Towards a Rigorous Analysis of AODVv2 (DYMO)

Peter Höfner, Sarah Edenhofer

W-RiPE
Austin, Texas
October 30, 2012

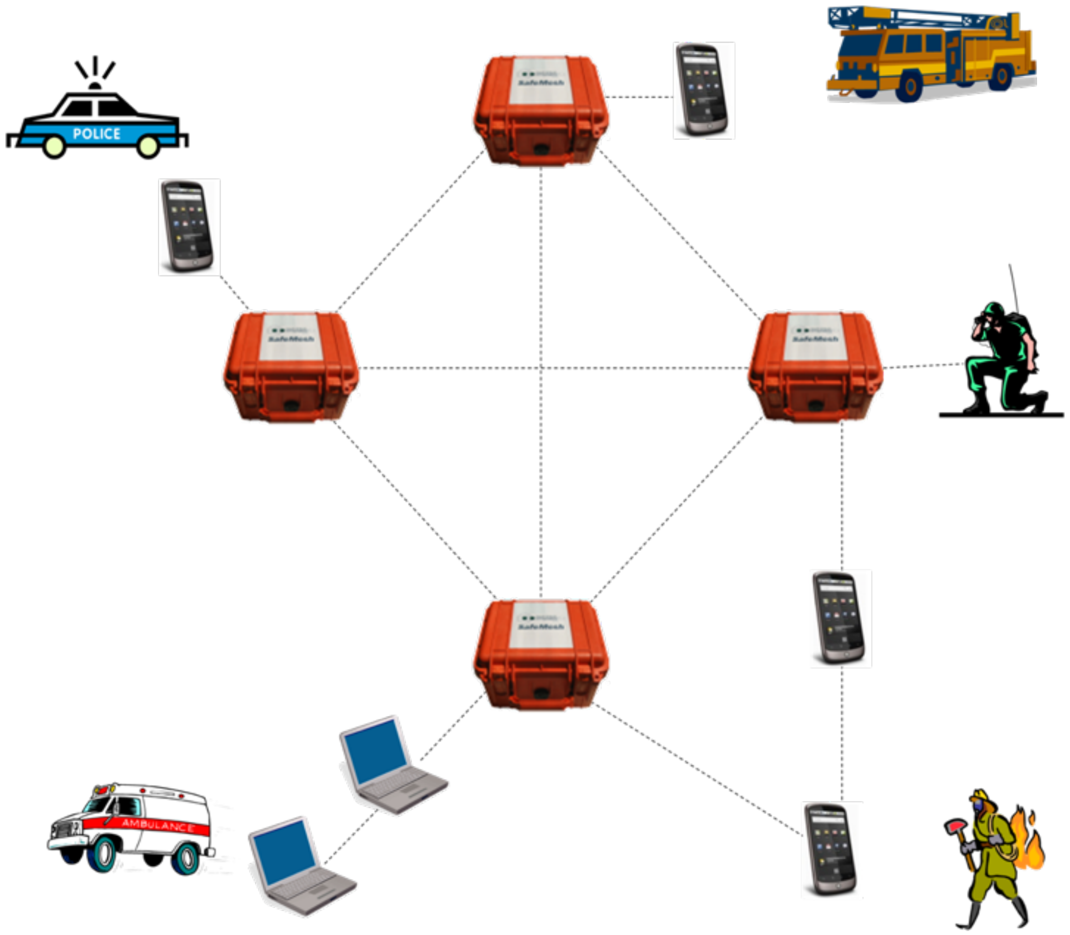
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Members

NICTA Partners

MANETs and WMNs

- Mobile Ad Hoc Networks (MANETs)
  Wireless Mesh Networks (WMNs)
  - key features: mobility, dynamic topology, wireless multihop backhaul
  - quick and low cost deployment
- Applications
  - public safety
  - emergency response, disaster recovery
  - transportation
  - smart grid
  - ...
- Limitations in reliability and performance



Formal Methods for Mesh Networks

- Goal
  - model, analyse, verify and increase the performance of wireless mesh routing protocols
  - develop suitable formal methods techniques

- Benefits
  - more reliable protocols
  - finding and fixing bugs
  - better performance
  - proving correctness
  - reduce “time-to-market”

AODVv2

- Dynamic MANET On-demand (AODVv2) Routing
  - routing protocol for WMNs and MANETs

