Formal Methods for Wireless Mesh Networks

Peter Höfner

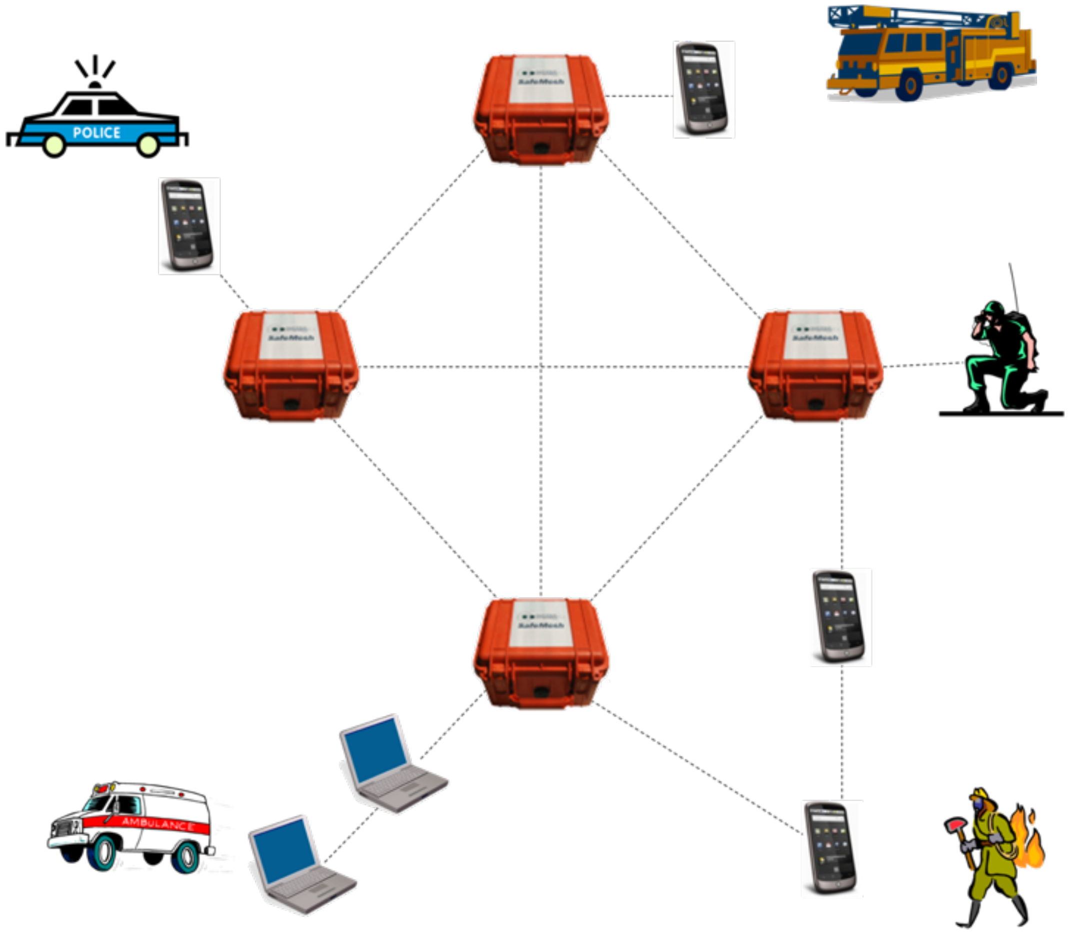
München
February 17, 2012

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

NICTA Members
NICTA Partners

What is the Problem?

- 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
  - mining
  - smart grid
  - ...
- Limitations in reliability and performance



© NICTA 2011

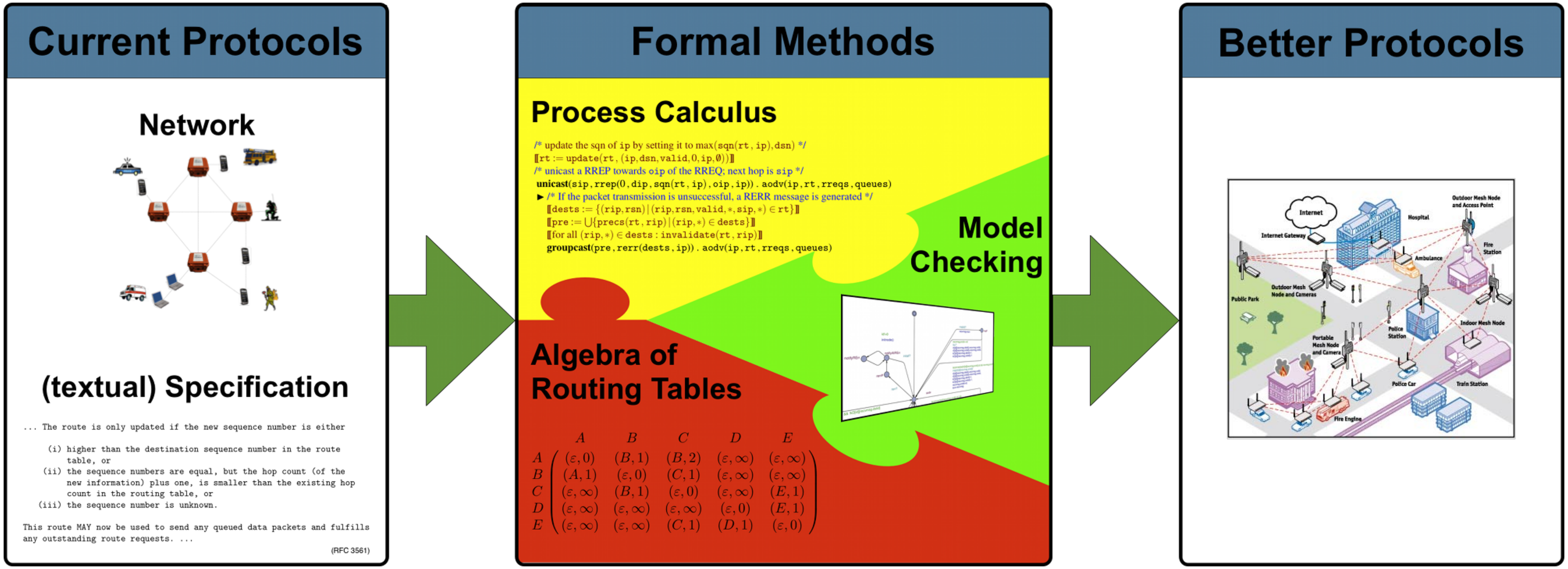
Formal Methods for Mesh Networks

- Goal
  - model, analyse, verify and increase the performance of wireless mesh protocols
  - develop suitable formal methods techniques
- Benefits
  - more reliable protocols
  - finding and fixing bugs
  - better performance
  - proving correctness
  - reduce “time-to-market”
- Team (Formal Methods)
  - Ansgar Fehnker, Rob van Glabbeek, Peter Höfner, Annabelle McIver, Marius Portmann, Wee Lum Tan

© NICTA 2011

Formal Methods for Mesh Networks

- Main Methods used so far
  - process algebra
  - model checking
  - routing algebra



© NICTA 2011

Ad Hoc On-Demand Distance Vector Protocol

- Routing protocol for WMNs

- Ad hoc (network is not static)
- On-Demand (routes are established when needed)
- Distance (metric is hop count)
- Vector (routing table has the form of a vector)

- Developed 1997-2001 by Perkins, Beldig-Royer and Das (University of Cincinnati)

© NICTA 2011

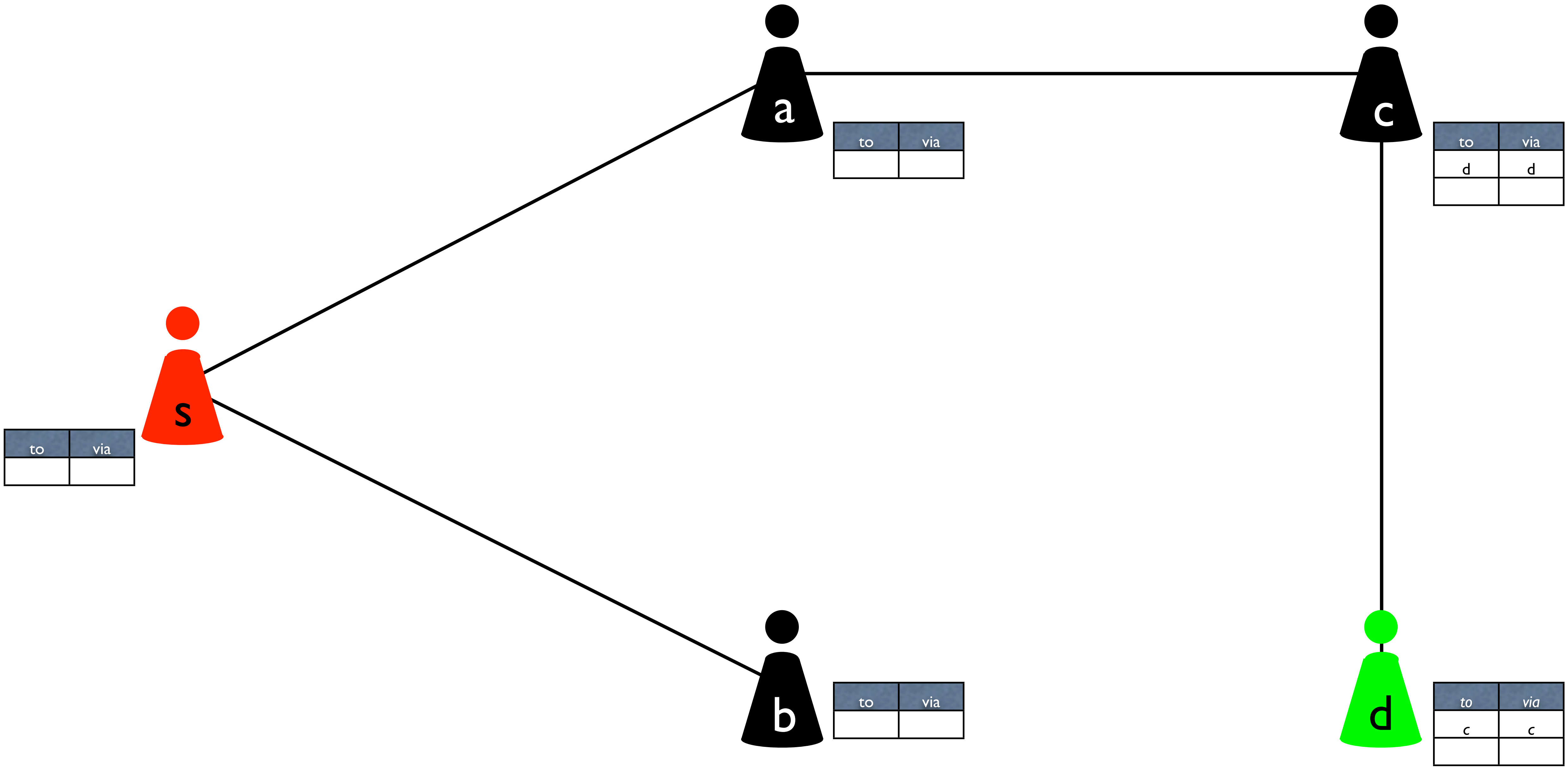
Ad Hoc On-Demand Distance Vector Protocol

