MANETKit: Supporting the Dynamic Deployment and Reconfiguration of Ad-Hoc Routing Protocols

ACM/IFIP/USENIX 10th International Middleware Conference

2nd December 2009

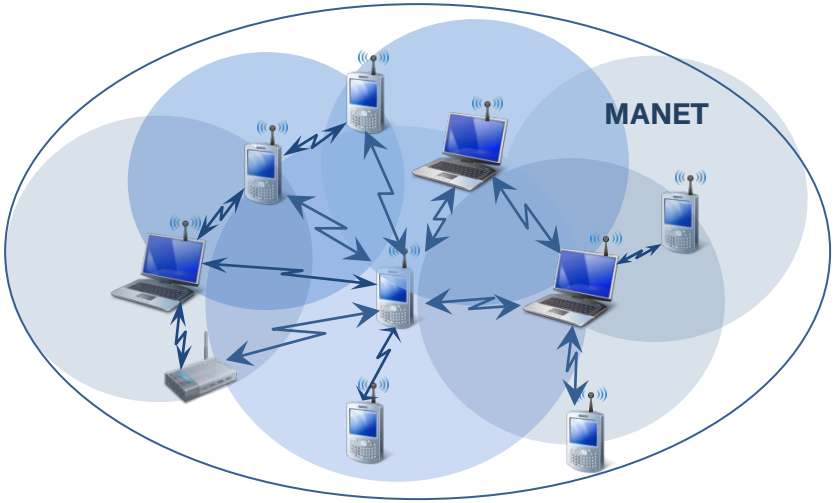
Rajiv Ramdhany
r.ramdhany@comp.lancs.ac.uk

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

Mobile Ad Hoc Networks

- MANETs are self-organising and rapidly-deployable networks
  - Bandwidth-constrained, variable-capacity and possibly asymmetric links
  - Highly heterogeneous
  - Network topology changes dynamically and unpredictably



dynamicity
heterogeneity

{size, density, mobility, churn, node_degree, link_capacity [$Node_x$, $Node_y$], NodeCapability[][], traff

{$size_{A,T1}$, $density_{A,T1}$, $mobility_{A,T1}$, $churn_{A,T1}$, $node\_degree_{A,T1}$, $link\_capacity_{A,T1}$ [$Node_x$, $Node_y$], $NodeCapability_{A,T1}$[][], $apps_{A,T1}$}

≠

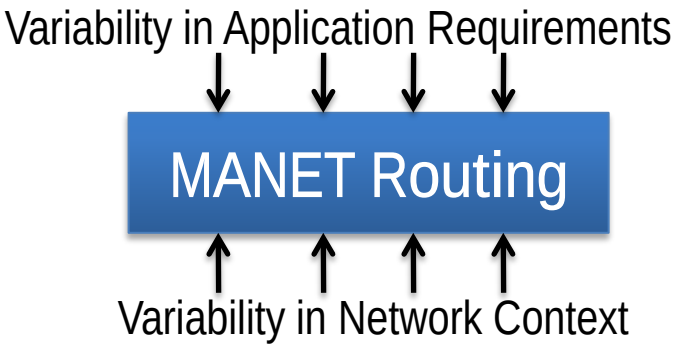
{$size_{A,T2}$, $density_{A,T2}$, $mobility_{A,T2}$, $churn_{A,T2}$, $node\_degree_{A,T2}$, $link\_capacity_{A,T2}$ [$Node_X$, $Node_Y$], $NodeCapability_A$[][], $apps_{A,T2}$}

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

Mobile Ad Hoc Networks

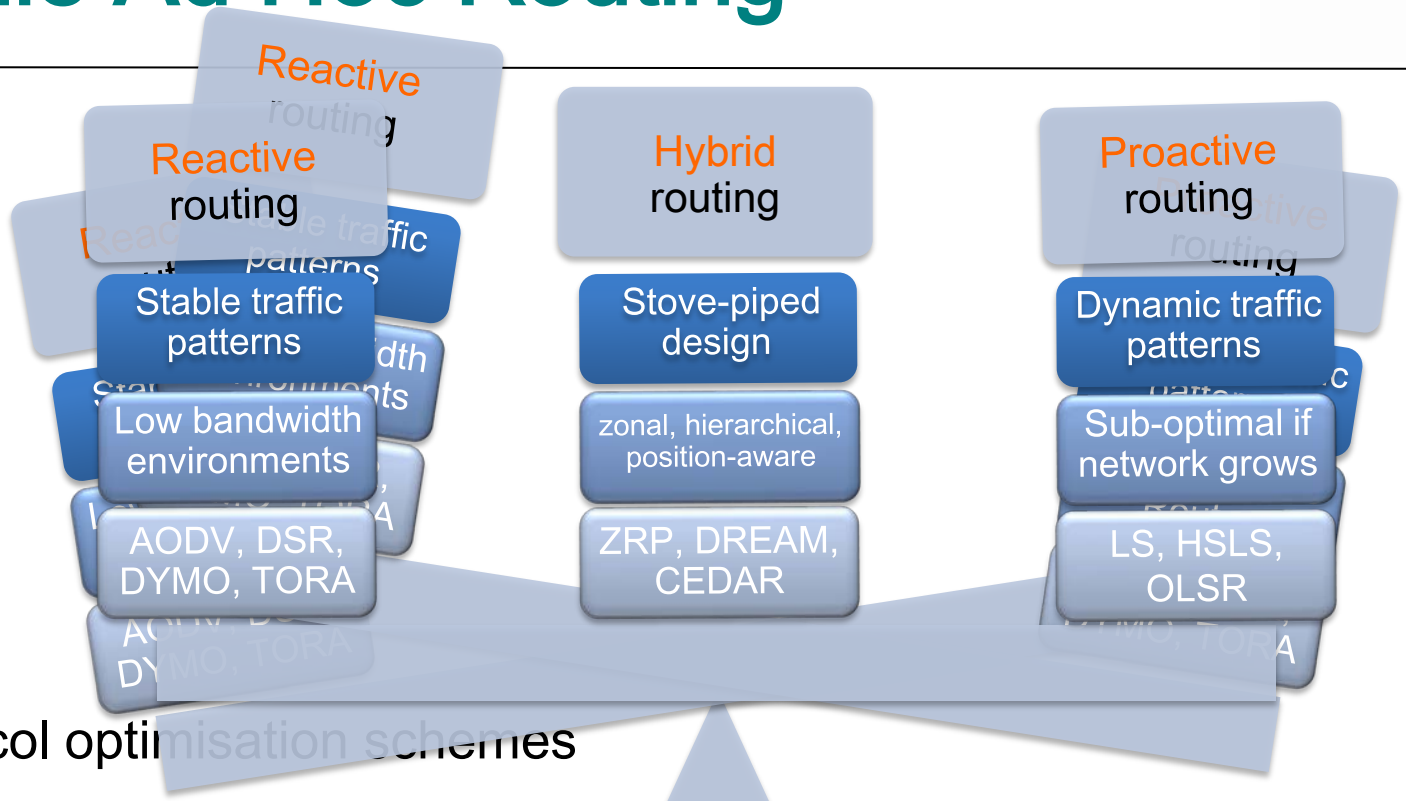
- MANETs are self-organising and rapidly-deployable networks
  - Bandwidth-constrained, variable-capacity and possibly asymmetric links
  - Highly heterogeneous
  - Network topology changes dynamically and unpredictably
  - Variability