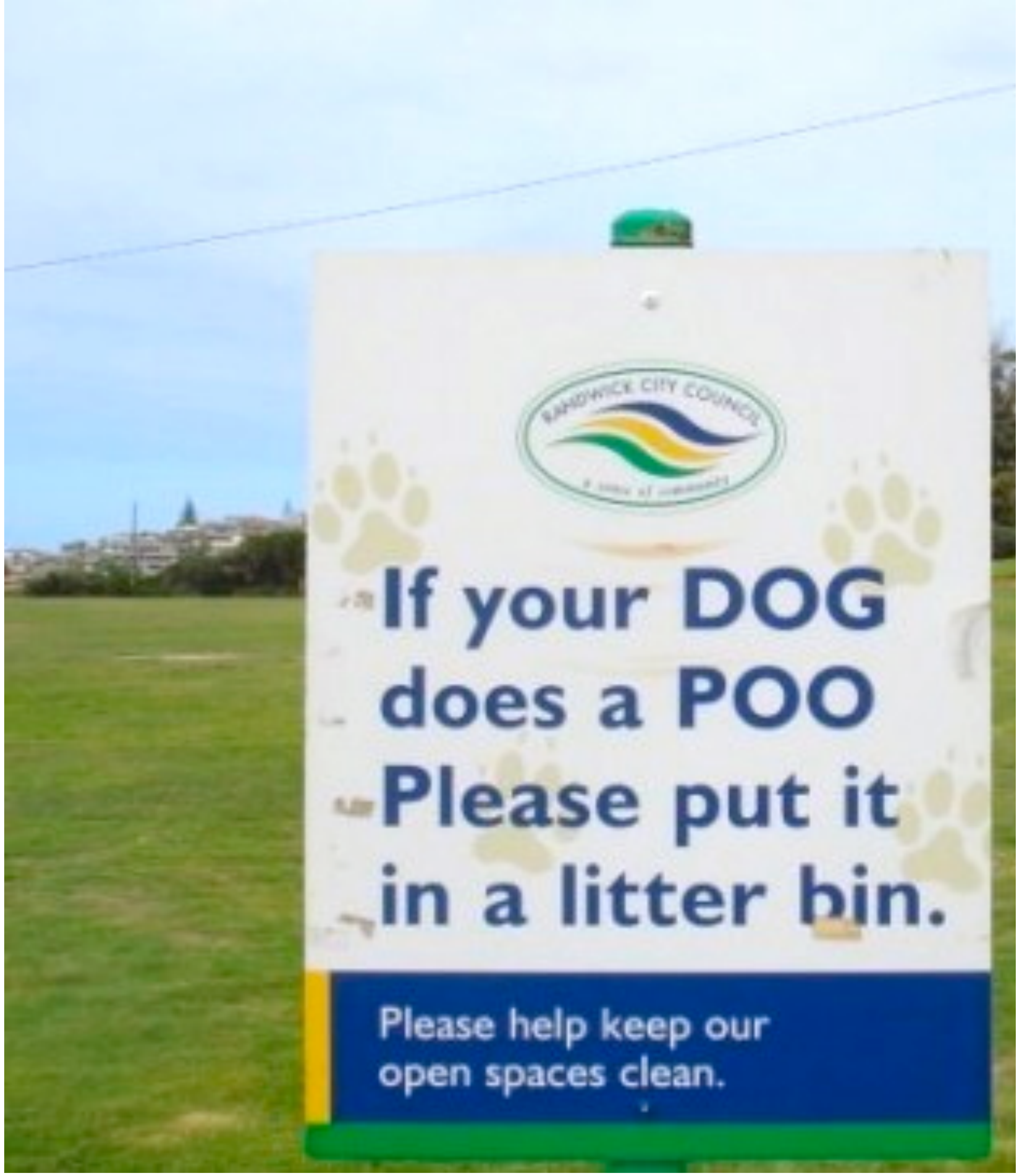
  - ad hoc (network is not static)
  - on-Demand (routes are established when needed)
  - distance (metric is hop count)