- AODV control messages
  - route request (RREQ)
  - route reply (RREP)
  - route error message (RERR)

- 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

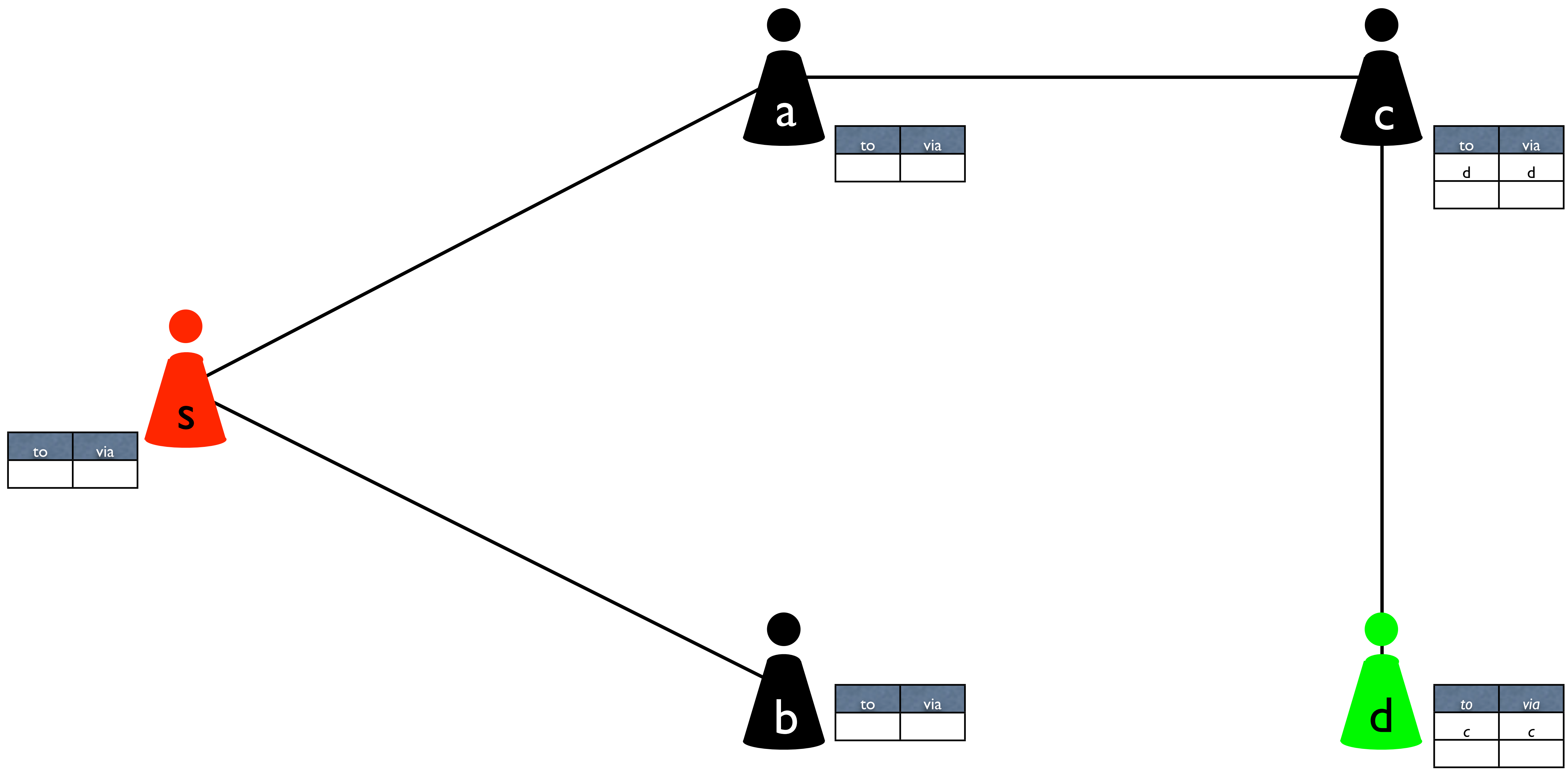
© NICTA 2011

AODV – An Example

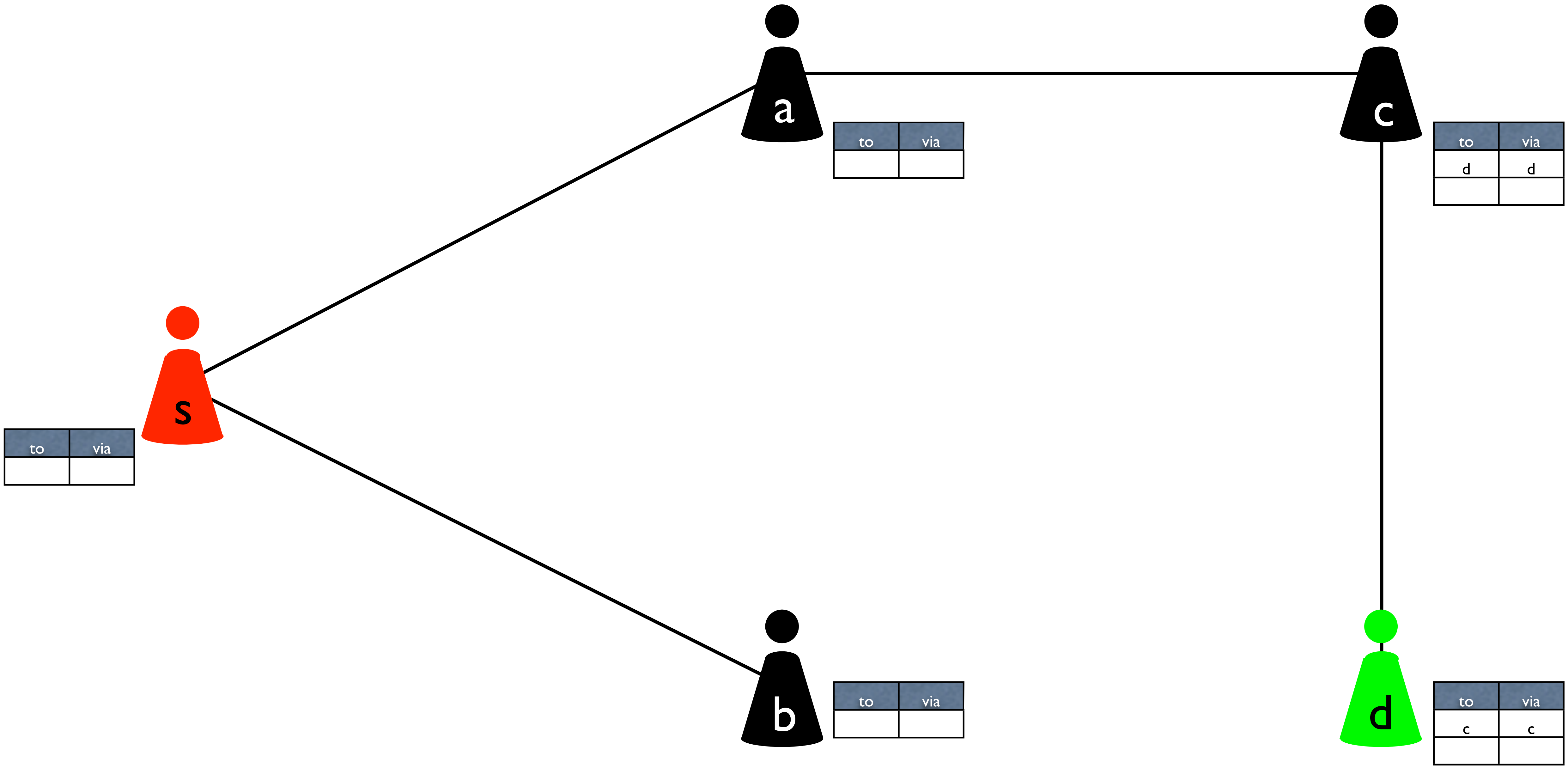