- Hard to design generically-applicable routing protocols

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

Mobile Ad Hoc Routing



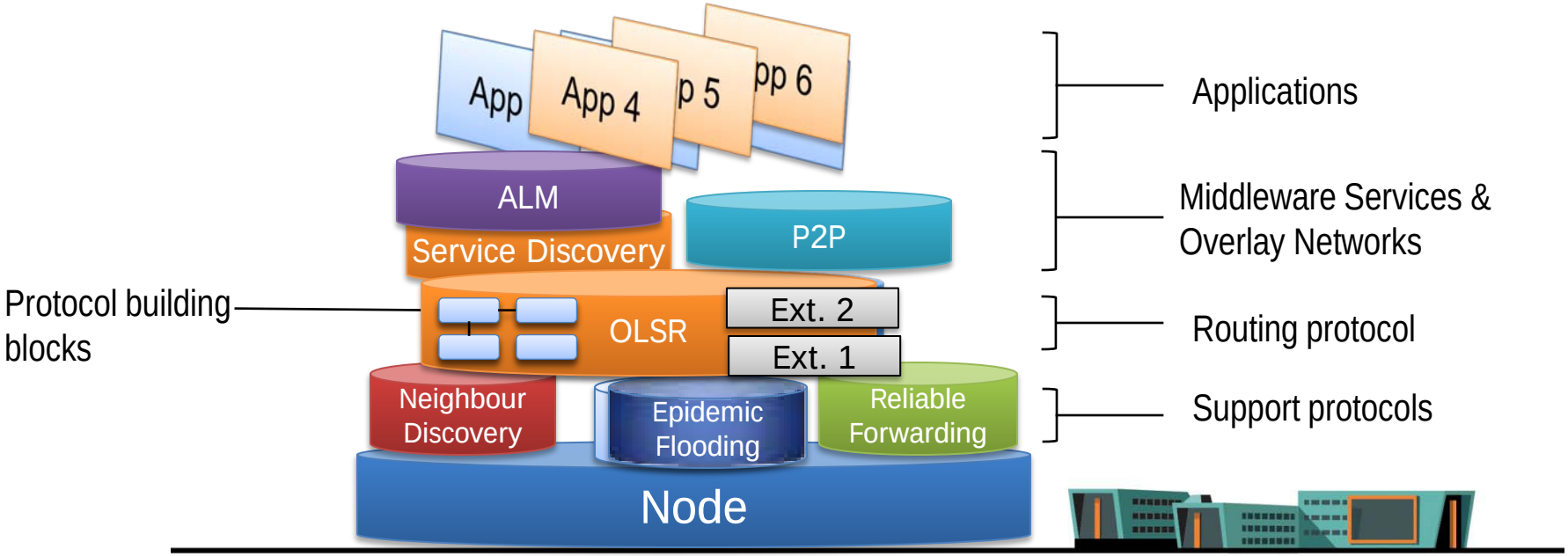
Protocol optimisation schemes

- Efficiency (routing overhead) - Route caching, Path accumulation, fish-eye routing, efficient | limited | epidemic flooding
- Efficiency (energy) - power-aware routing
- Resilience, load balancing, QoS differentiation: Multi-path routing, Pre-emptive routing

- Proliferation of ad hoc routing protocols:
  - DSR, AODV, DYMO, OLSR, HSLS, ZRP, TORA, CEDAR, DREAM, LAR, GPSR, etc
- A *one-size-fits-all* ad hoc routing algorithm is a chimera

Frameworks for Ad Hoc Routing

- Support dynamic deployment of ad hoc routing protocols serially
- Shorten protocol development cycle
- Support fine-grained dynamic reconfiguration

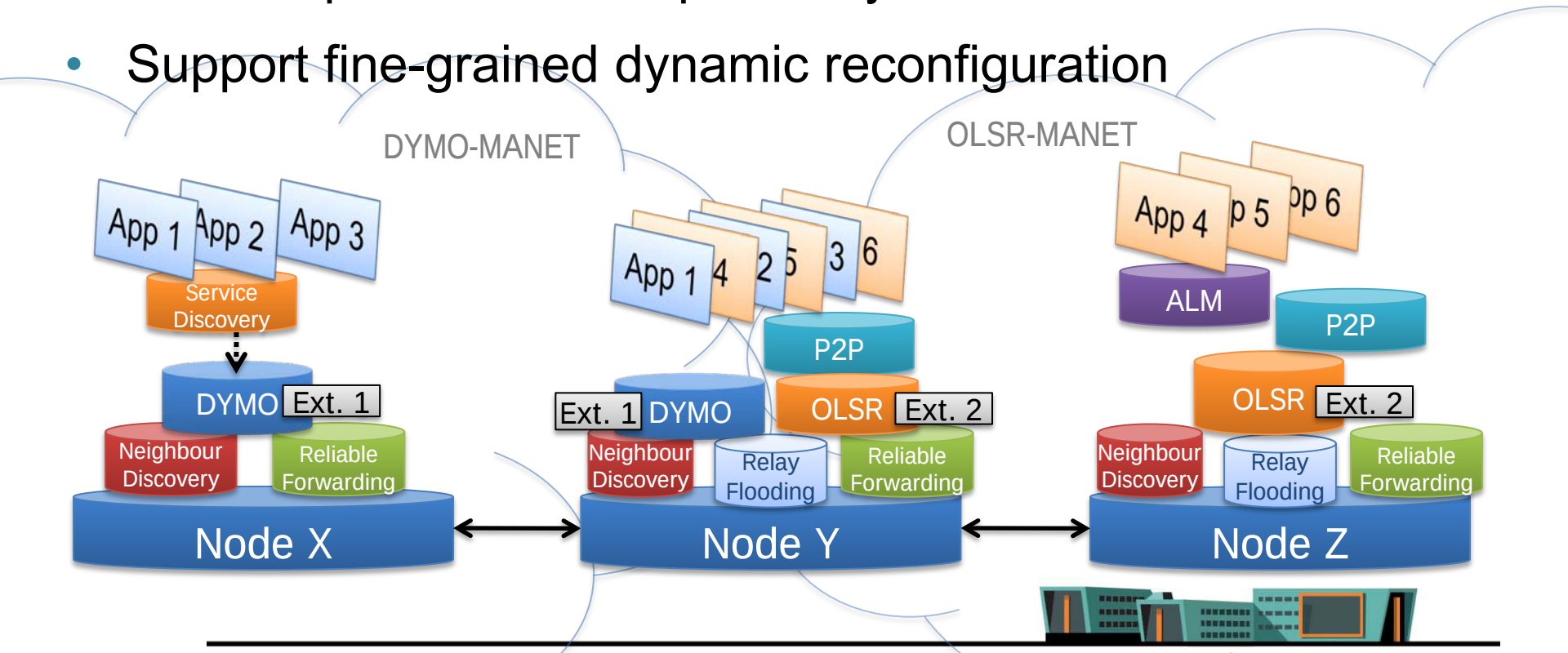


InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

Frameworks for Ad Hoc Routing

- Support dynamic deployment of ad hoc routing protocols serially *and* simultaneously
- Shorten protocol development cycle
- Support fine-grained dynamic reconfiguration