  - latest draft July 2012,
    previously known as DYMO

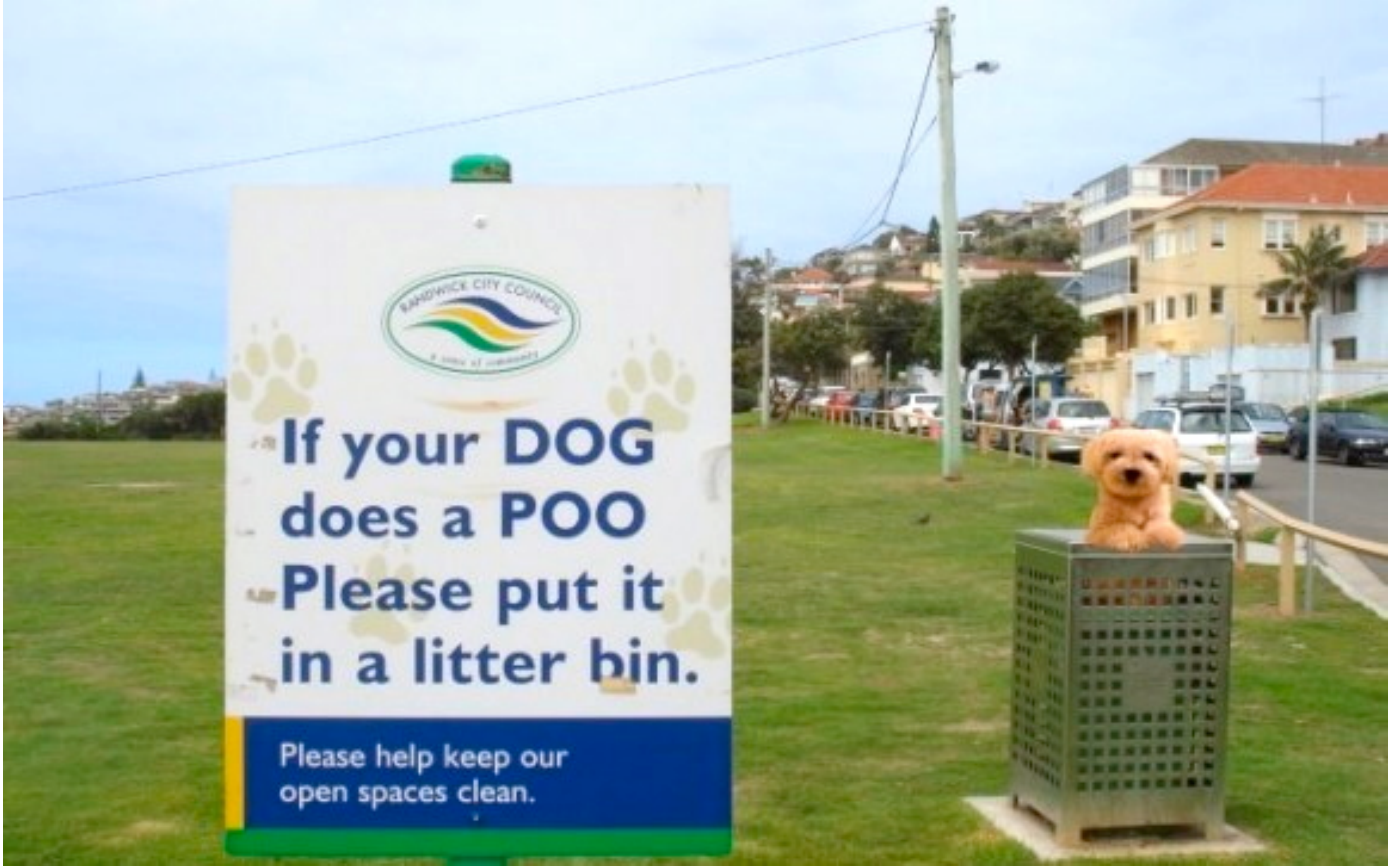
Towards a Rigorous Analysis

- Standards (IETF RFCs) are not precise
  - – written in English

Why Formal Specification?



Why Formal Specification?


RANDWICK CITY COUNCIL
a sense of community
If your DOG does a POO Please put it in a litter bin.
Please help keep our open spaces clean.

Formal Specification

- Standards (IETF RFCs) are not precise
  - written in English
  - ambiguous (sometimes incomplete)
  - no formal specification

- Rigorous Analysis needs Formal Specification

- Previous Experience with AODV: Compliant implementations
  - have different behaviours
  - are not compatible
  - have serious flaws

Complete and Accurate Formalisation of AODVv2

```
[ ip = tip ]        /* node is target node */
    ⟦sn := sn + 1⟧        /* increment node's own sequence number */
    /* generate rrep message */
    unicast(nhop(rt,oip),rrep(ip,10,oip,osn,ip,sn,0,∅)).DYMO(ip,sn,rt,store)
    ▶ /* if the transmission is unsuccessful, a RERR message is generated */
        ⟦unodes := {(rip, sqn(rt,rip)) | rip ∈ kD(rt) ∧ nhop(rt,rip) = nhop(rt,oip)}⟧
        ⟦rt := invalidate(rt,unodes)⟧
        broadcast(rerr(ip,10,unodes)).DYMO(ip,sn,rt,store)
+ [ ip ≠ tip ]        /* node is not target node */
    (
        [ tip ∈ kD(rt) ∧ sqn(rt,tip) > tsn ]        /* intermediate node generates route reply */
            ⟦sn := sn + 1⟧        /* intermediate node increments its own sequence number */
            unicast(nhop(rt,oip),rrep(ip,10,oip,osn,ip,sn,0,{(tip,sqn(rt,tip),dist(rt,tip))})).
                (
                unicast(nhop(rt,tip),rrep(ip,10,tip,tsn,ip,sn,0,inodes ∪ {(oip, osn, odist+1)})).
                ▶ /* If the transmission of the rrep to tip is unsuccessful, a RERR message is generated */
                    ⟦unodes := {(rip, sqn(rt,rip)) | rip ∈ kD(rt) ∧ nhop(rt,rip) = nhop(rt,tip)}⟧
                    ⟦rt := invalidate(rt,unodes)⟧
                    broadcast(rerr(ip,10,unodes)).DYMO(ip,sn,rt,store)
                )
            ▶ /* If the transmission of the rrep to oip is unsuccessful, a RERR message is generated */
```

Proposed Formal Method

- Based on Process Algebra AWN
  - – inspired by π-calculus and LOTOS; based on ω-calculus
  - – main process expressions

$X(\mathit{exp}_1, \ldots, \mathit{exp}_n)$	process calls
$P + Q$	nondeterministic choice
$[\varphi]P$	if-construct
$[\![\texttt{var} := \mathit{exp}]\!]P$	assignment followed by P
$\textbf{broadcast}(\mathit{ms}).P$	broadcast message followed by P
$\textbf{unicast}(\mathit{dest},\mathit{ms}).P \blacktriangleright Q$	unicast ms to dest; if successful proceed with P; otherwise with Q
$\textbf{receive}(\texttt{msg}).P$	receive message

Requirements for Formal Methods

- "Formal languages are useful tools for specifying parts of protocols. However, as of today, there exists no well-known language that is able to capture the full syntax and semantics of reasonably rich IETF protocols.”