s is looking for a route to d

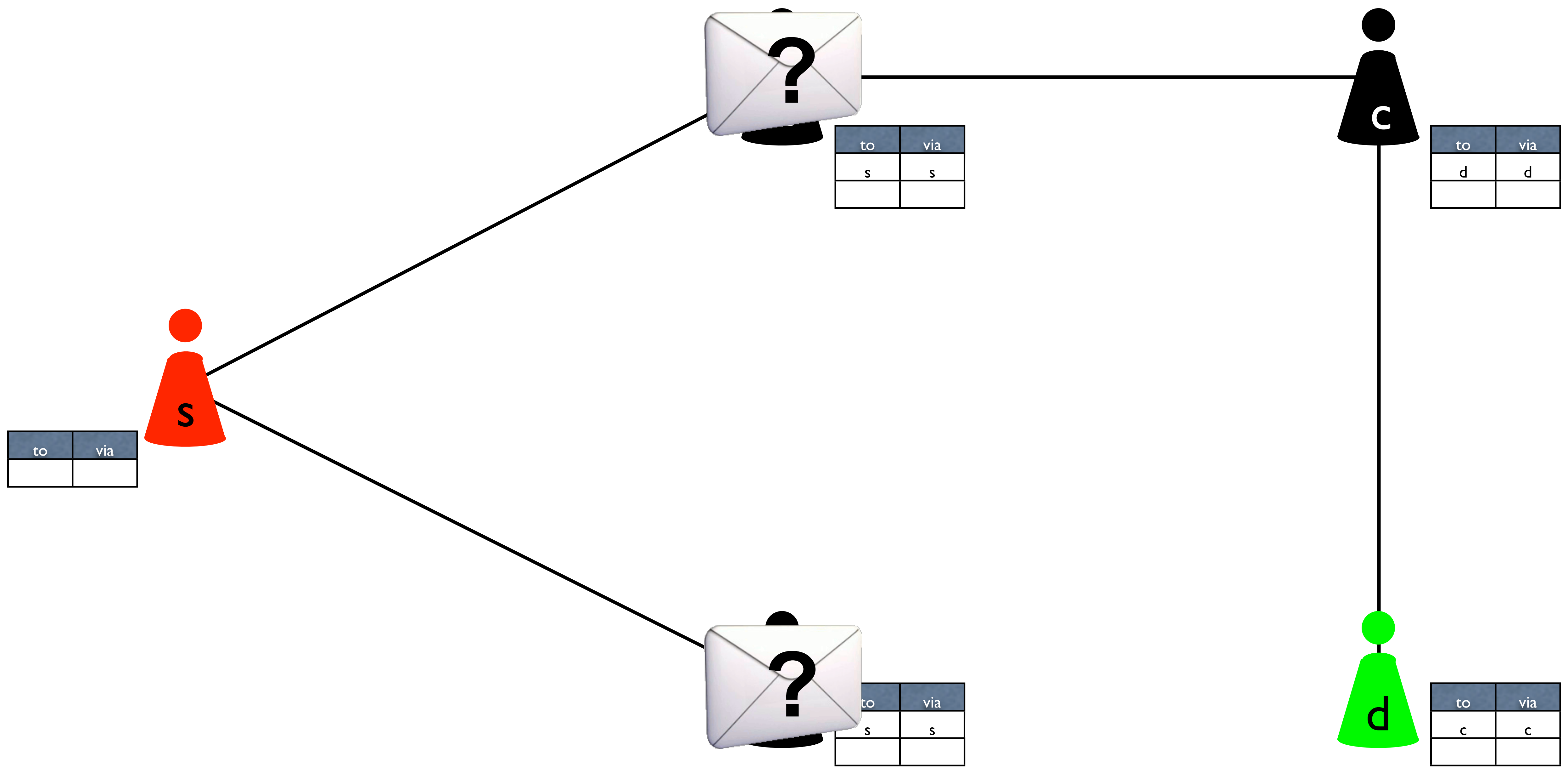
AODV – An Example



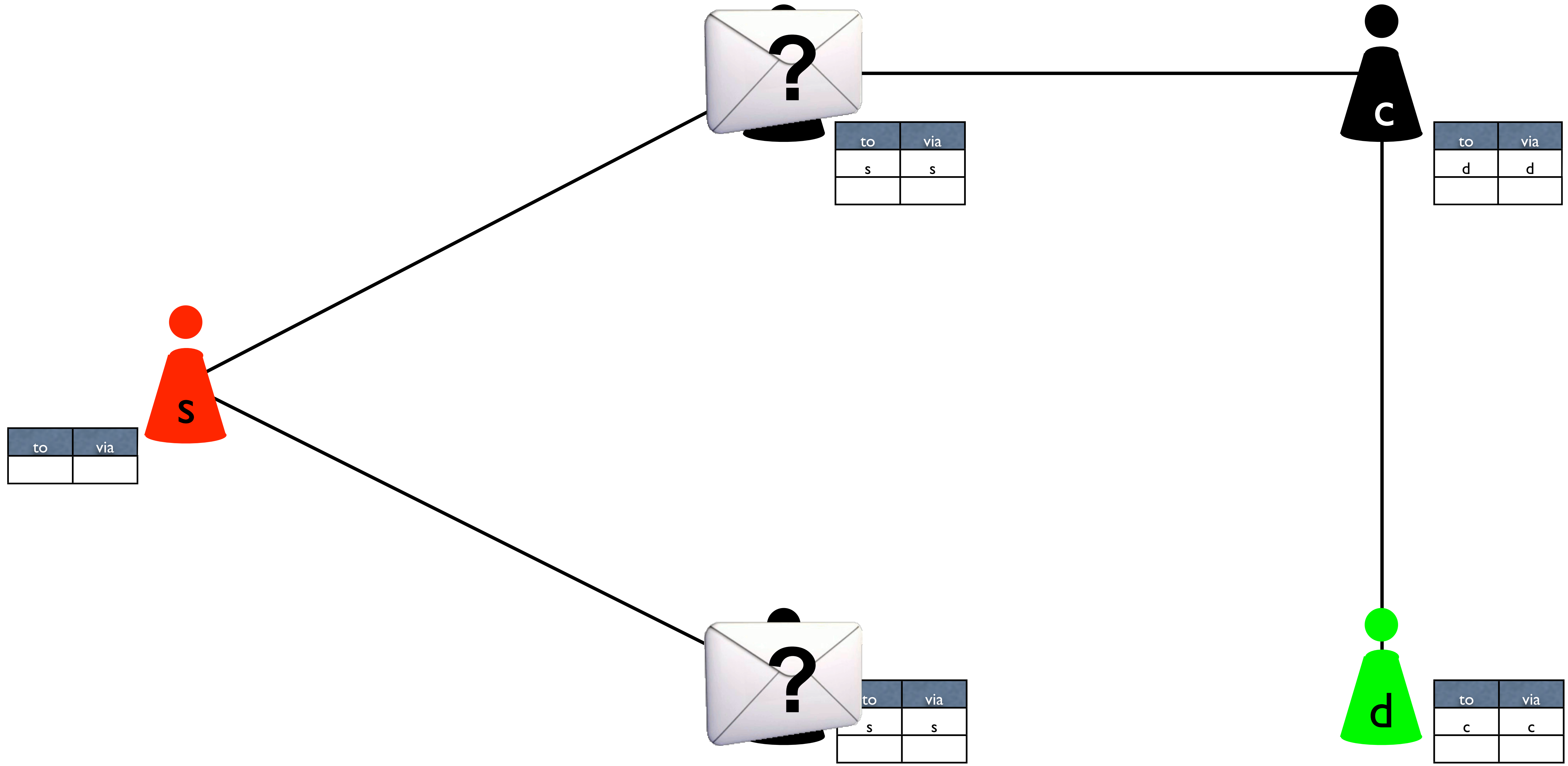
AODV - An Example



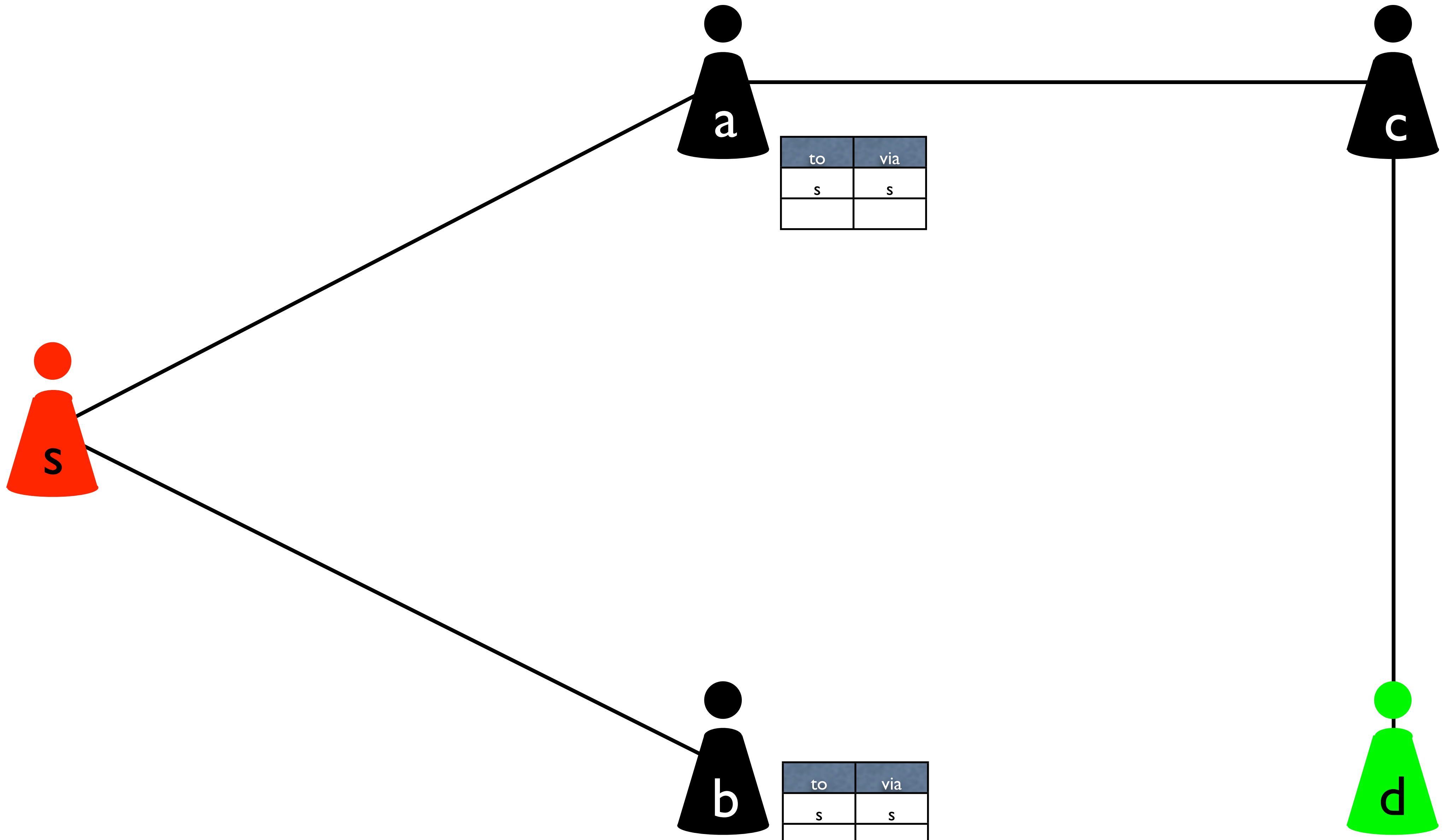
AODV – An Example



AODV - An Example