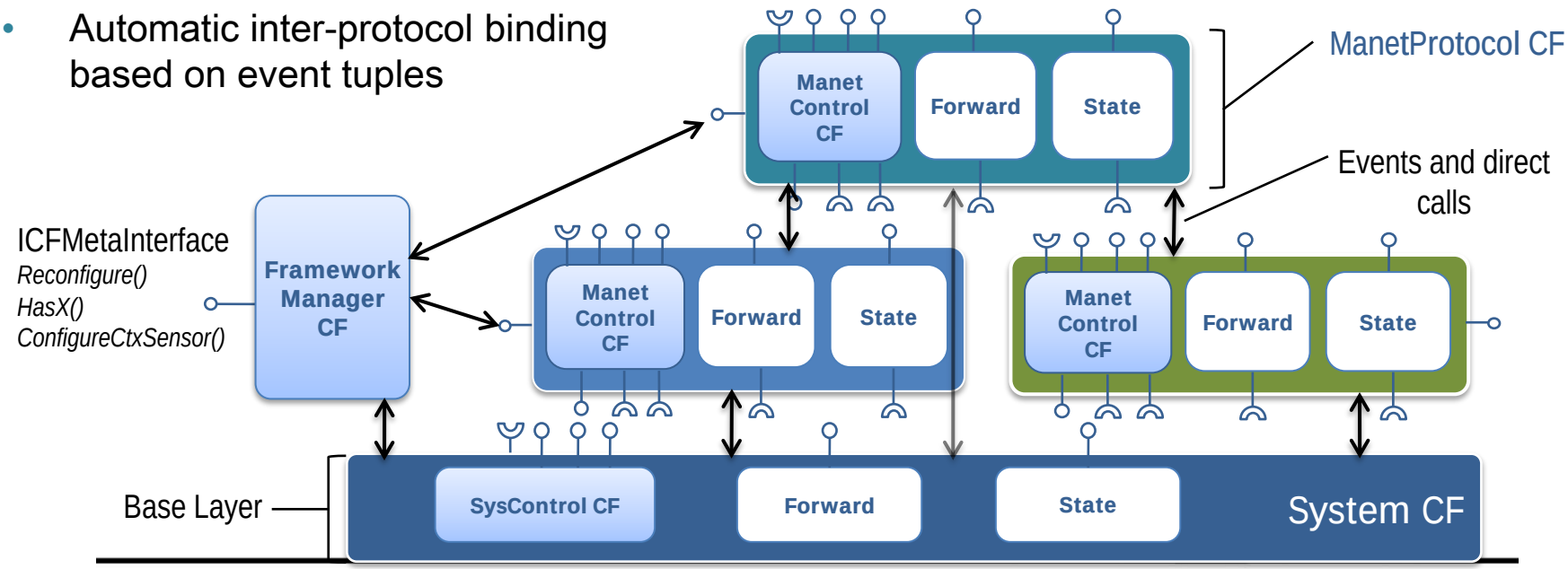


InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit: Protocol Composition (1/2)

- Underpinned by run-time deployable software components (OpenCom) and Component Frameworks (CFs)
- Recursive use of ManetProtocolCF
- Inter-Protocol interaction: use of events and standard interfaces
  - Each ManetProtocolCFdefines a tuple: <required-events, provided-events>
  - Automatic inter-protocol binding based on event tuples

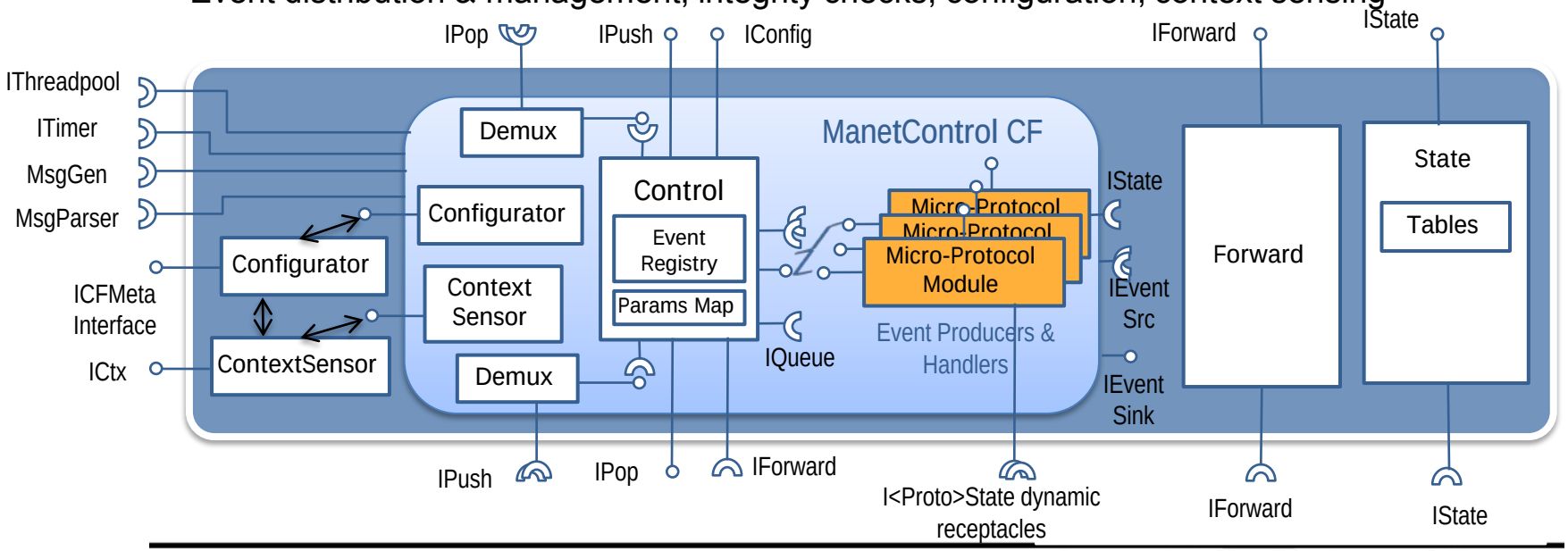


InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit: Protocol Composition (2/2)