  [IETF]

- IETF Requirements (for formal methods)
  - relatively easy to extract code
  - complete specification
  - implementation independent

- Easy to use
  - only a few (well-known) programming constructs

Rigorous Analysis of AODV

- Achievements
  - full concise specification of AODVv2 (Internet-Draft 23 + Intermediate Route Reply)
    - 6 processes (~120 lines; instead of 40 pages English prose)
    - without time
  - first analysis of routing properties (shortcomings of AODV)
    - route discovery
    - message loss
    - non-optimal routes
    - loop freedom
  - found ambiguities, contradictions, shortcomings

Rigorous Analysis of AODV

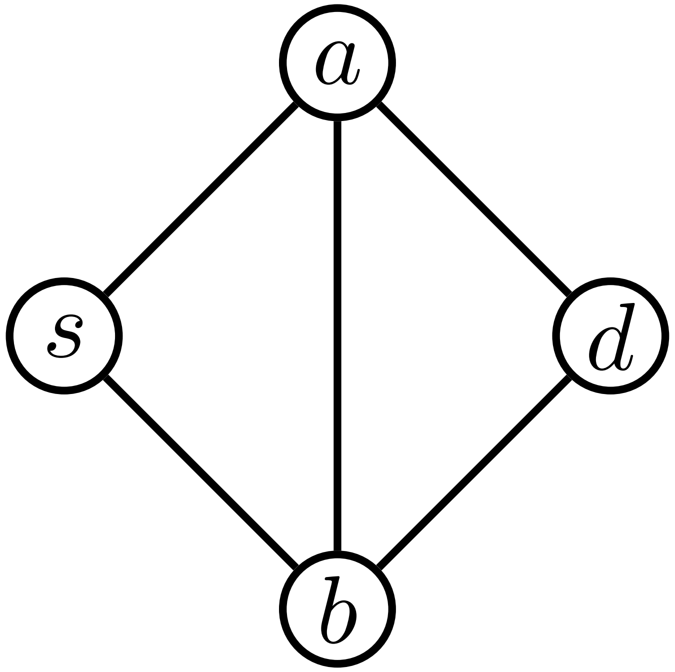
- Achievements
  - – proved that formal analysis can be quick
    - started March 2012
    - changed to newest draft inJuly
    - finished beginning of August
    - (in fact even faster if specification would be given formally)
  - – our developed method does not only work for AODV

A First Analysis

```
[ $\mathtt{ip} = \mathtt{tip}$ ]    /* node is target node */
    $[\![\mathtt{sn} := \mathtt{sn}+1]\!]$    /* increment node's own sequence number */
    /* generate rrep message */
    unicast(nhop(rt,oip),rrep(ip,10,oip,osn,ip,sn,0,$\emptyset$)).DYMO(ip,sn,rt,store)
    ▶ /* if the transmission is unsuccessful, a RERR message is generated */
        $[\![\mathtt{unodes} := \{(\mathtt{rip},\mathtt{sqn}(\mathtt{rt},\mathtt{rip}))\,|\,\mathtt{rip} \in \mathtt{kD}(\mathtt{rt}) \wedge \mathtt{nhop}(\mathtt{rt},\mathtt{rip}) = \mathtt{nhop}(\mathtt{rt},\mathtt{oip})\}]\!]$
        $[\![\mathtt{rt} := \mathtt{invalidate}(\mathtt{rt},\mathtt{unodes})]\!]$
        broadcast(rerr(ip,10,unodes)).DYMO(ip,sn,rt,store)
+ [ $\mathtt{ip} \neq \mathtt{tip}$ ]    /* node is not target node */
    (
        [ $\mathtt{tip} \in \mathtt{kD}(\mathtt{rt}) \wedge \mathtt{sqn}(\mathtt{rt},\mathtt{tip}) > \mathtt{tsn}$ ]    /* intermediate node generates route reply */
            $[\![\mathtt{sn} := \mathtt{sn}+1]\!]$    /* intermediate node increments its own sequence number */
            unicast(nhop(rt,oip),rrep(ip,10,oip,osn,ip,sn,0,{(tip,sqn(rt,tip),dist(rt,tip))})).
                (
                unicast(nhop(rt,tip),rrep(ip,10,tip,tsn,ip,sn,0,inodes $\cup$ {(oip, osn, odist+1)})).
                ▶ /* If the transmission of the rrep to tip is unsuccessful, a RERR message is generated */
                    $[\![\mathtt{unodes} := \{(\mathtt{rip},\mathtt{sqn}(\mathtt{rt},\mathtt{rip}))\,|\,\mathtt{rip} \in \mathtt{kD}(\mathtt{rt}) \wedge \mathtt{nhop}(\mathtt{rt},\mathtt{rip}) = \mathtt{nhop}(\mathtt{rt},\mathtt{tip})\}]\!]$
                    $[\![\mathtt{rt} := \mathtt{invalidate}(\mathtt{rt},\mathtt{unodes})]\!]$
                    broadcast(rerr(ip,10,unodes)).DYMO(ip,sn,rt,store)
                )
            ▶ /* If the transmission of the rrep to oip is unsuccessful, a RERR message is generated */
```