AODV - An Example



Routing table at a:

to	via
s	s

Routing table at c:

to	via
d	d
a	a
s	a

Routing table at s:

to	via
a	a
b	b

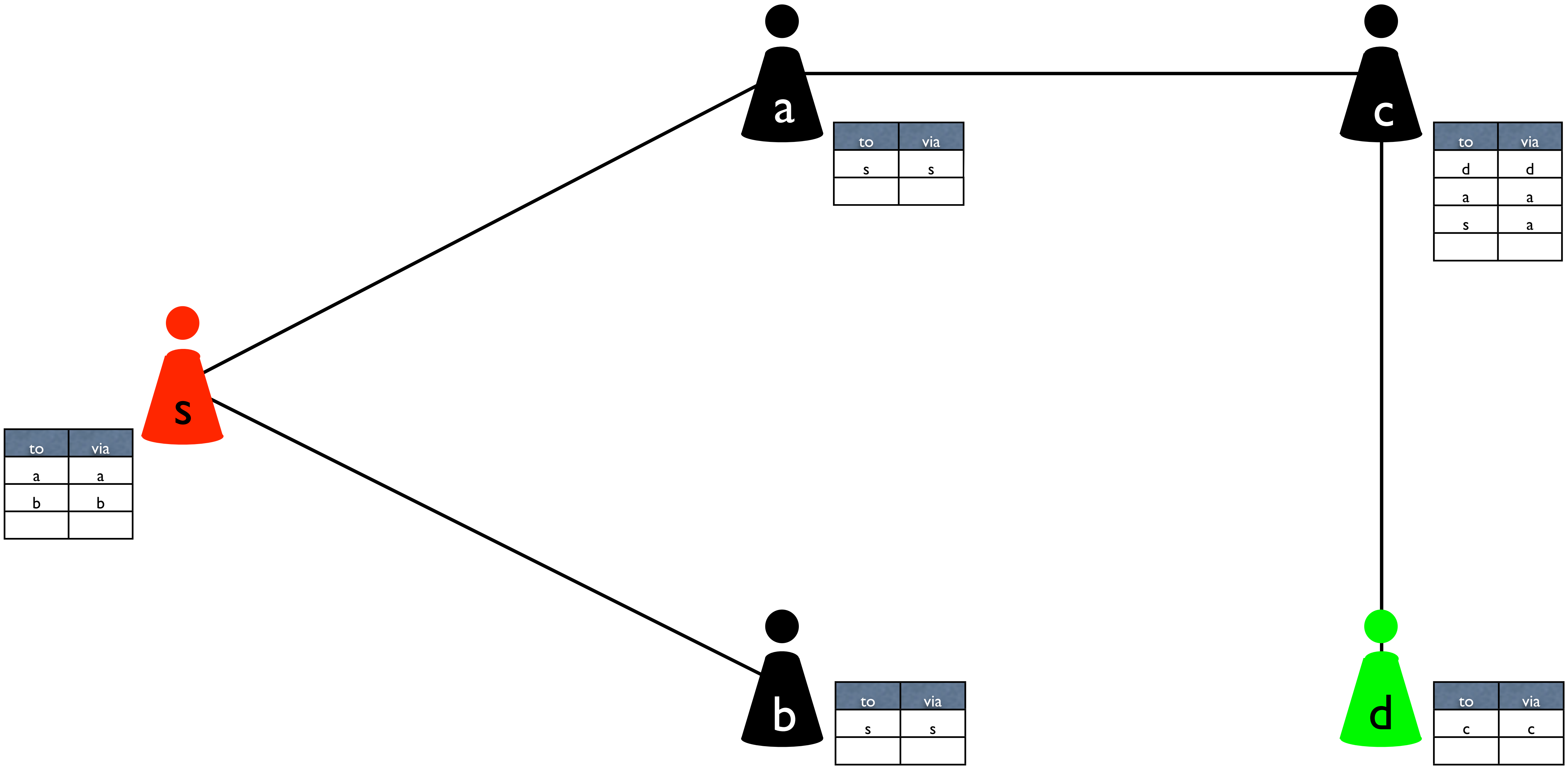
Routing table at b:

to	via
s	s

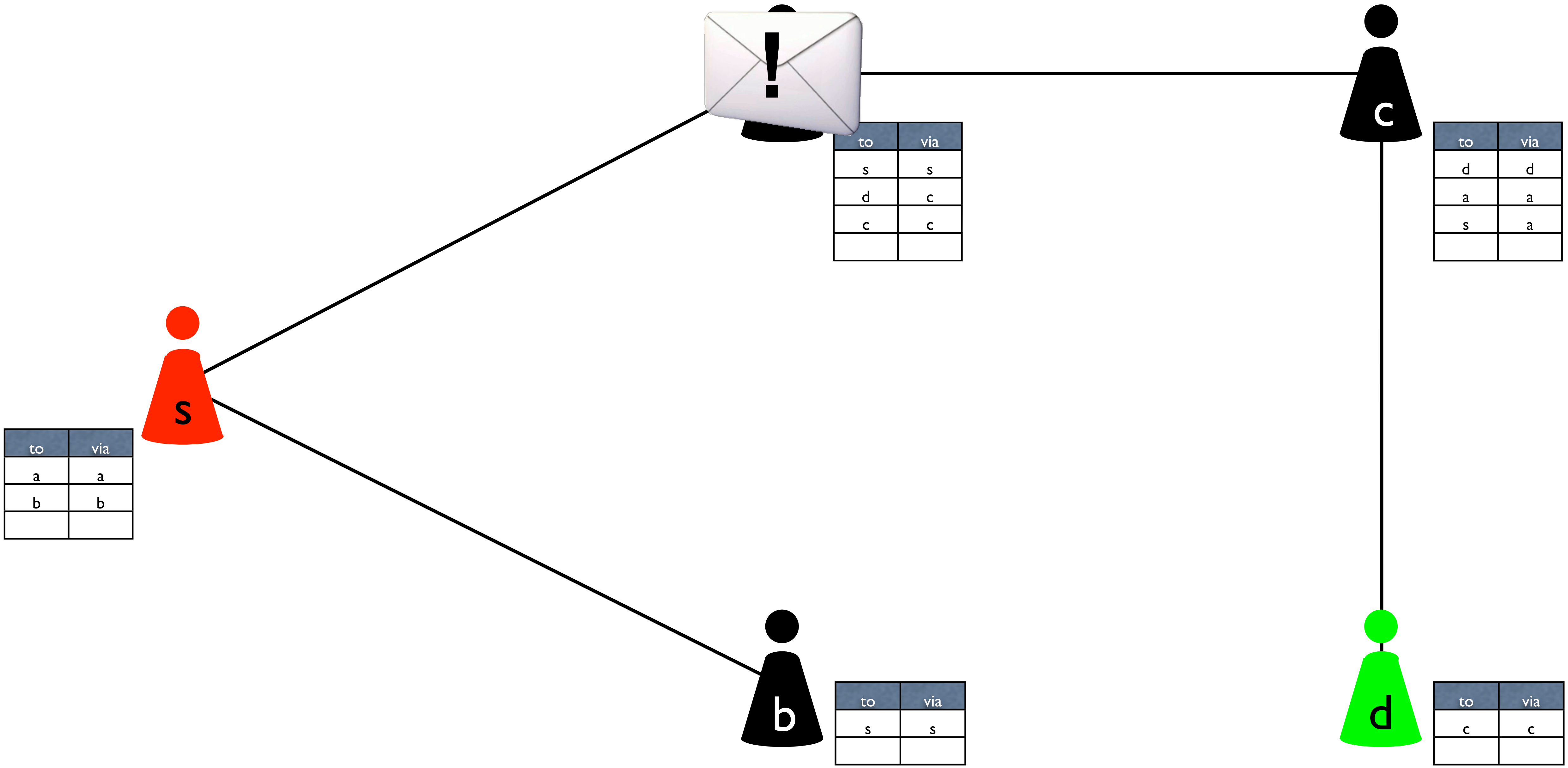
Routing table at d:

to	via
c	c

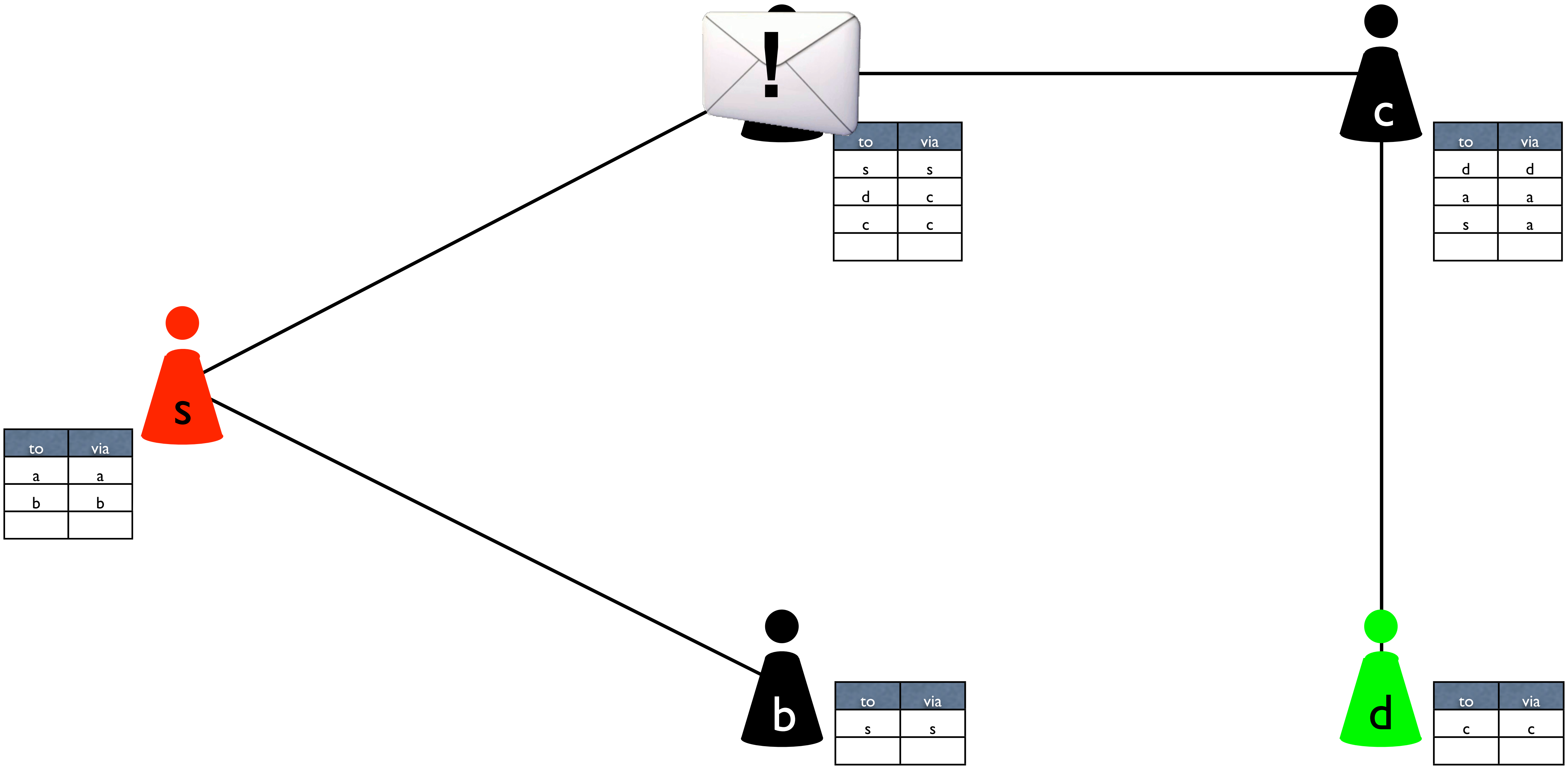
AODV – An Example



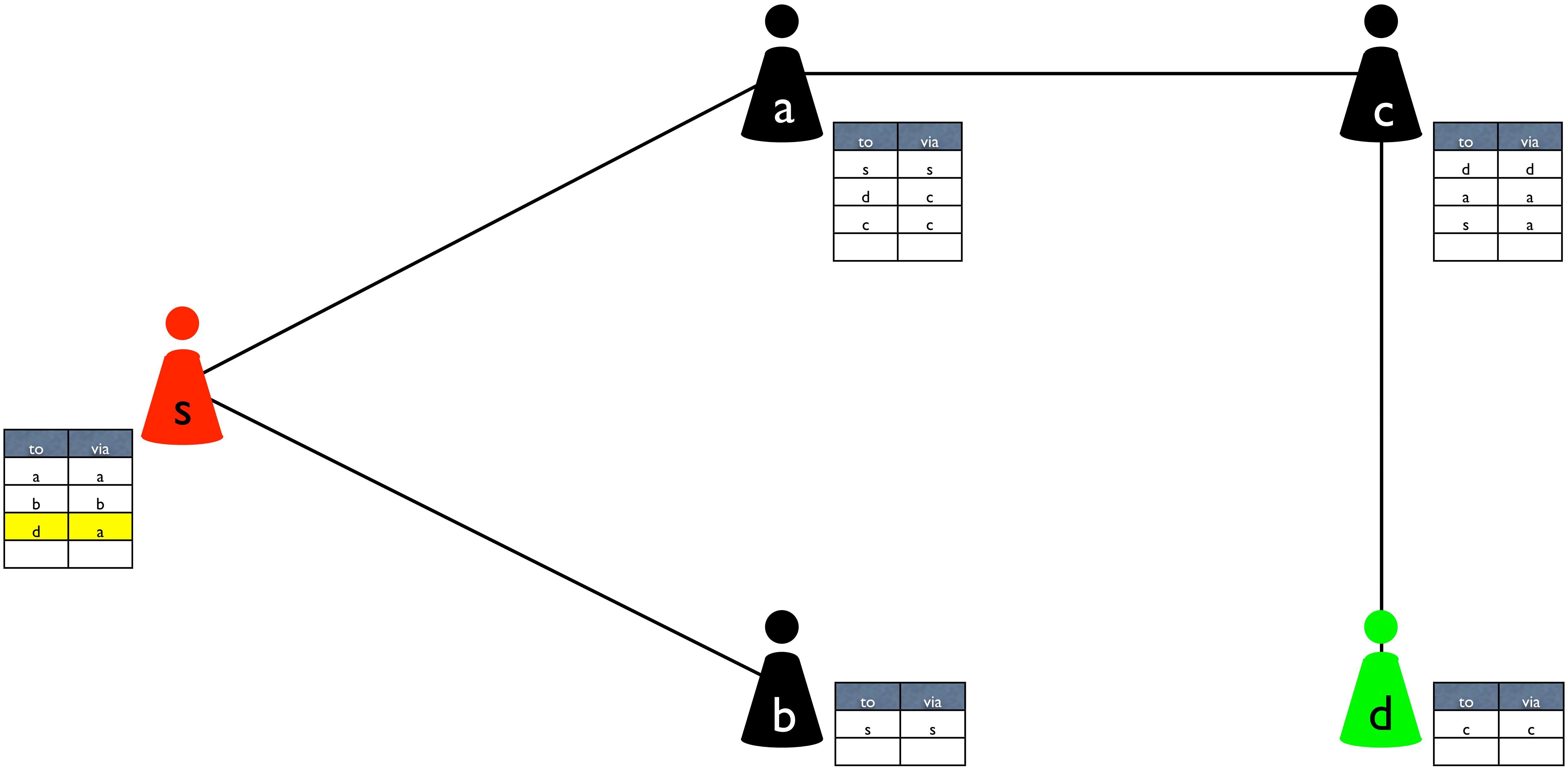
AODV – An Example



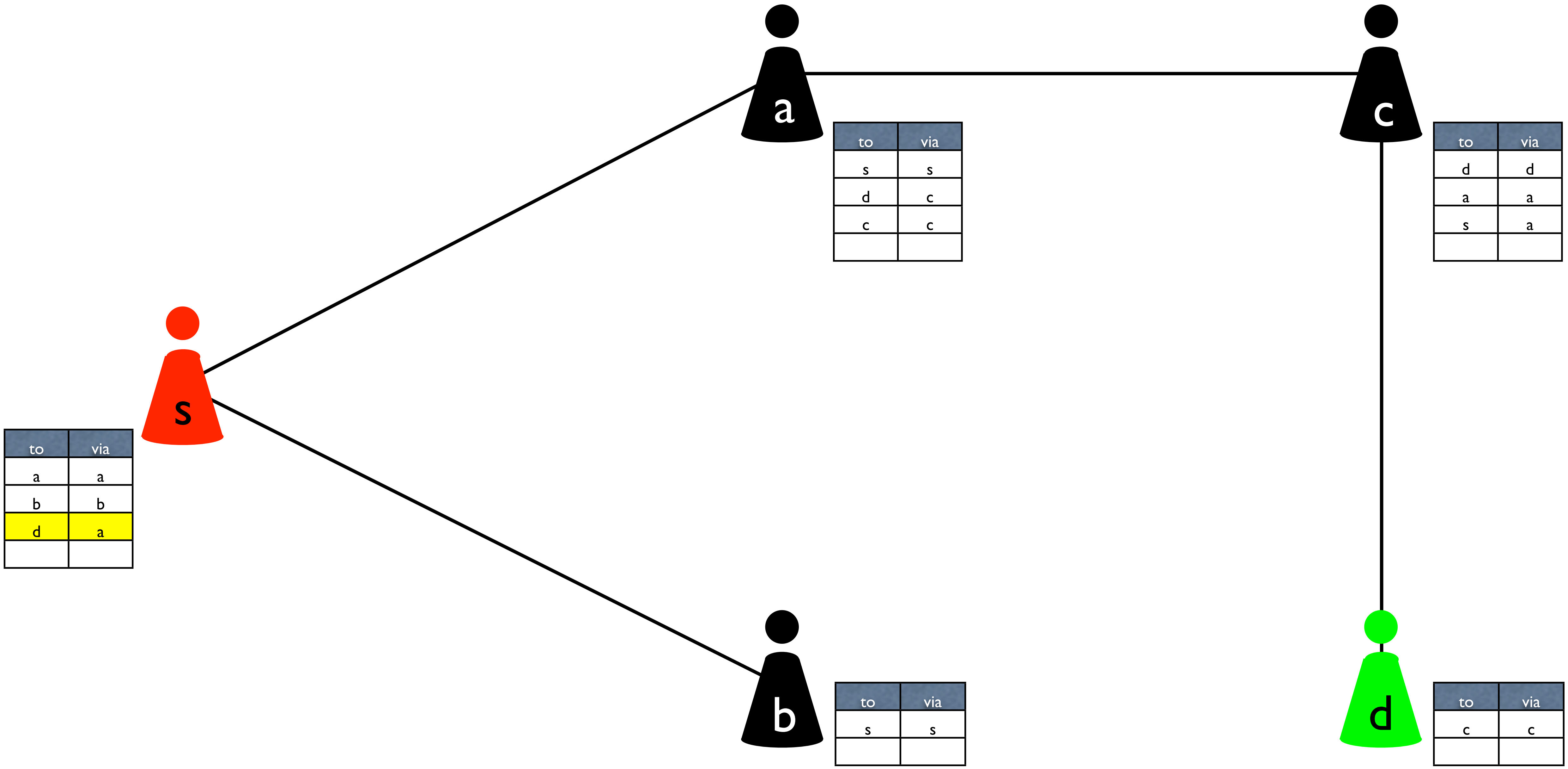
AODV – An Example



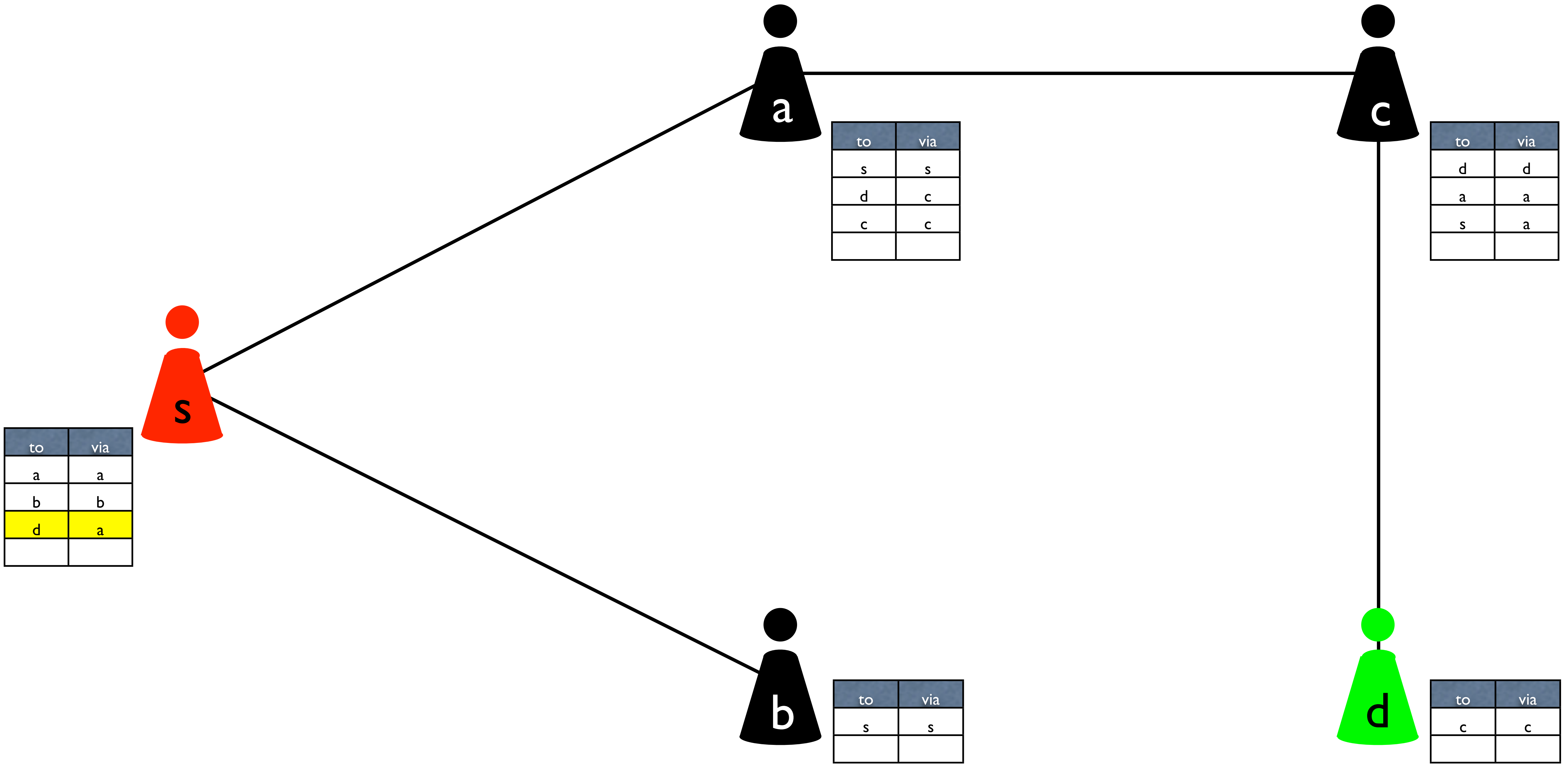
AODV – An Example



AODV - An Example



AODV – An Example



s has found a route to d

Ad Hoc On-Demand Distance Vector Protocol

- Properties of AODV
  - route correctness
  - loop freedom
  - route discovery
  - packet delivery

© NICTA 2011

Ad Hoc On-Demand Distance Vector Protocol

NICTA

- Properties of AODV
  - route correctness ✓
  - loop freedom ✓ (at least for some interpretations)
  - route discovery ✗
  - packet delivery ✗

© NICTA 2011

Ad Hoc On-Demand Distance Vector Protocol

- Properties of AODV
  - route correctness ✓
  - loop freedom ✓
  - route discovery ✗
  - packet delivery ✗
- so far only simulation and test-bed evaluations
  - important, valid methods
  - limitations
    - resource intensive, time-consuming, no generality

© NICTA 2011

Ad Hoc On-Demand Distance Vector Protocol

NICTA

- Properties of AODV
  - route correctness ✓
  - loop freedom ✓ (at least for some interpretations)
  - route discovery ✗
  - packet delivery ✗
- so far only simulation and test-bed evaluations
  - important, valid methods
  - limitations
    - resource intensive, time-consuming, no generality

© NICTA 2011

RFC 3561

NICTA

- Request for Comments (de facto standard)

```
sequence number field is set to false.  The route is only updated if
the new sequence number is either

(i)       higher than the destination sequence number in the route
          table, or

(ii)      the sequence numbers are equal, but the hop count (of the
          new information) plus one, is smaller than the existing hop
          count in the routing table, or

(iii)     the sequence number is unknown.
```