- The ManetProtocol CF
  - Control-Forward-State (CFS) pattern
    - Applies to ad hoc routing and support protocols
  - Composed of generic and user-provided components
    - Micro-protocol modules to be implemented by user
  - Comes by default with generic machinery
    - Event distribution & management, integrity checks, configuration, context sensing

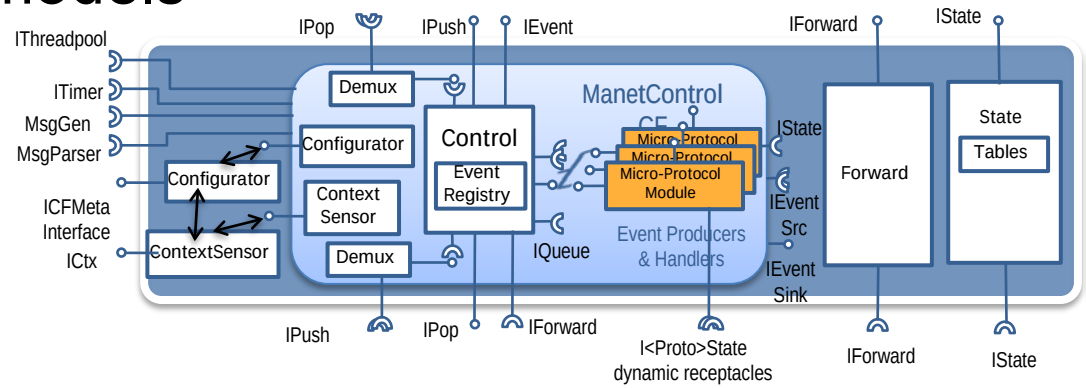


InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit: Other Key Features

- Alternative concurrency models
  - Single-threaded
  - Thread-per-message or
  - Thread-per-ManetProtocol

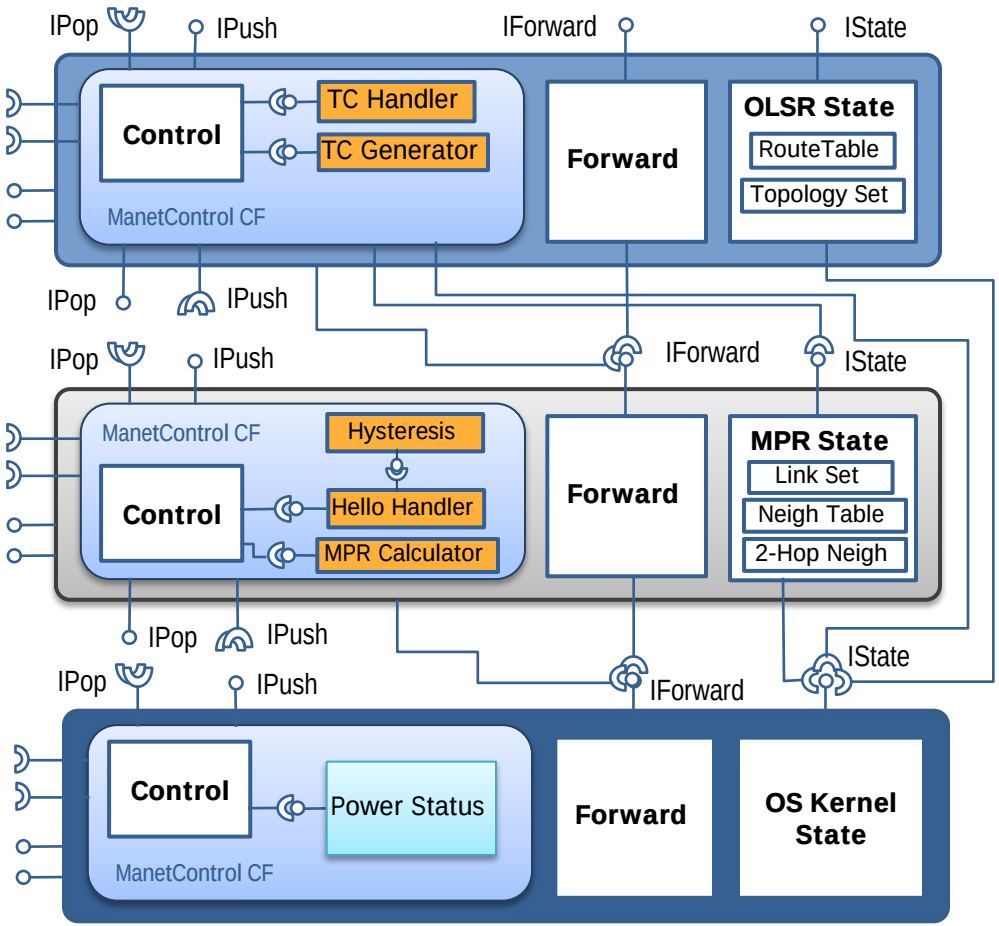


- Dynamic Reconfiguration
  - Context monitoring (link quality, system load ...)
    - Context components deployed and context sensors re-configured
  - Reconfiguration Enactment (Local)
    - Fine-grained: use OpenCom reflective capabilities and CFs integrity checking facilities
    - Coarse-grained: update event tuples of ManetProtocol instances