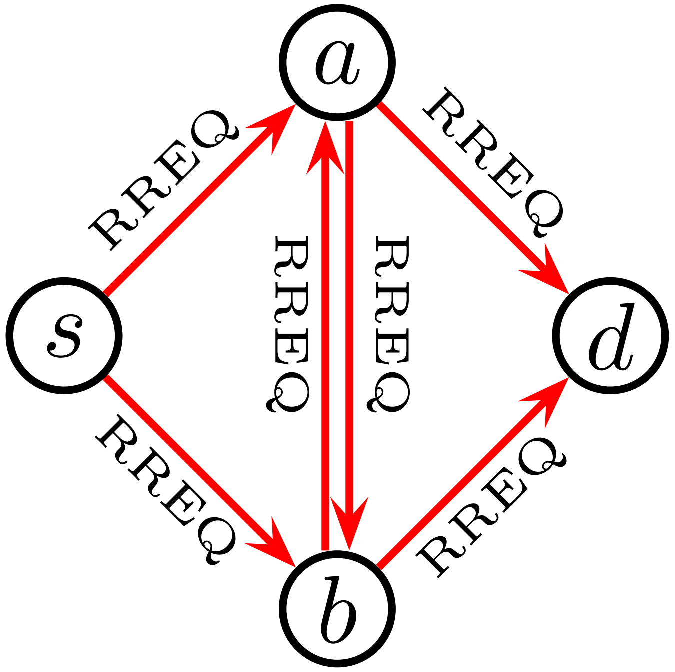
AODV and AODVv2

- Main Mechanism
  - if route is needed
    BROADCAST RREQ
  - if node has information about a destination
    UNICAST RREP
  - if unicast fails or link break is detected
    SEND RERR



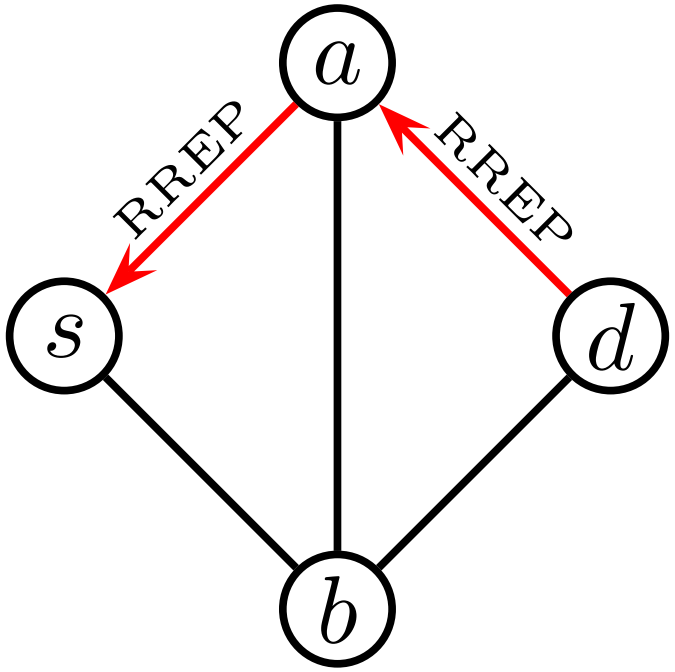
AODV and AODVv2

- Main Mechanism
  - if route is needed
    BROADCAST RREQ
  - if node has information about a destination
    UNICAST RREP
  - if unicast fails or link break is detected
    SEND RERR

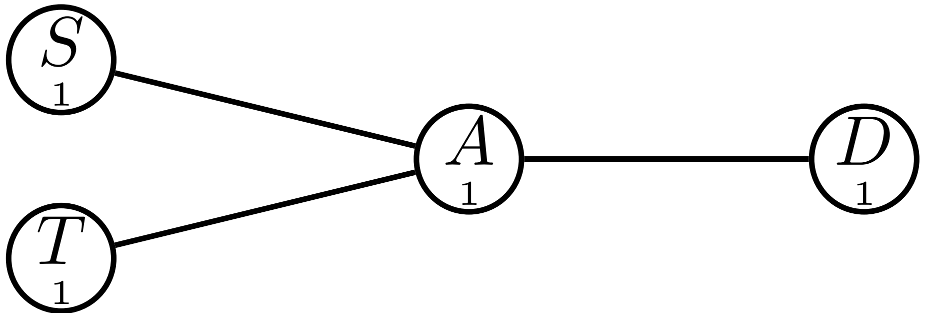


AODV and AODVv2

- Main Mechanism
  - – if route is needed
    BROADCAST RREQ
  - – if node has information about a destination
    UNICAST RREP
  - – if unicast fails or link break is detected
    SEND RERR



