Towards an Algebra of Routing Tables

Peter Höfner and Annabelle McIver

NICTA
MACQUARIE UNIVERSITY
UNSW THE UNIVERSITY OF NEW SOUTH WALES SYDNEY · CANBERRA · AUSTRALIA

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

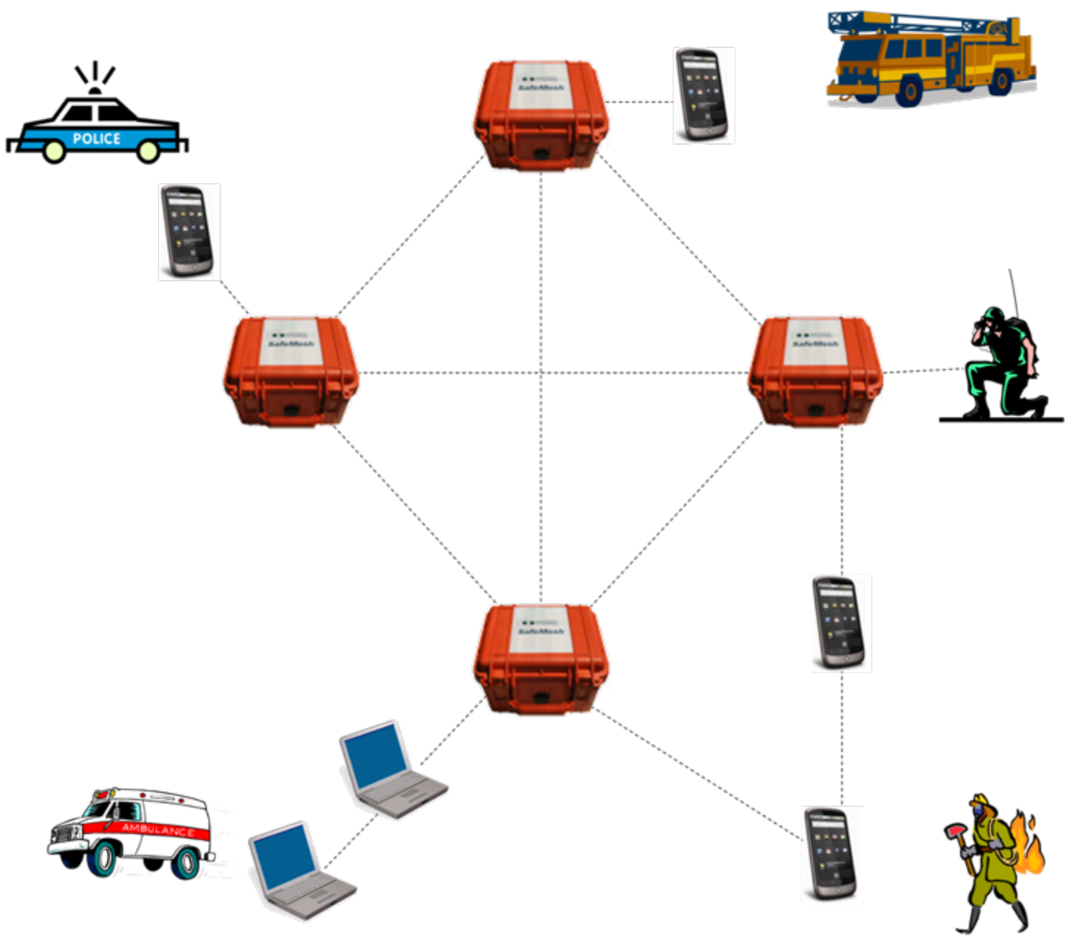
NICTA Members
ANU THE AUSTRALIAN NATIONAL UNIVERSITY
UNSW THE UNIVERSITY OF NEW SOUTH WALES
NSW Department of State and Regional Development

Victoria The Place To Be
THE UNIVERSITY OF MELBOURNE
The University of Sydney
Queensland Government
Griffith UNIVERSITY
QUT Queensland University of Technology
THE UNIVERSITY OF QUEENSLAND AUSTRALIA
NICTA Partners

What is the Problem?