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit-OLSR



OLSR ManetProtocol CF

P_{events} = {TC_OUT}

R_{events} = {TC_IN, NEIGH_CHANGE, MPR_CHANGE}

MPR ManetProtocol CF

P_{events} = {HELLO_OUT, MPR_CHANGE }

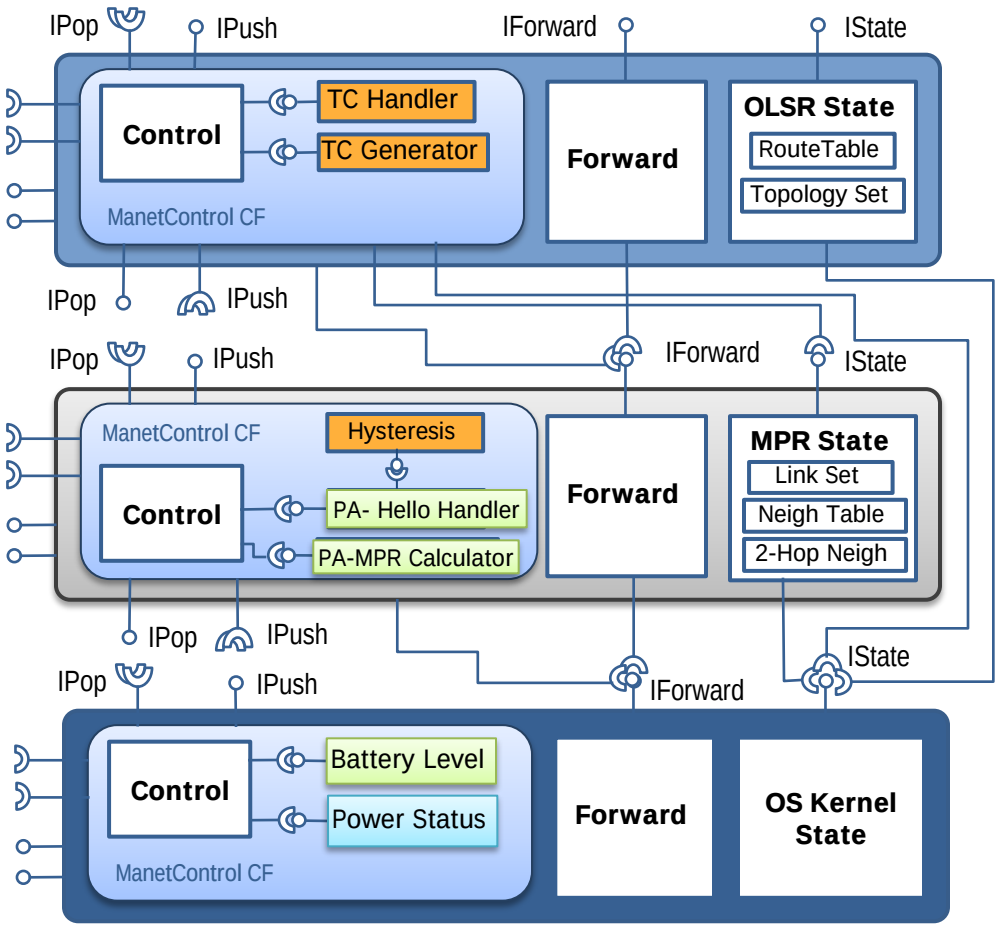
R_{events} = {HELLO_IN, POWER_STATUS}

System CF

P_{events} = {HELLO_IN, TC_IN, POWER_STATUS }

R_{events} = {HELLO_OUT, TC_OUT}

MANETKit-OLSR *power-aware*



OLSR ManetProtocol CF

P_{events} = {TC_OUT}

R_{events} = {TC_IN, NEIGH_CHANGE, MPR_CHANGE}

MPR ManetProtocol CF

P_{events} = {HELLO_OUT, MPR_CHANGE }

R_{events} = {HELLO_IN, POWER_STATUS, BATTERY_LEVEL}

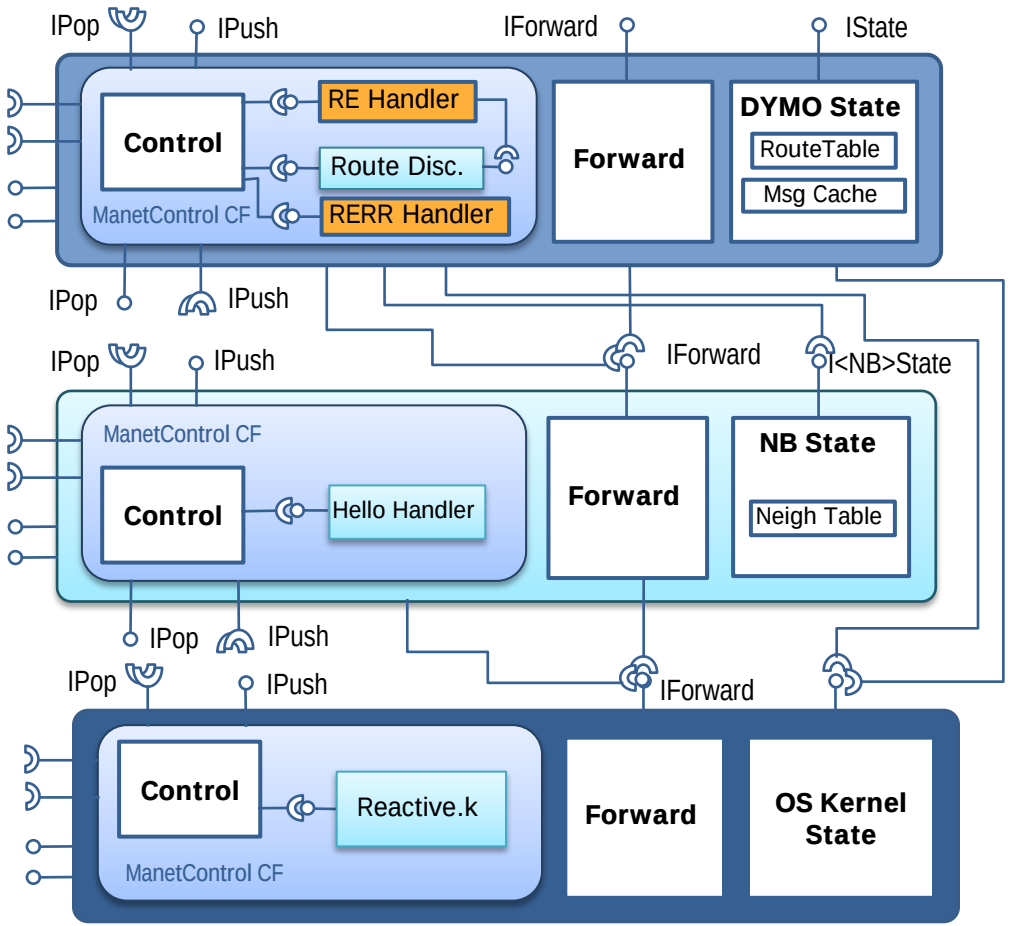
System CF

P_{events} = {HELLO_IN, TC_IN, POWER_STATUS, BATTERY_LEVEL }

R_{events} = {HELLO_OUT, TC_OUT}

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

MANETKit-DYMO



DYMO ManetProtocol CF

P_{events} = {RE_OUT, RERR_OUT, RTE_FOUND}

R_{events} = {RE_IN, RERR_IN, NEIGH_CHANGE, NO_ROUTE, RTE_UPDATE, SND_RTE_ERR}

Neighbour Discovery ManetProtocol CF

P_{events} = {HELLO_OUT, NEIGH_CHANGE }

R_{events} = {HELLO_IN}

System CF

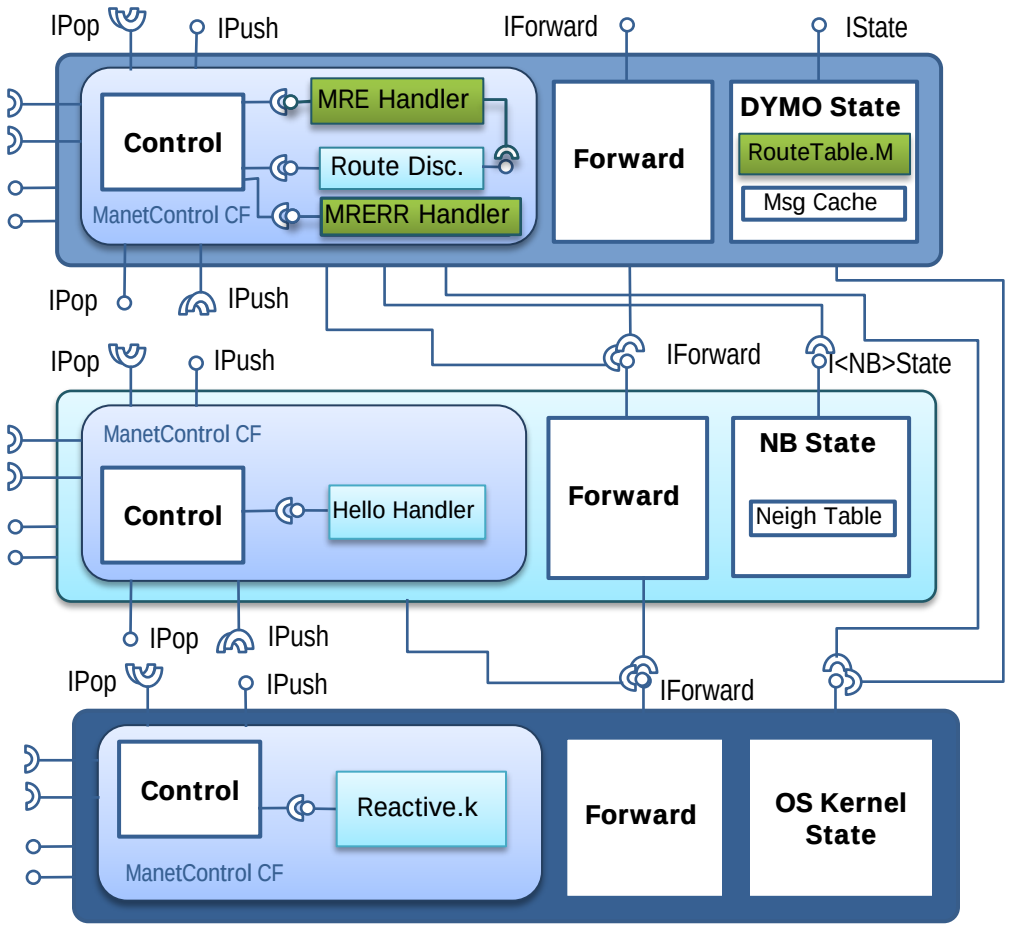