AODV: Failure of Route Discovery Process

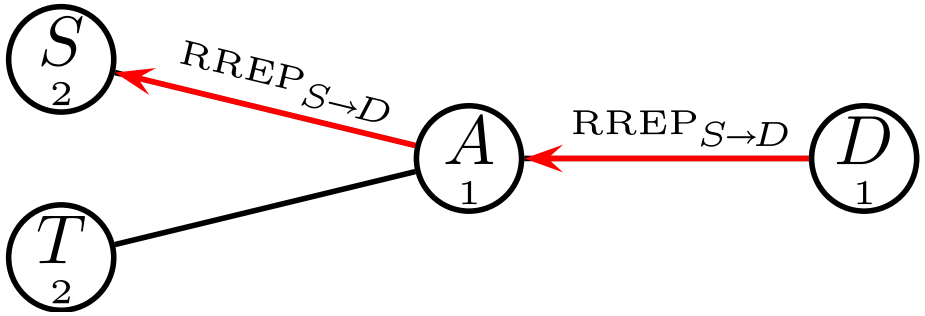


- Route replies are dropped if they do not carry new information; this might yield route discovery failure

  [IETF Mailing List]

  – Problem: only “new” information is forwarded

AODV: Failure of Route Discovery Process

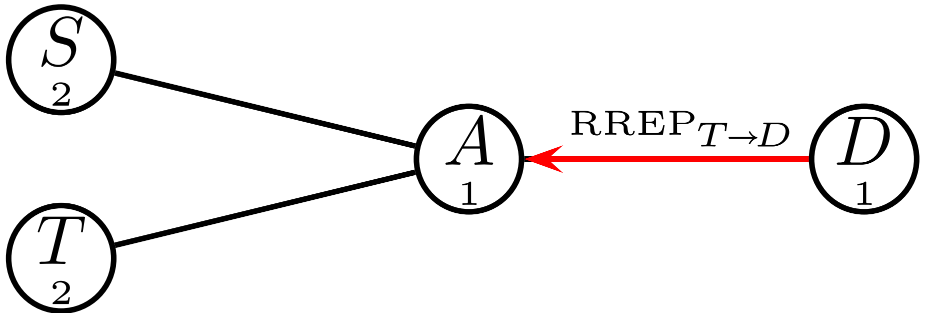


- Route replies are dropped if they do not carry new information; this might yield route discovery failure

  [IETF Mailing List]

  – Problem: only “new” information is forwarded

AODV: Failure of Route Discovery Process

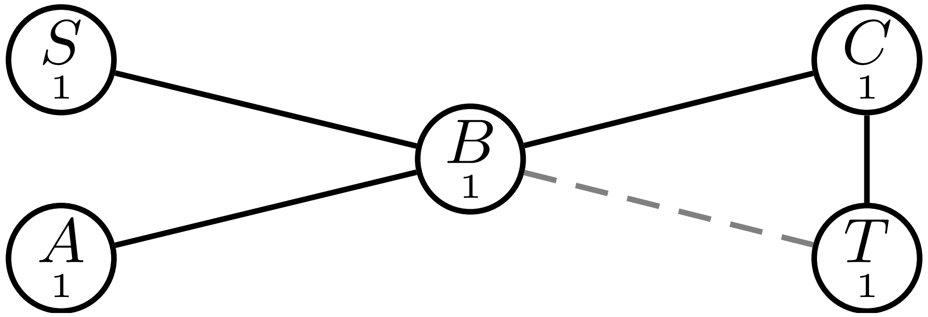


- Route replies are dropped if they do not carry new information; this might yield route discovery failure

  [IETF Mailing List]

  – Problem: only “new” information is forwarded

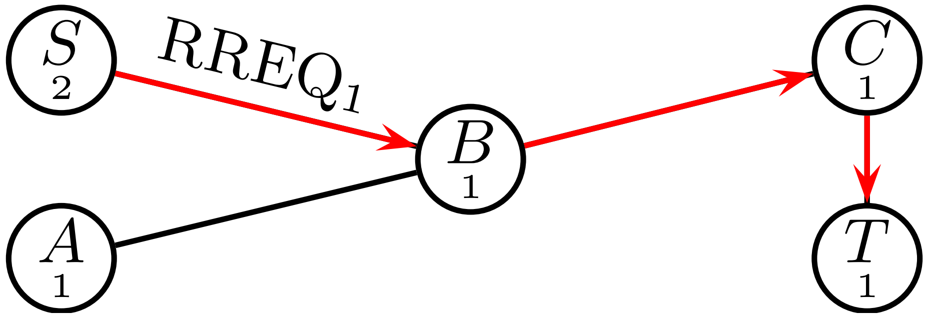
AODVv2: Failure of Route Discovery Process