NICTA

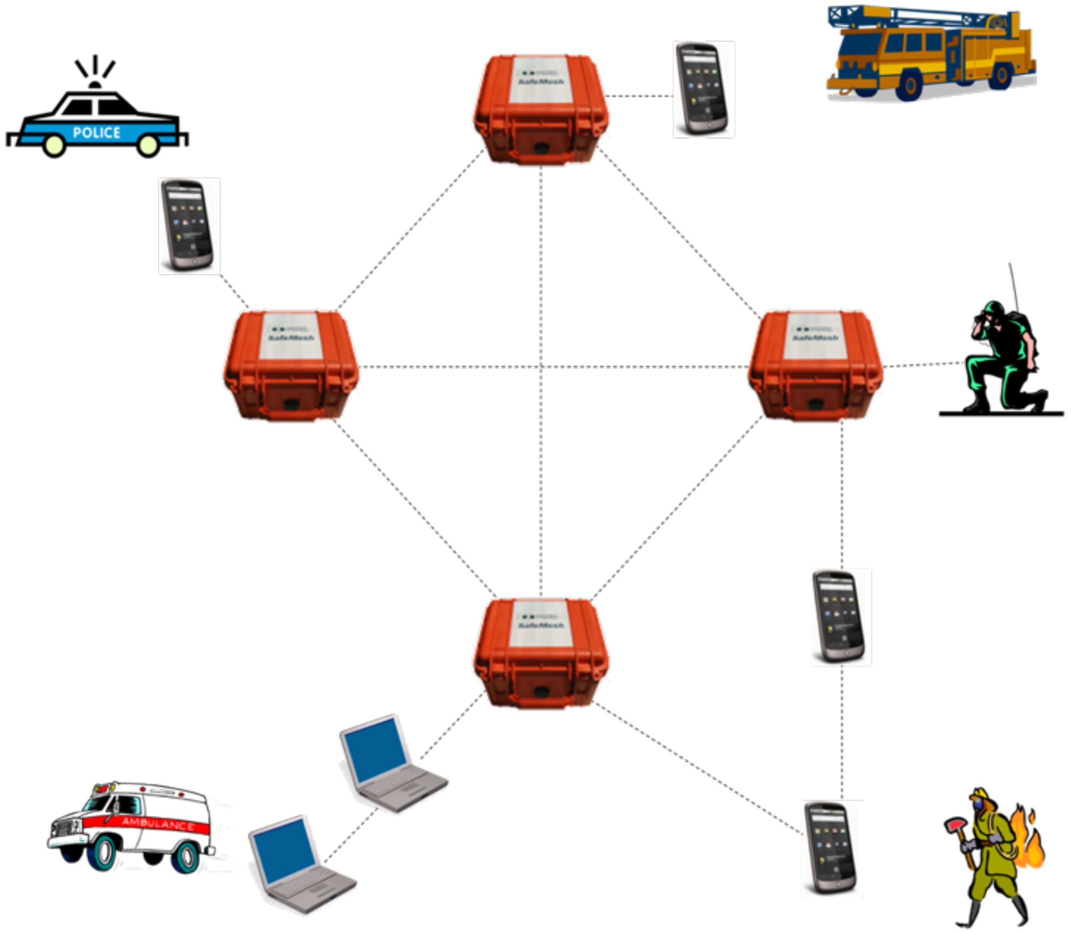
- Wireless Mesh Networks
  - key advantage: no backhaul wiring required
  - quick and low cost deployment
- Applications
  - public safety (e.g. CCTV)
  - emergencies (e.g. earthquakes)
  - mobile phone services
  - transportation
  - mining
  - military actions/counter terrorism
  - ...



© NICTA 2011

What is the Problem?