© NICTA 2011

Process Algebra

NICTA

```
+ [ (oip, rreqid) ∉ rreqs ]      /* the RREQ is new to this node */
  /* update the route to oip in rt */
  [[rt := update(rt, (oip, osn, valid, hops + 1, sip, ∅))]]
  /* update rreqs by adding (oip, rreqid) */
  [[rreqs := rreqs ∪ {(oip, rreqid)}]]
  (
    [ dip = ip ]      /* this node is the destination node */
      /* update the sqn of ip by setting it to max(sqn(rt, ip), dsn) */
      [[rt := update(rt, (ip, dsn, valid, 0, ip, ∅))]]
      /* unicast a RREP towards oip of the RREQ; next hop is sip */
      unicast(sip, rrep(0, dip, sqn(rt, ip), oip, ip)) . AODV(ip, rt, rreqs, queues)
      ▶ /* If the packet transmission is unsuccessful, a RERR message is generated */
        [[dests := {(rip, rsn) | (rip, rsn, valid, *, sip, *) ∈ rt}]]
        [[pre := ⋃{precs(rt, rip) | (rip, *) ∈ dests}]]
        [[for all (rip, *) ∈ dests : invalidate(rt, rip)]]
        groupcast(pre, rerr(dests, ip)) . AODV(ip, rt, rreqs, queues)
    + [ dip ≠ ip ]      /* this node is not the destination node */
      (
        [ dip ∈ aD(rt) ∧ dsn ≤ sqn(rt, dip) ∧ sqn(rt, dip) ≠ 0 ]      /* valid route to dip that is
        fresh enough */
          /* update rt by adding sip to precs(rt, dip) */
          [[r := addpre(σ_route(rt, dip), {sip}); rt := update(rt, r)]]
```

© NICTA 2011

Process Algebra

- Desired Properties
  - guaranteed broadcast
  - conditional unicast
  - data structure

- Inspired by
  - π- Calculus
  - ω- Calculus
  - (LOTOS)

© NICTA 2011

Structure of WMNs

- User
  - – Network as a “cloud”
- Collection of nodes
  - – connect / disconnect / send / receive
  - – “parallel execution” of nodes
- Nodes
  - – data management
    - data packets, messages, IP addresses ...
  - – message management (avoid blocking)
  - – core management
    - broadcast / unicast / groupcast ...
  - – “parallel execution” of sequential processes

© NICTA 2011

Nodes (Sequential Process Expressions)

- Syntax of sequential process expressions

$$
\begin{aligned}
SP \quad ::= \quad & X(exp_1, \ldots, exp_n) \mid [\varphi]SP \mid [\![\texttt{var} := exp]\!]SP \mid SP + SP \mid \\
& \alpha.SP \mid \textbf{unicast}(dest, ms).SP \blacktriangleright SP \\
\alpha \quad ::= \quad & \textbf{broadcast}(ms) \mid \textbf{groupcast}(dests, ms) \mid \textbf{send}(ms) \mid \\
& \textbf{deliver}(data) \mid \textbf{receive}(\texttt{msg})
\end{aligned}
$$

© NICTA 2011

Structural Operational Semantics I

- internal state determined by expression and valuation

$$
\begin{array}{rcll}
\xi, \mathbf{broadcast}(ms).p & \xrightarrow{\mathbf{broadcast}(\xi(ms))} & \xi, p & \\
\xi, \mathbf{groupcast}(dests, ms).p & \xrightarrow{\mathbf{groupcast}(\xi(dests),\xi(ms))} & \xi, p & \\
\xi, \mathbf{unicast}(dest, ms).p \blacktriangleright q & \xrightarrow{\mathbf{unicast}(\xi(dest),\xi(ms))} & \xi, p & \\
\xi, \mathbf{unicast}(dest, ms).p \blacktriangleright q & \xrightarrow{\neg\mathbf{unicast}(\xi(dest))} & \xi, q & \\
\xi, \mathbf{send}(ms).p & \xrightarrow{\mathbf{send}(\xi(ms))} & \xi, p & \\
\xi, \mathbf{deliver}(data).p & \xrightarrow{\mathbf{deliver}(\xi(data))} & \xi, p & \\
\xi, \mathbf{receive}(\mathtt{msg}).p & \xrightarrow{\mathbf{receive}(m)} & \xi[\mathtt{msg} := m], p & (\forall m \in \mathtt{MSG})
\end{array}
$$

© NICTA 2011

Network

- Node expressions: $M ::= \quad ip : P : R \quad | \quad M \| M$
- Operational Semantics (snippet)

$$\frac{P \xrightarrow{\mathbf{broadcast}(m)} P'}{ip\!:\!P\!:\!R \xrightarrow{R:\mathbf{*cast}(m)} ip\!:\!P'\!:\!R}$$

$$\frac{P \xrightarrow{\mathbf{unicast}(dip,m)} P' \qquad dip \in R}{ip\!:\!P\!:\!R \xrightarrow{\{dip\}:\mathbf{*cast}(m)} ip\!:\!P'\!:\!R} \qquad \frac{P \xrightarrow{\neg\mathbf{unicast}(dip)} P' \qquad dip \notin R}{ip\!:\!P\!:\!R \xrightarrow{\tau} ip\!:\!P'\!:\!R}$$

$$ip\!:\!P\!:\!R \xrightarrow{\mathbf{connect}(ip,ip')} ip\!:\!P\!:\!R \cup \{ip'\}$$

$$ip\!:\!P\!:\!R \xrightarrow{\mathbf{disconnect}(ip,ip')} ip\!:\!P\!:\!R - \{ip'\}$$

© NICTA 2011

A Bit of Theoretical Results

NICTA

- process algebra is blocking (our model is non-blocking)
- process algebra is isomorphic to one without data structure --- a process for every substitution instance
- resulting algebra is in *de Simone* format
(by this strong bisimulation is a congruence)
- both parallel operators are associative
(follows by a meta result of Cranen, Mousavi, Reniers)