- Sequence numbers are increased when reply is initiated
  - – major flaw fixed
  - – problem with overtaking messages
    - occurs in replies *and* requests
    - unclear how often this shortcoming occurs

- Consequence: *route discovery* cannot be guaranteed
  - – possible solution: always forward route replies

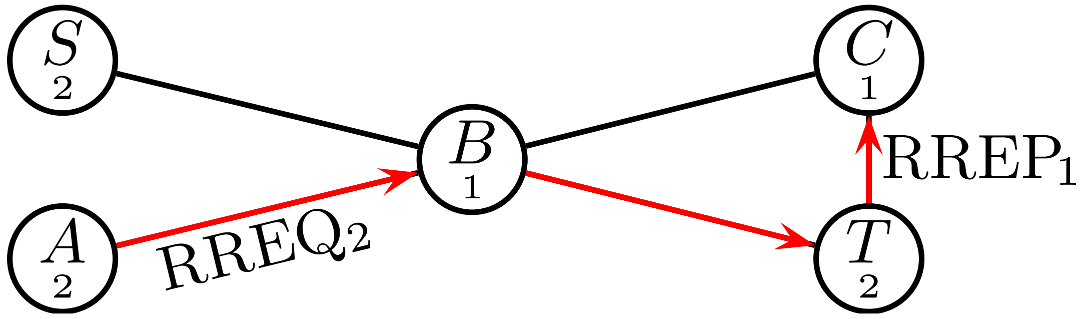
AODVv2: Failure of Route Discovery Process



- Sequence numbers are increased when reply is initiated
  - – major flaw fixed
  - – problem with overtaking messages
    - occurs in replies *and* requests
    - unclear how often this shortcoming occurs

- Consequence: *route discovery* cannot be guaranteed
  - – possible solution: always forward route replies

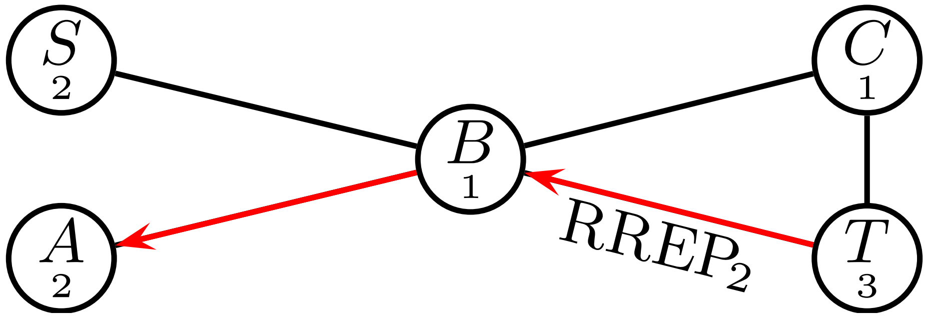
AODVv2: Failure of Route Discovery Process



- Sequence numbers are increased when reply is initiated
  - major flaw fixed
  - problem with overtaking messages
    - occurs in replies *and* requests
    - unclear how often this shortcoming occurs

- Consequence: *route discovery* cannot be guaranteed
  - possible solution: always forward route replies

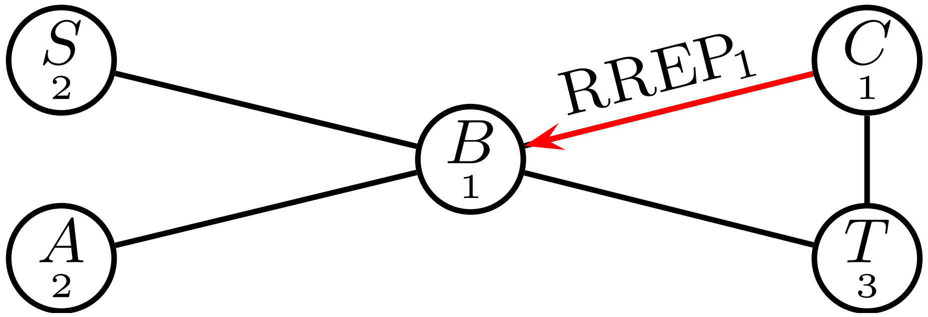
AODVv2: Failure of Route Discovery Process



- Sequence numbers are increased when reply is initiated
  - – major flaw fixed
  - – problem with overtaking messages
    - occurs in replies *and* requests
    - unclear how often this shortcoming occurs

- Consequence: *route discovery* cannot be guaranteed
  - – possible solution: always forward route replies