- WMNs promise to be fully
  – self-configuring
  – self-healing
  – self-optimising

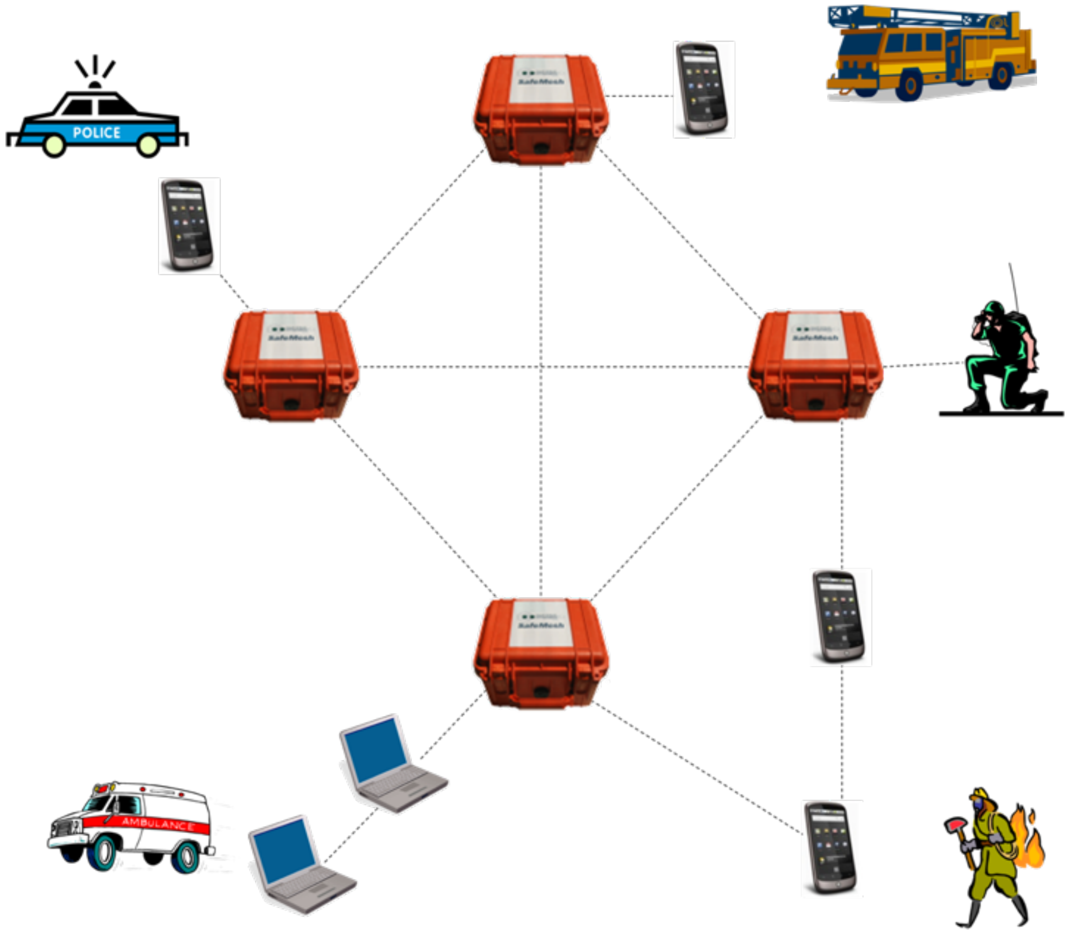


© NICTA 2011

What is the Problem?

NICTA

- WMNs promise to be fully
  - self-configuring
  - self-healing
  - self-optimising
- **THAT IS NOT TRUE** (in reality)
- Limitations in reliability and performance
- Limitations confirmed by
  - end users (e.g. police)
  - own experiments
    - Cisco, Motorola, Firetide, ...
  - industry