P_{events} = {NO_ROUTE, RTE_UPDATE, SND_RTE_ERR, RE_IN, RERR_IN, HELLO_IN }

R_{events} = {HELLO_OUT, RE_OUT, RTE_FOUND}

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit-DYMO *multi-path*



DYMO ManetProtocol CF

P_{events} = {RE_OUT, RERR_OUT, RTE_FOUND}

R_{events} = {RE_IN, RERR_IN, NEIGH_CHANGE, NO_ROUTE, RTE_UPDATE, SND_RTE_ERR}

Neighbour Discovery ManetProtocol CF

P_{events} = {HELLO_OUT, NEIGH_CHANGE }

R_{events} = {HELLO_IN}

System CF

P_{events} = {NO_ROUTE, RTE_UPDATE, SND_RTE_ERR, RE_IN, RERR_IN, HELLO_IN }

R_{events} = {HELLO_OUT, RE_OUT, RERR_OUT, RTE_FOUND}

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

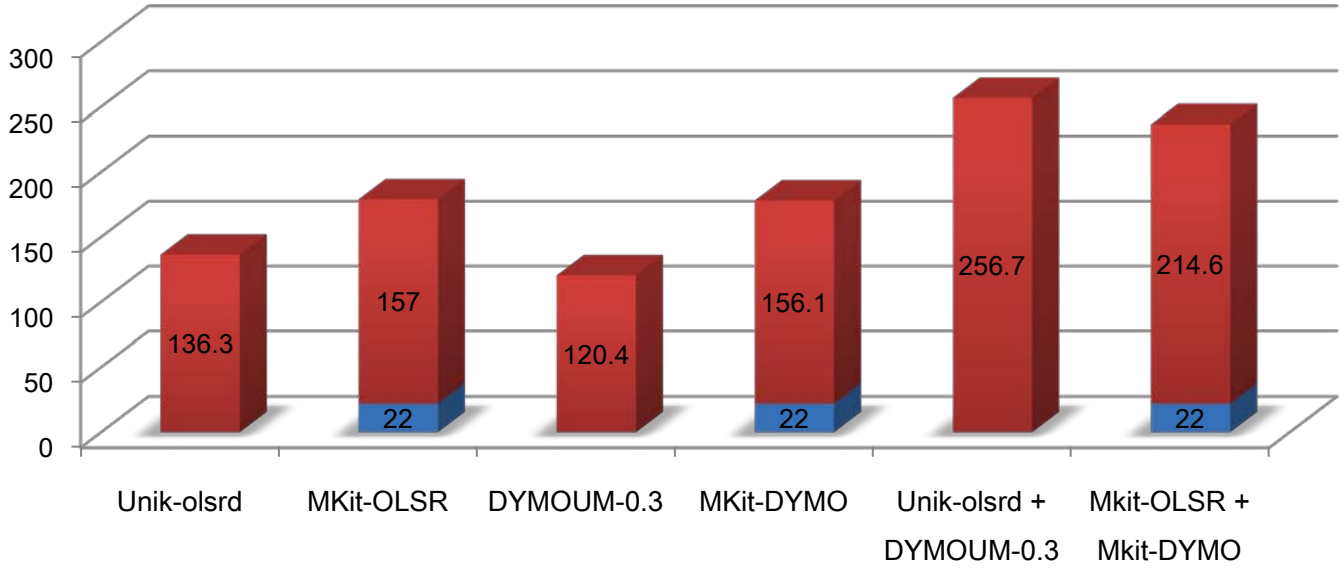
Evaluation(1/4)

- Testbed consists of an 802.11b/g network of 5 Linux computers
  - MANETKit implemented in C
  - Used Unik-olsrd and DYMOUM v0.3 as comparators
  - Single-threaded model, identical protocol parameters
- Measurements
  - Resource overhead
  - Protocol performance
  - Time to develop ad hoc routing protocols

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

Evaluation(2/4)

- Resource Overhead
  - Memory footprint (KB)

