND slow convergence
- Can we use a different timer for each node?
  - Yes: supported in protocols like OLSR and OSPF
- How do we choose the actual value?
  - Pop-Routing

Pop-Routing

- Pop-Routing [1] computes the optimal t_H and t_{TC} values on a per-node basis
- Best trade-off between overhead and convergence speed is achieved when:

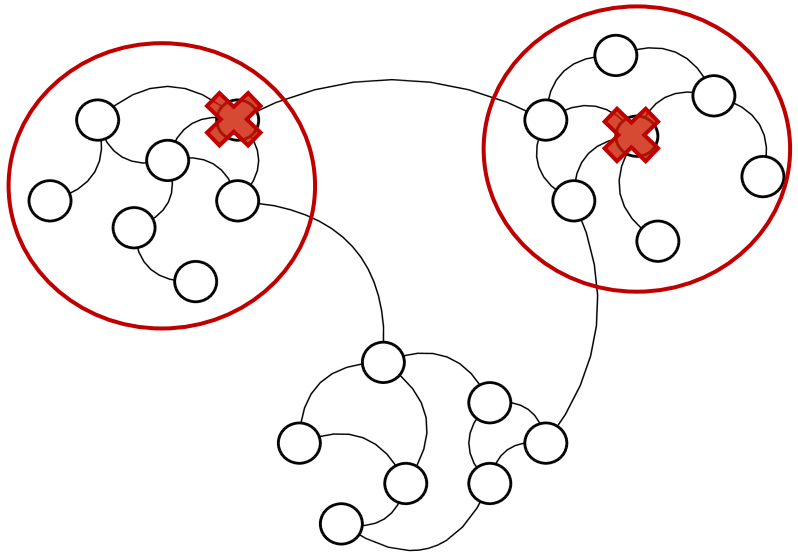
$$t_{\mathrm{H}}(i) \propto \frac{\sqrt{d_i}}{\sqrt{b_i}} \qquad t_{\mathrm{TC}}(i) \propto \frac{\sqrt{E}}{\sqrt{b_i}}$$

- where
  - d_i is the degree of node i
  - E is the number of edges
  - b_i is the betweenness centrality of node i

[1] L. Maccari and R. Lo Cigno, "Pop-Routing: Centrality-Based Tuning of Control Messages for Faster Route Convergence", INFOCOM 2016

Betweenness centrality

- Measures "how much central" a node is in a graph
  - Roughly: b_i is the fraction of shortest paths that pass through node i



- Main takeaways:
  - The higher the centrality the lower the timers
  - Every node can compute its timers independently

Evaluating Pop-Routing

- Evaluation through
  - Simulation ✔
  - Emulation ✔
  - Real network ✘ ← WiSHFUL OC3
    - WiSHFUL UPIs for experiment control
    - Fed4FIRE testbeds
- Target: high automation
  - Nodes setup
  - Setup a desired topology
  - Control the experiment
  - Data logging/retrieval

WiSHFUL Controller

- Experiment control
  - Automatic nodes discovery
  - Implement desired topology
    - .graphml file
    - Any networkx random graph or own graph generator
    - Optional random link multipliers
  - Control nodes through UPIs (see next)
  - Customize scenario (failure strategy)
  - “Wait for convergence” mechanism
  - Repeat Experiments
- Output: time series of broken paths over time

UPIs utilized

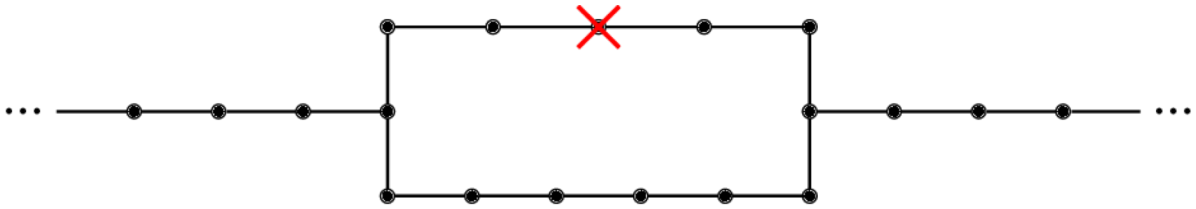
- Automatic nodes discovery/control
  - Not strictly a UPI, but still provided by the framework
  - Very useful controller/agent architecture
- UPI_R
  - set_tx_power()
  - set_modulation_rate()
  - interface_down() (defined and implemented in this project)