© NICTA 2011

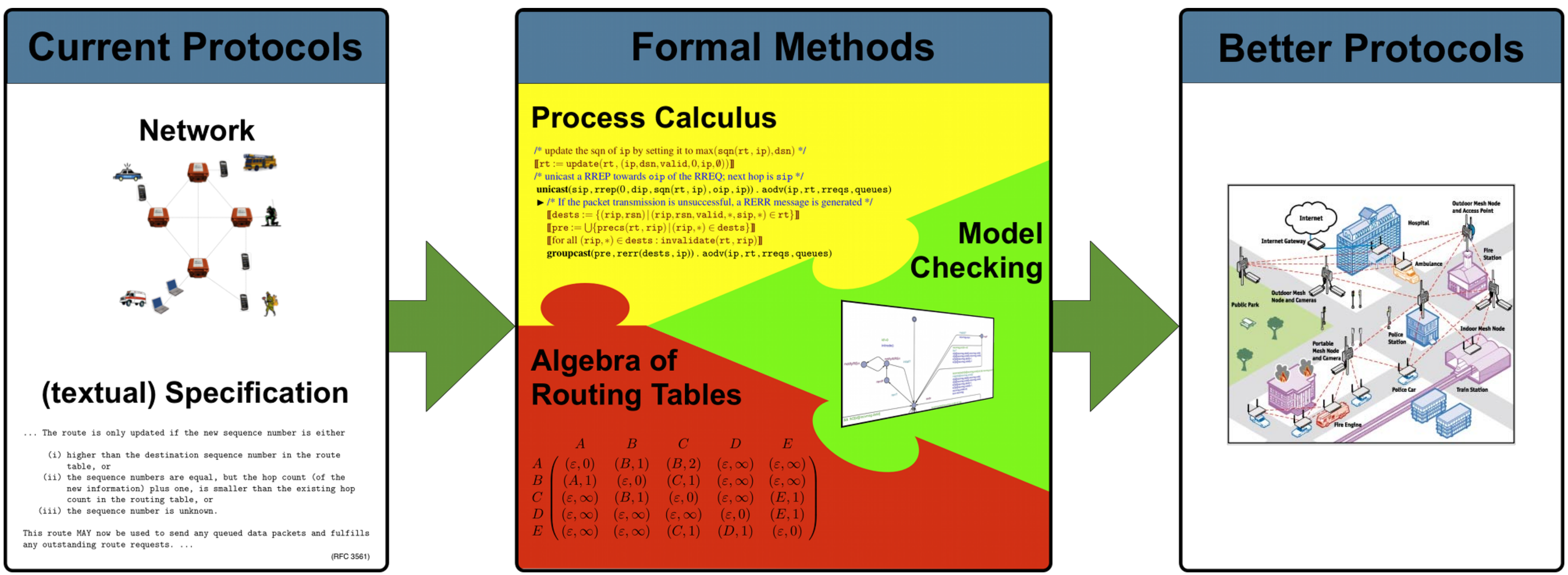
Formal Methods for Mesh Networks

- 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”

© NICTA 2011

Formal Methods for Mesh Networks

- Main Methods used so far
  - process algebra
  - model checking
  - **algebra of routing tables**



© NICTA 2011

Ad Hoc On-Demand Distance Vector Protocol

NICTA

- 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)
  - (Hello messages)

- Information at nodes
  - own IP address
  - a local sequence number (freshness/timer)
  - a routing table
    - local knowledge
    - entries: $(\texttt{dip}\,,\,\texttt{dsn}\,,\,\texttt{val}\,,\,\texttt{hops}\,,\,\texttt{nhip}\,,\,\texttt{pre})$

© NICTA 2011

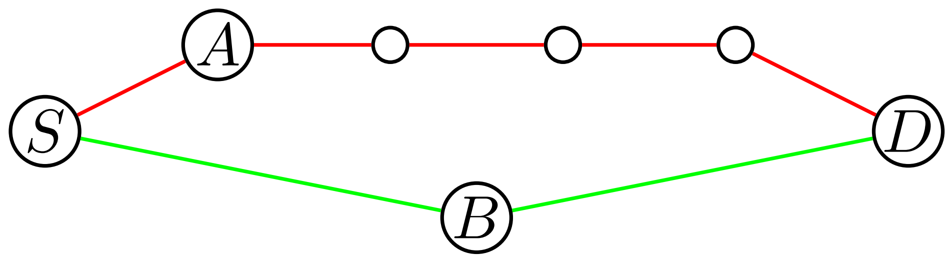
Routing Algebra – Elements, Operators

- Routing table entries (no sequence number so far)