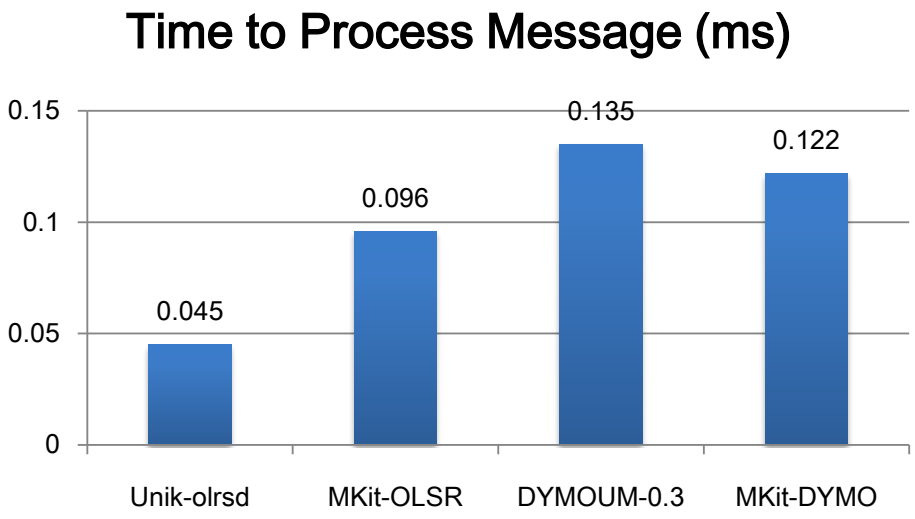


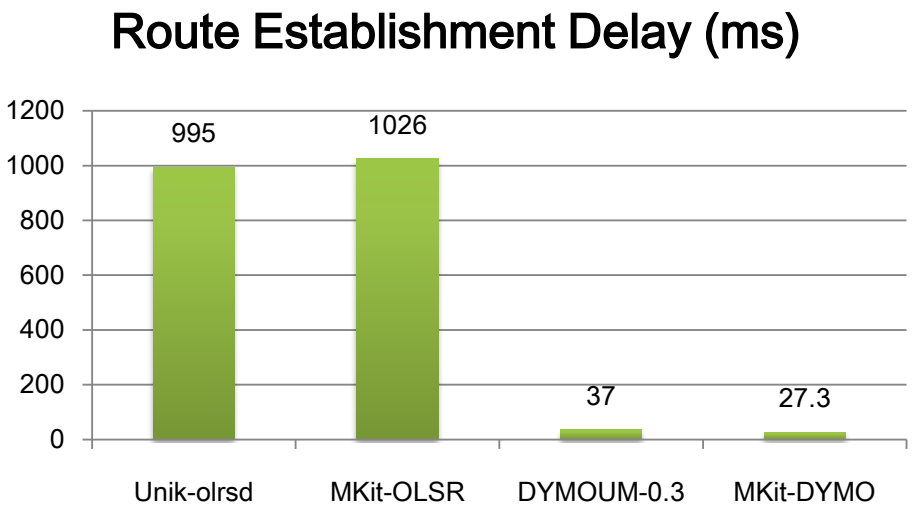
- MANETKit-OLSR incurs 31% memory overhead
- MANETKit-DYMO incurs 48% memory overhead
- Deploying MANETKit-OLSR + MANETKit-DYMO is 8% smaller than Unik-olsrd + DYMOUM

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

Evaluation(3/4)

- Protocol Performance
  - Time to Process Message
    - Very small difference in time to process a protocol message (< 0.2 ms)
  - Route Establishment Delay
    - MANETKit-OLSR is 3% slower than Unik-olsrd
    - MANETKit-DYMO is 35% faster than DYMOUM-0.3
    - On-the-par performance attained



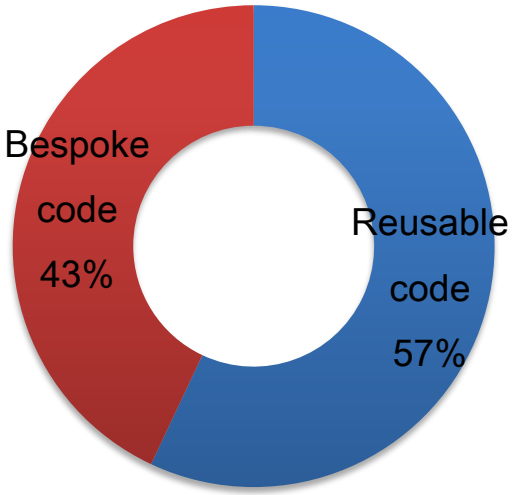


InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk
Computing Department

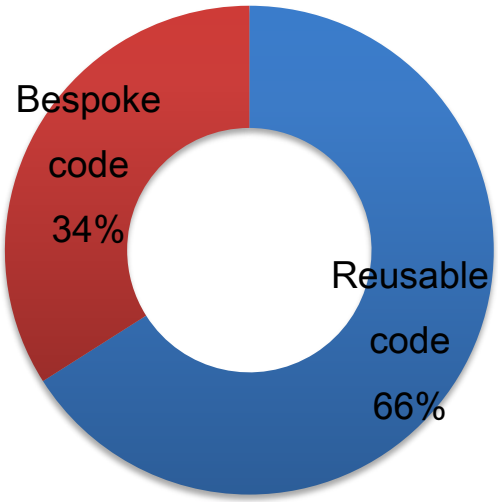
Evaluation(4/4)

- Time to develop and port Protocols
  - Degree of code reuse

MKit-OLSR



MKit-DYMO



InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

Future Work

- Further evaluation and experimentation
  - Hybrid OLSR and DYMO deployment issues
  - Additional protocol implementations
  - Further investigation of reconfiguration strategies to hybridise protocols and experiment with optimisation schemes
- Integrate MANETKit in a distributed reconfiguration environment
  - Incorporate policy-driven decision making
  - Concensus-based distributed reconfiguration

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

Conclusion

- Framework approach to
  - Compose reactive, proactive and hybrid ad hoc routing protocols
  - Support dynamic deployment of the protocols serially *and* simultaneously
  - Dynamically reconfigure protocols at a fine-grained level to produce protocol variants
- Overhead/flexibility tradeoff in MANETKit's favour

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

Questions?