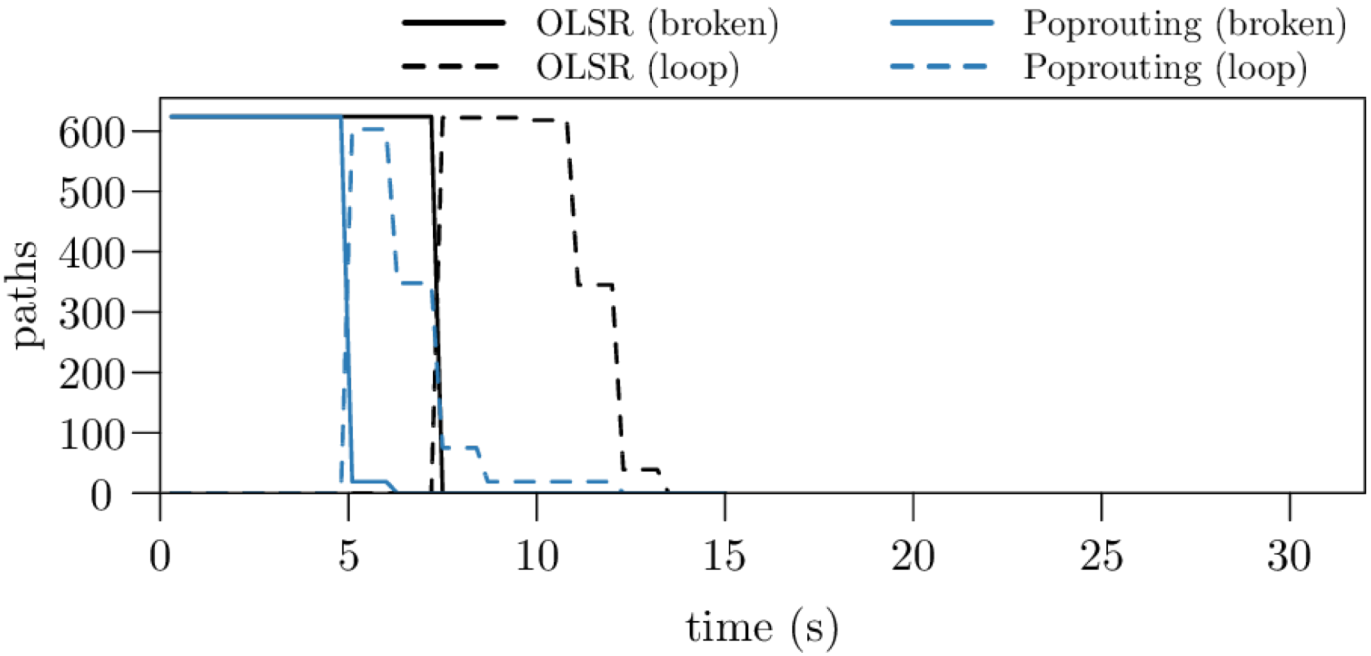
UPIs utilized (cont'd)

- UPI_N
  - start_adhoc()
  - flush_iptables() (defined and implemented in this project, used instead of clear_nf_tables())
  - filter_mac() (defined and implemented in this project)
- UPI_G
  - start_local_control_program()
  - get_hostname() (defined and implemented in this project)
  - run_terminal_command() (defined and implemented in this project)

Example outcome

- Setup a line topology with a branch using 42 nodes on the w.iLab.t testbed





WiSHFUL Feedback

- jFed
  - Very useful when trying to setup the first test experiments with a few nodes
  - Requires a lot of manual operation/intervention for setting up large experiments or in case of failures
  - Finally switch to OMNI for higher automation (thanks to TU Berlin for hints and help)

WiSHFUL Feedback (cont'd)

- WiSHFUL framework
  - Very clear first time setup (account, certificate, jFed) +
  - Nodes not "WiSHFUL-ready" (better) – +
  - Extremely easy to extend the framework +
    - Missing UPIs not a limitation
  - Would be great to have support for OpenWRT –
  - Non-uniform UPIs implementation –
    - Ex. 1: Python library for iptables manipulation
    - Ex. 2: raw "sudo" commands for interface manipulation
- Overall: very positive experience

Conclusion

- Overall outcome
  - The experiments led to 2 publications
    - M. Segata, N. Facchi, L. Maccari, G. Gemmi, and R. Lo Cigno, "Centrality-based Route Recovery in Wireless Mesh Networks", in IEEE ICC 2018
    - M. Segata, N. Facchi, L. Maccari, and R. Lo Cigno, "RoRoute: Tools to Experiment with Routing Protocols in WMNs"
  - Setting up the same kind of experiments without WiSHFUL would have been extremely cumbersome

Future with WiSHFUL

- Future is now!
  - We continue to use WiSHFUL for testing our P2P video streaming platform

imec

TRINITY COLLEGE DUBLIN

cnit

berlin

nCENTRIC

RUTGERS

UFRJ

SEOUL NATIONAL UNIVERSITY

WiSHFUL

THANK YOU!

The research leading to these results has received funding from the European Horizon 2020 Programme under grant agreement n° 645274 (WiSHFUL project).