$$(\texttt{nhip}\,,\,\texttt{hops})$$

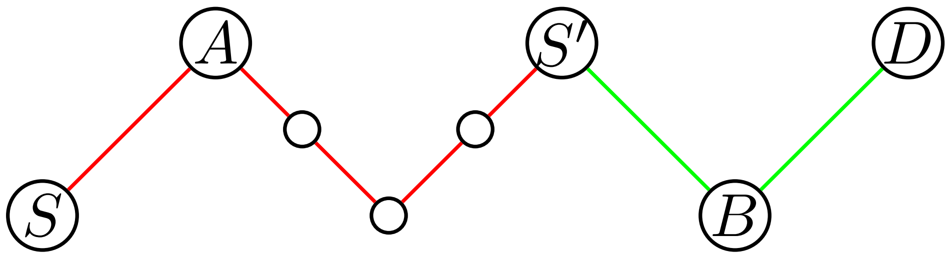
- Choice:

$$(A,5)+(B,2)=(B,2)$$



- Multiplication (destination and source must coincide)

$$(A,5)\cdot(B,2)=(A,7)$$



- Special symbols: $(\_,0)$, $(\_,\infty)$
- Idea: back to Backhouse, Carré, Griffin, Sobrinho

© NICTA 2011

Routing Algebra – Elements, Operators

- Matrices over routing table entries

$$
\begin{array}{c} \\ A \\ B \\ C \\ D \\ \vdots \end{array}
\begin{array}{ccccc} A & B & C & D & \dots \end{array}
$$

$$
\begin{array}{c} A \\ B \\ C \\ D \\ \vdots \end{array}
\begin{pmatrix}
(\_, 0) & (B, 1) & (B, 2) & (\_, \infty) & \\
(A, 1) & (\_, 0) & (C, 1) & (\_, \infty) & \dots \\
(\_, \infty) & (B, 1) & (\_, 0) & (\_, \infty) & \\
(\_, \infty) & (\_, \infty) & (\_, \infty) & (\_, 0) & \\
 & \vdots & & & \ddots
\end{pmatrix}
\text{routing table of } A
$$

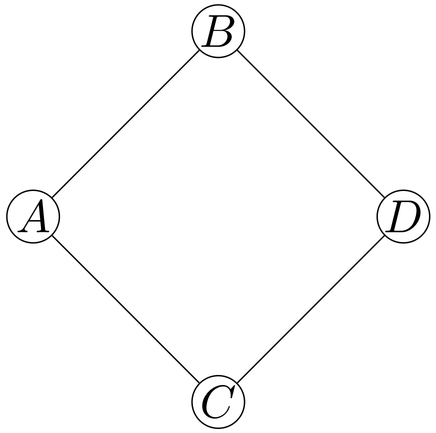
“routes” to B

- standard matrix operations
- further abstraction possible
  (semirings, test, domain, modules ...)

© NICTA 2011

Example

- A route request is broadcast



$$
\begin{pmatrix}
(\_,0) & (B,1) & (C,1) & (\_,\infty) \\
(A,1) & (\_,0) & (\_,\infty) & (D,1) \\
(A,1) & (\_,\infty) & (\_,0) & (D,1) \\
(\_,\infty) & (B,1) & (C,1) & (\_,0)
\end{pmatrix}
\bullet
\begin{pmatrix}
(\_,0) & (\_,\infty) & (\_,\infty) & (\_,\infty) \\
(\_,\infty) & (\_,\infty) & (\_,\infty) & (\_,\infty) \\
(\_,\infty) & (\_,\infty) & (\_,\infty) & (\_,\infty) \\
(\_,\infty) & (\_,\infty) & (\_,\infty) & (\_,\infty)
\end{pmatrix}
\bullet
\begin{pmatrix}
(\_,0) & (B,1) & (\_,\infty) & (\_,\infty) \\
(\mathbf{D},\mathbf{3}) & (\_,0) & (\_,\infty) & (\_,\infty) \\
(A,1) & (\_,\infty) & (\_,0) & (D,1) \\
(C,2) & (\_,\infty) & (C,1) & (\_,0)
\end{pmatrix}
$$

topology   sender   routing table

$$
= \begin{pmatrix}
(\_,0) & (B,1) & (\_,\infty) & (\_,\infty) \\
(\mathbf{A},\mathbf{1}) & (\_,0) & (\_,\infty) & (\_,\infty) \\
(A,1) & (\_,\infty) & (\_,0) & (D,1) \\
(C,2) & (\_,\infty) & (C,1) & (\_,0)
\end{pmatrix}
$$

updated routing table

© NICTA 2011

Further Abstraction

- Interpret matrix as an arbitrary element of a semiring
- Kleene algebra allows iteration,
- (Co)Domain and tests model projections