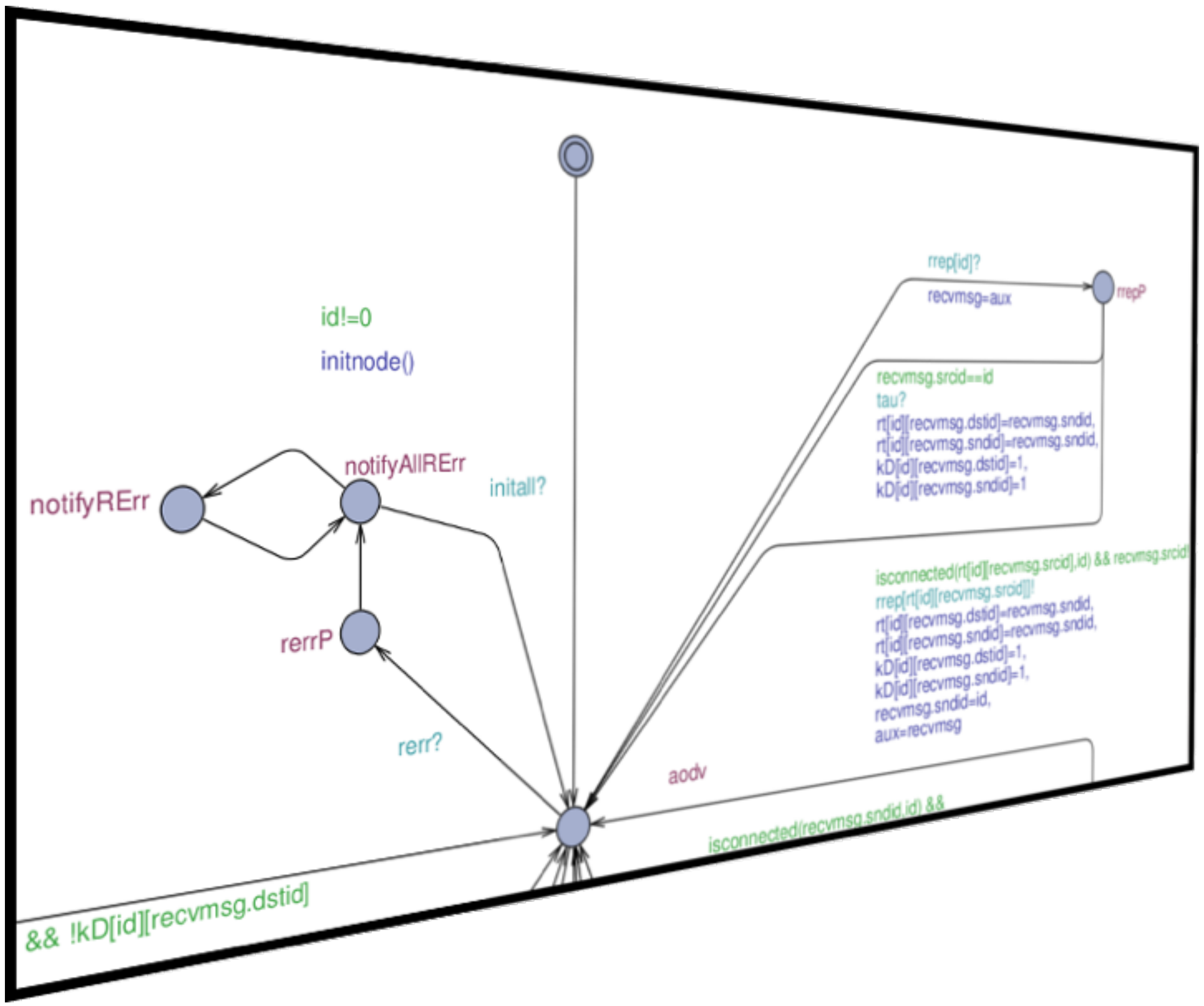
© NICTA 2011

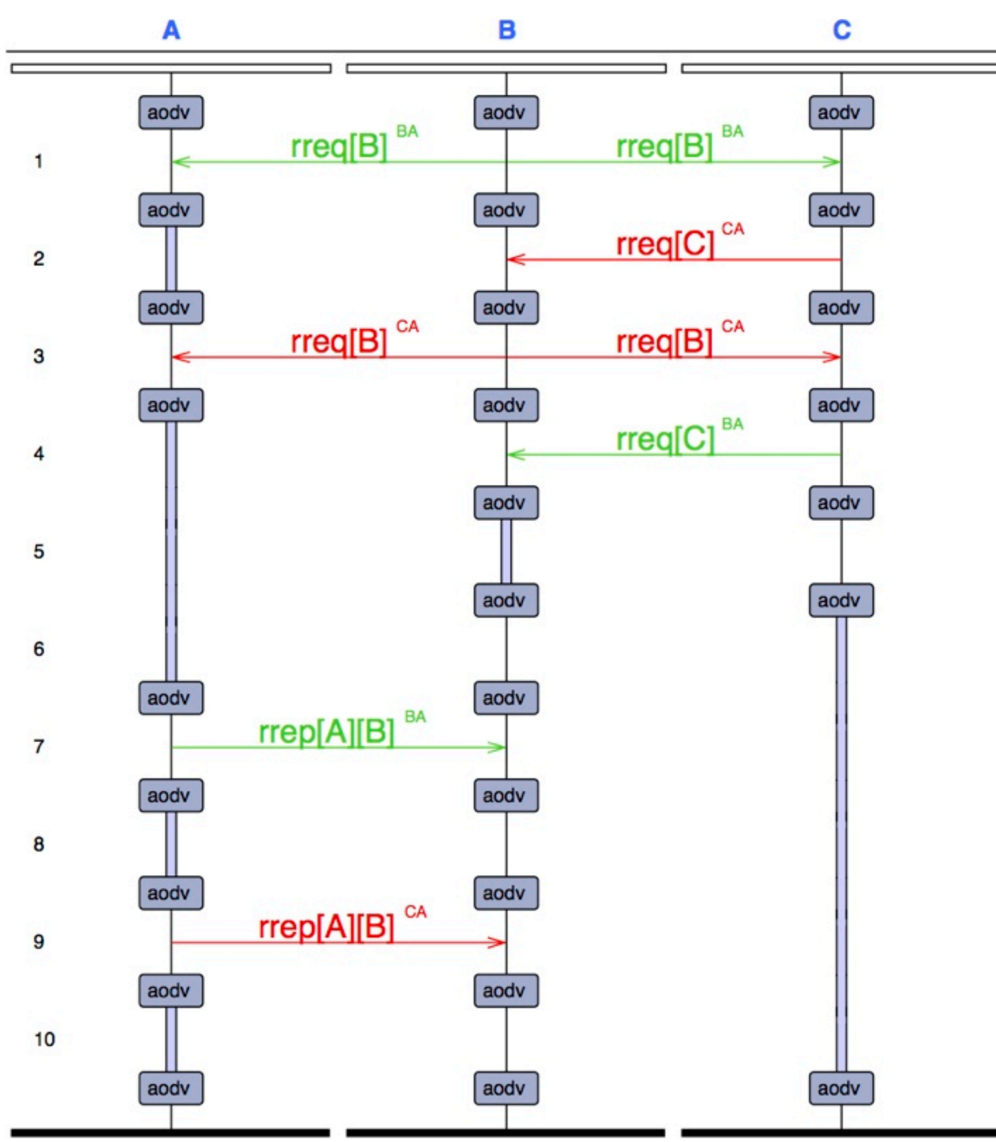
Case Study

- AODV Routing Protocol
- Achievements
  - full concise specification of AODV (RFC 3561) (no time)
  - verified/disproved properties
    - route discovery
    - packet delivery
    - loop freedom
      - first (correct) proof
      - disproved loop freedom for variants of AODV (as implemented in at least open source implementation)
  - found several ambiguities, mistakes, shortcomings
  - found solutions for some limitations

© NICTA 2011

Model Checking





© NICTA 2011

Model Checking

- Model checking routing algorithms
  - executable models
- Complementary to process algebra
  - find bugs and typos in model of process algebra
  - check properties of specification applied to particular topology
  - easy adaption in case of change
  - automatic verification
- Achievements
  - implemented process algebra specification of AODV
  - found/replayed shortcomings

© NICTA 2011

UPPAAL Model Checker

- Well established model checker
- Uses networks of timed automata
- Has been used for protocol verification

- Synchronisation mechanisms
  - binary handshake synchronisation (unicast communication)
  - broadcast synchronisation (broadcast communication)
- Common data structures
  - arrays, structs, ...
  - C-like programming language
- Provides mechanisms for time and probability

© NICTA 2011

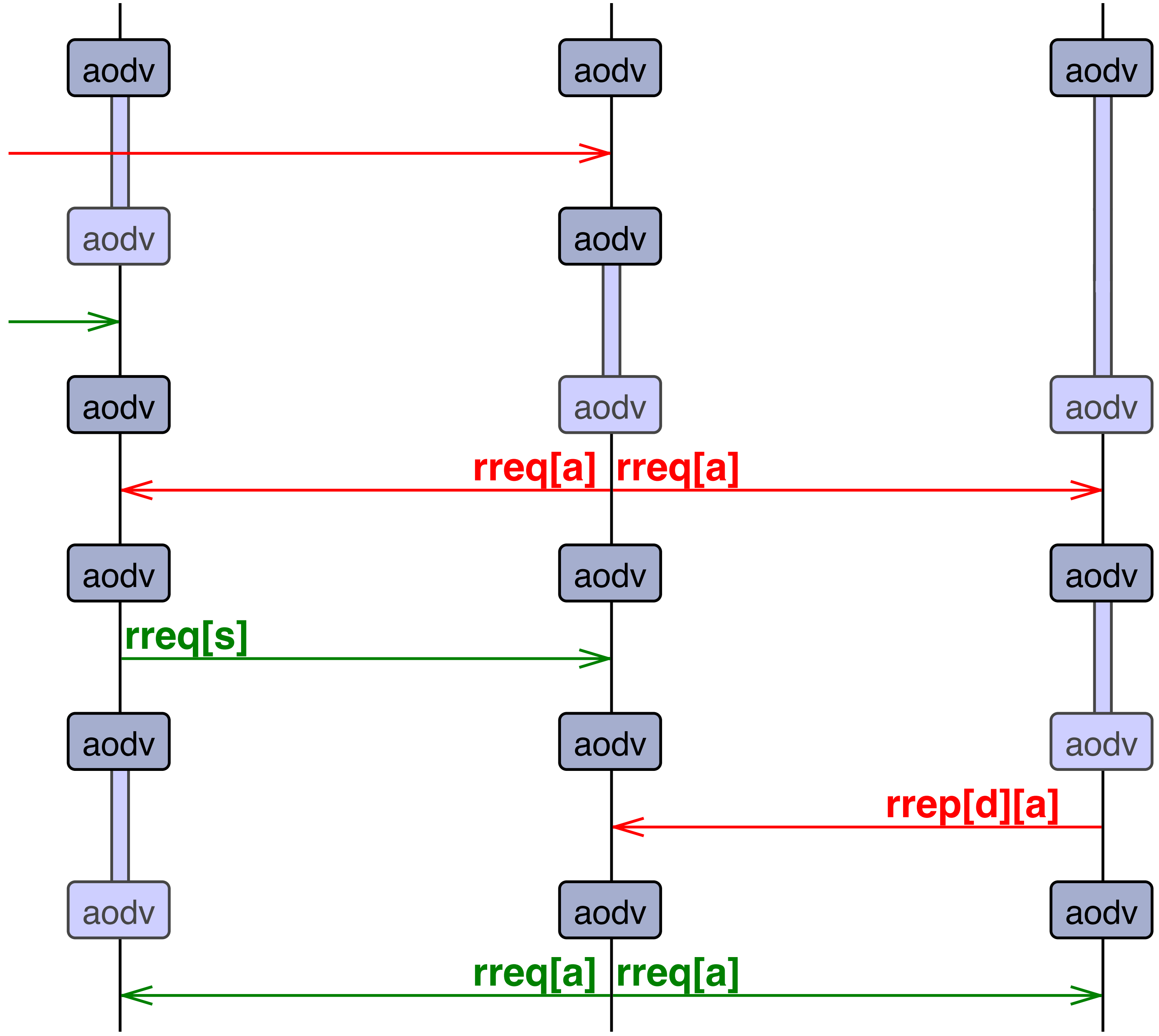
Experiments Set-Up

- Exhaustive search
  - various properties
  - all different topologies up to 5 nodes (one topology change)
  - 2 route discovery processes
  - 17400 scenarios
  - variants of AODV (4 models)

© NICTA 2011

Results: Route Discovery (2004)

- Route discovery fails in a linear 3-node topology



© NICTA 2011

Results: Route Discovery

- exhaustive search
  (potential failure in route discovery)
  - static topology: 47.3%
  - dynamic topology (add link): 42.5%
  - dynamic topology (remove link): 73.7%
- AODV repeats route request
- Other solution: forward route reply

© NICTA 2011

Conclusion/Future Work I

- So far concentrated on AODV
  - well known
  - IETF standard
- Extend formal methods to other protocols
  - OSLR, DYMO, ...
- Add further necessary concepts
  - time
  - probability (links, measurements)
  - define quality of protocols

© NICTA 2011

Conclusion/Future Work II

- Automating process-algebraic proofs
- Advantages
  - verified correctness
  - proof replay for variants
- Isabelle/HOL
  - is it possible within reasonable time?
  - what kind of encoding (deep vs shallow) should follow the process algebraic concepts
  - trace-based proofs, how to argue/store a history
  - ...

© NICTA 2011

NICTA

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