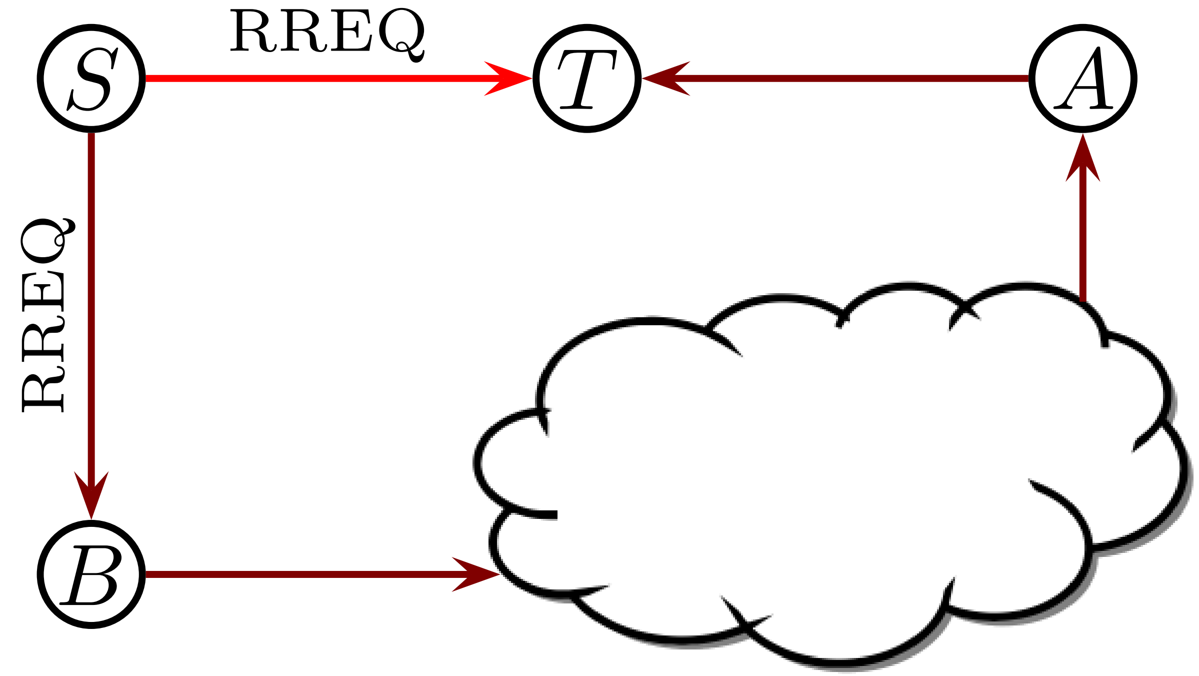
AODVv2: Failure of Route Discovery Process



- Sequence numbers are increased when reply is initiated
  - major flaw fixed
  - problem with overtaking messages
    - occurs in replies *and* requests
    - unclear how often this shortcoming occurs

- Consequence: *route discovery* cannot be guaranteed
  - possible solution: always forward route replies

Non-Optimal Route Selection



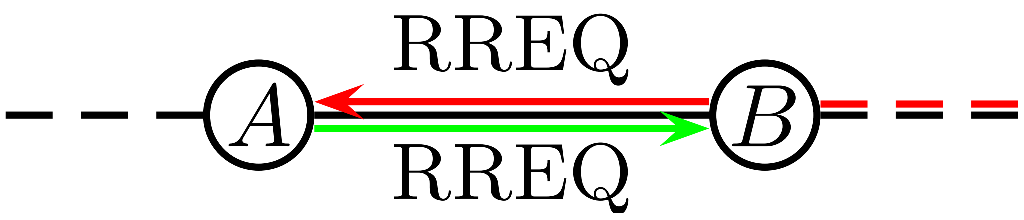
- during route discovery *only* nodes lying on route from source to destination find optimal routes

  [MiskovicKnightly10]

- problem of AODV and AODVv2
  - duration [of poorly selected paths] can extend to minute time scales

- modification: forward route request

Loop Freedom



- Loop freedom of AODV
  - does not only depend on sequence numbers, but also on
    - error handling
    - self-entries
  - is not guaranteed by the RFC
    - depends on interpretation
    - depends on (the experience of) the software engineer
  - some implementations, such as ns2-AODV, contain loops
  - often caused by self-entries
- Loop freedom of AODVv2
  - can be most likely guaranteed (at least in our interpretation)
  - safer: exclude self-entries

Conclusion

- Formal specification of AODVv2
  - complete, accurate (without time)
  - based on process algebra AWN
- First analysis
  - new shortcomings found
  - solutions proposed
  - done by counterexamples
- Proofs
  - independent of topology
  - modularity / reusability
    - simple to adapt variants of AODVv2
  - simulation and test-bed experiment would have to be repeated for each interpretation

Future Work

- Extend formal methods to other protocols
  - OSLR, B.A.T.M.A.N., ...
- Add further necessary concepts
  - time
  - probability (links, (quantitative) measurements)
- Formalise the “Quality” of a protocol
  - formalise measurements (PDR,...)
  - compare AODV vs AODVv2
    - there are papers stating that one is better than the other (and vice versa)

NICTA

From **imagination** to **impact**