© NICTA 2011

Sent Messages

- sending messages

$$a + p \cdot b \cdot q \cdot (1 + c)$$

- by distributivity

$$a \;\;+\;\; p \cdot b \cdot q \;\;+\;\; p \cdot b \cdot q \cdot c$$

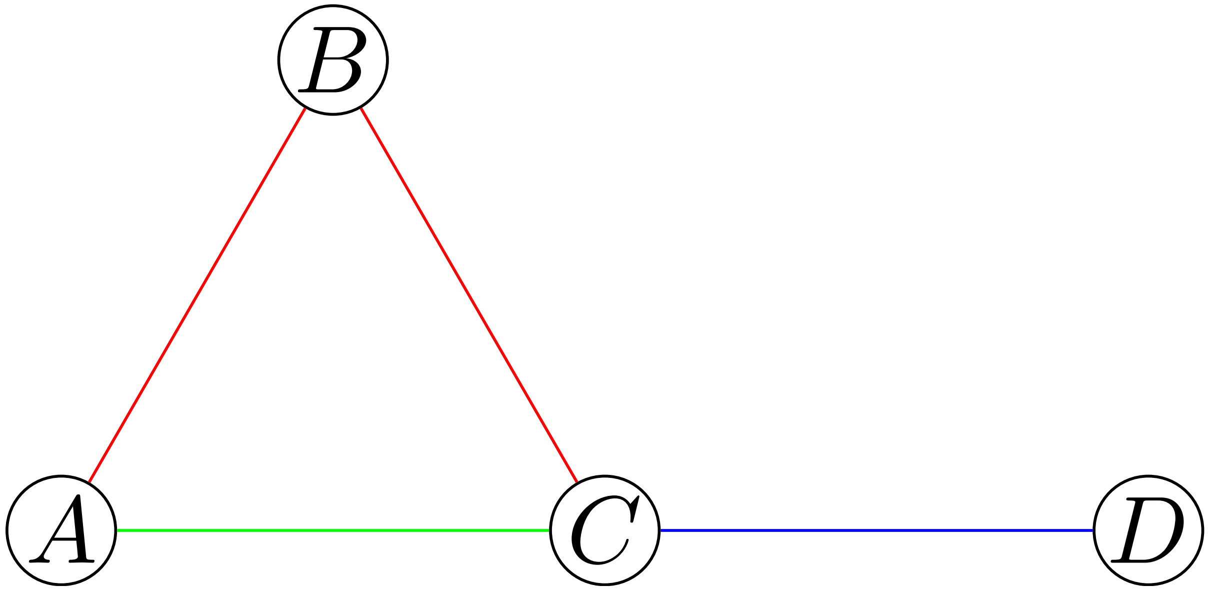
  snapshot, 1-hop connection learnt, content sent
- broadcast, unicast, groupcast are the same (modelled by different topologies)
- Kleene star models flooding the network (modal operators terminate flooding)

- QUESTION: Can unicast modelled purely algebraically?

© NICTA 2011

Lost and Found

- Adding sequence numbers



$$
\begin{aligned}
r \cdot b &= (B,2,5)\cdot(D,1,10) &= (B\cdot D, 2+1, \max(5,10)) &= (B,3,10)\\
g \cdot b &= (C,1,3)\cdot(D,1,10) &= (C\cdot D, 1+1, \max(3,10)) &= (C,2,10)
\end{aligned}
$$

$$r\cdot b + g\cdot b \neq (r+g)\cdot b$$

© NICTA 2011

Lost and Found

- Restrict multiplication
  - partial defined operation
  - only topologies allowed on the left-hand side
  - Kleene star has to be adapted

- Module like structure
  (scalars are subalgebra)

© NICTA 2011

Miscellaneous

- Ad hoc prototype in Haskell
- Theorems at algebraic level proven with Prover9
- Use Isabelle/HOL to switch between model and algebra

© NICTA 2011

Future Work

NICTA

- Include sequence numbers
- Important properties loop freedom, route correctness
- Improvement/refinement
- So far concentrated on AODV
  - well known, IETF standard, known limitations
- Extend formal methods to other protocols
  - OSLR, DYMO, DLR ...
- Add further necessary concepts
  - time, probability