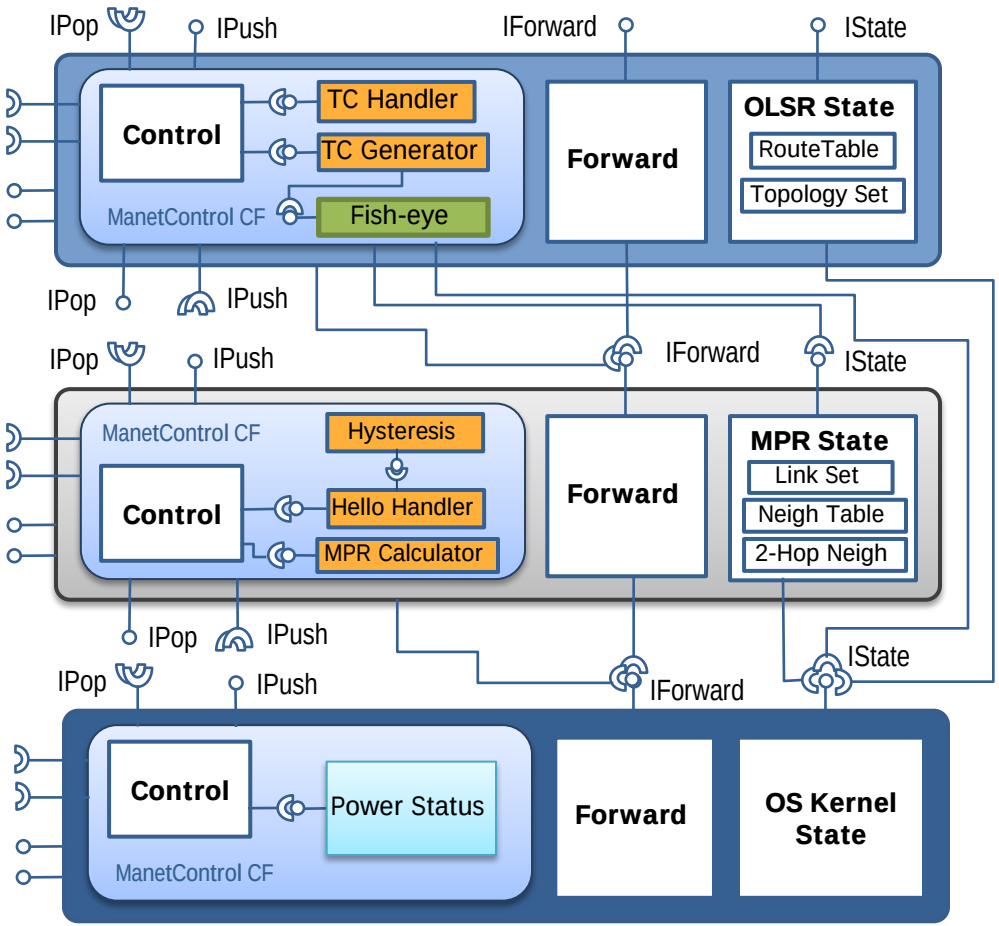
MANETKit's List of Generic Components:
RouteDiscovery, Configurator, Counter, ContextSensor, DecisionEngine, EventBus, EventClassifier, EventRegistry, FromDevice, FromHost, Queue, FrontDropQueue, Jitter, MsgCache, NotifierQueue, PacketBBGen, PacketBBParser, RadioSim, Reactive.k, Reactive.u, RoutingTable, RoutingTable.m, RouteDiscovery, Scheduler, SocketParser, Threadpool, iThreadpool, ToDevice, ToHost, FrameworkManager, Control, PolicyDB, ProtocolState

Thank you for listening

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit-OLSR *fish-eye*



OLSR ManetProtocol CF

P_{events} = {TC_OUT}

R_{events} = {TC_IN, NEIGH_CHANGE, MPR_CHANGE}

MPR ManetProtocol CF

P_{events} = {HELLO_OUT, MPR_CHANGE }

R_{events} = {HELLO_IN, POWER_STATUS}

System CF

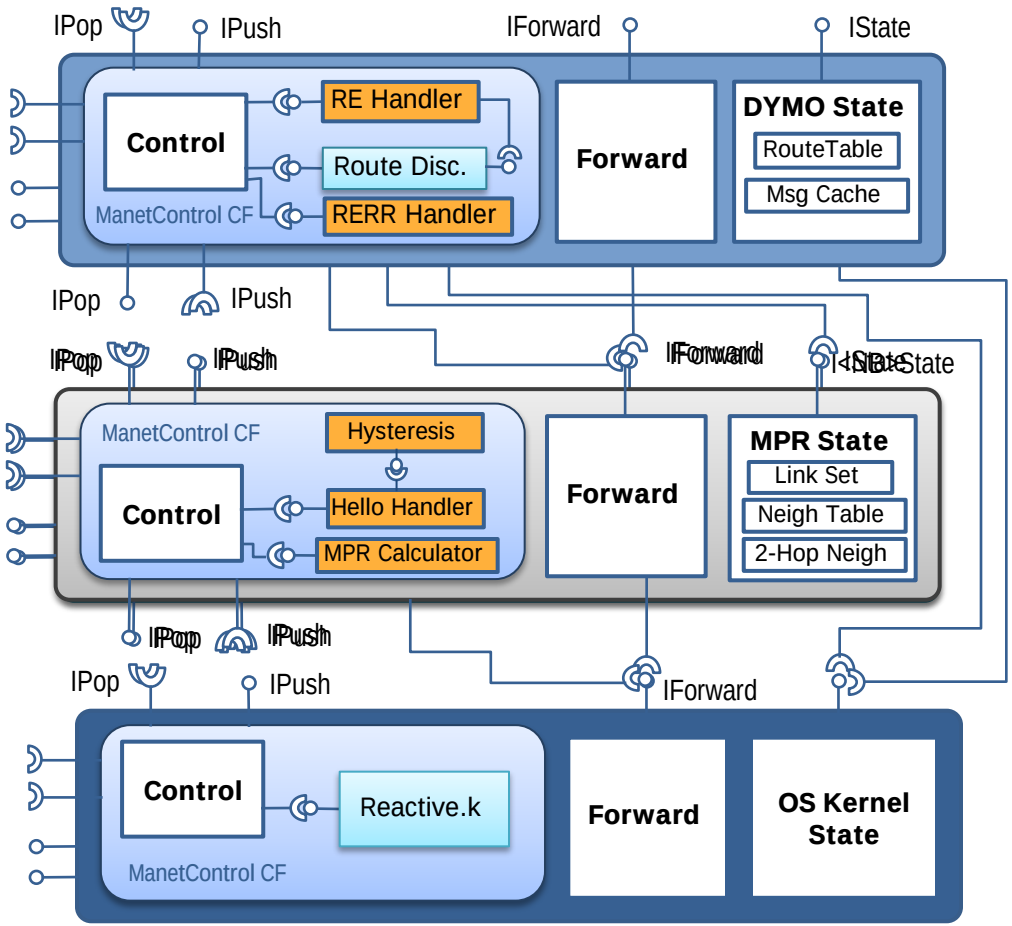
P_{events} = {HELLO_IN, TC_IN, POWER_STATUS }

R_{events} = {HELLO_OUT, TC_OUT}

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk

InfoLab21
Computing Department

MANETKit-DYMO *relay-flood*



DYMO ManetProtocol CF

P_{events} = {RE_OUT, RERR_OUT, RTE_FOUND}

R_{events} = {RE_IN, RERR_IN, NEIGH_CHANGE, NO_ROUTE, RTE_UPDATE, SND_RTE_ERR}

Neighbour Discovery ManetProtocol CF

P_{events} = {HELLO_OUT, NEIGH_CHANGE}

R_{events} = {HELLO_IN}

System CF

P_{events} = {NO_ROUTE, RTE_UPDATE, SND_RTE_ERR, RE_IN, RERR_IN, HELLO_IN }

R_{events} = {HELLO_OUT, RE_OUT, RTE_FOUND}

InfoLab21 | Lancaster University | Lancaster | LA1 4WA | UK
www.infolab21.lancs.ac.uk