© NICTA 2011

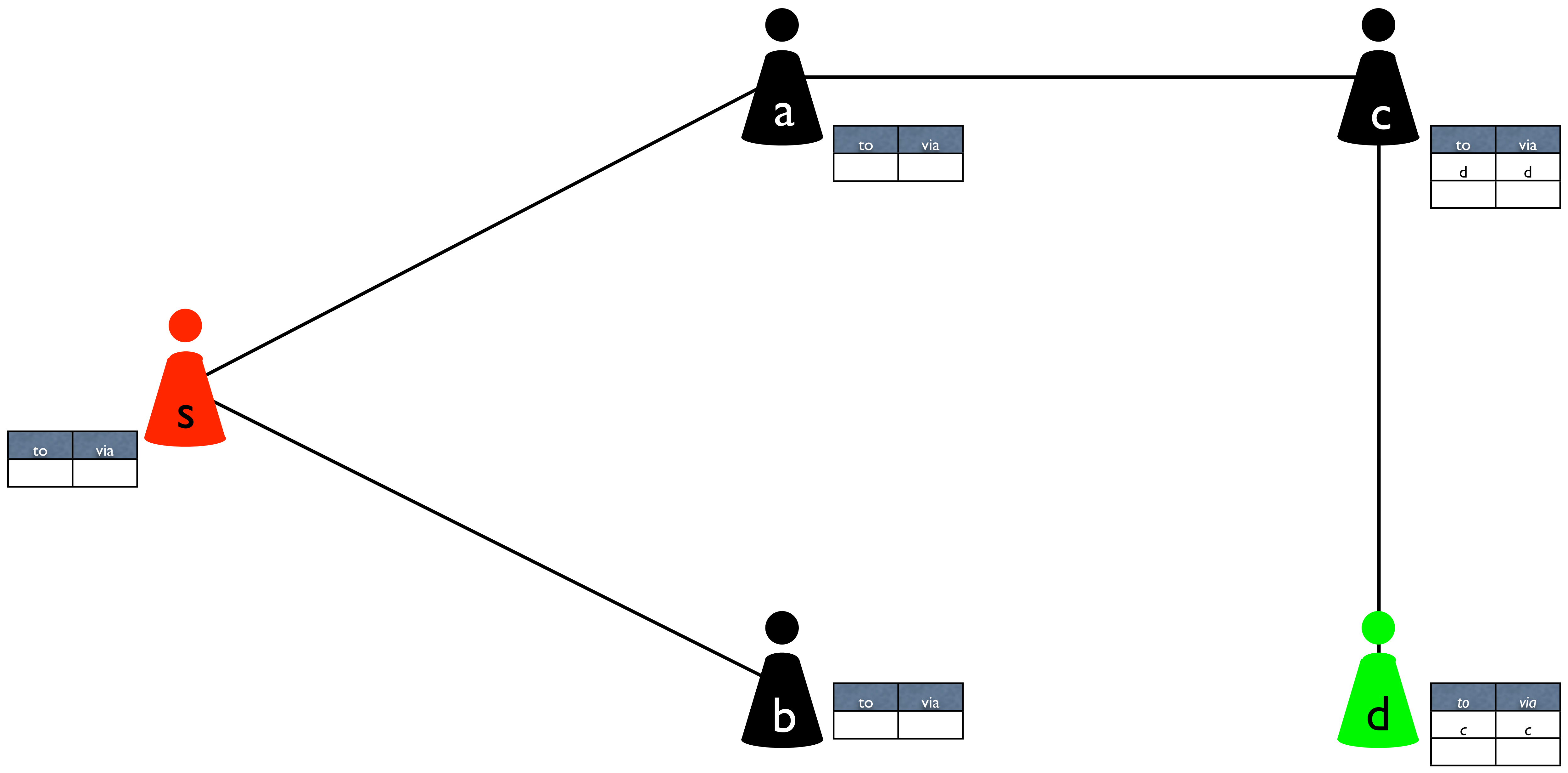
From **imagination** to **impact**

NICTA

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

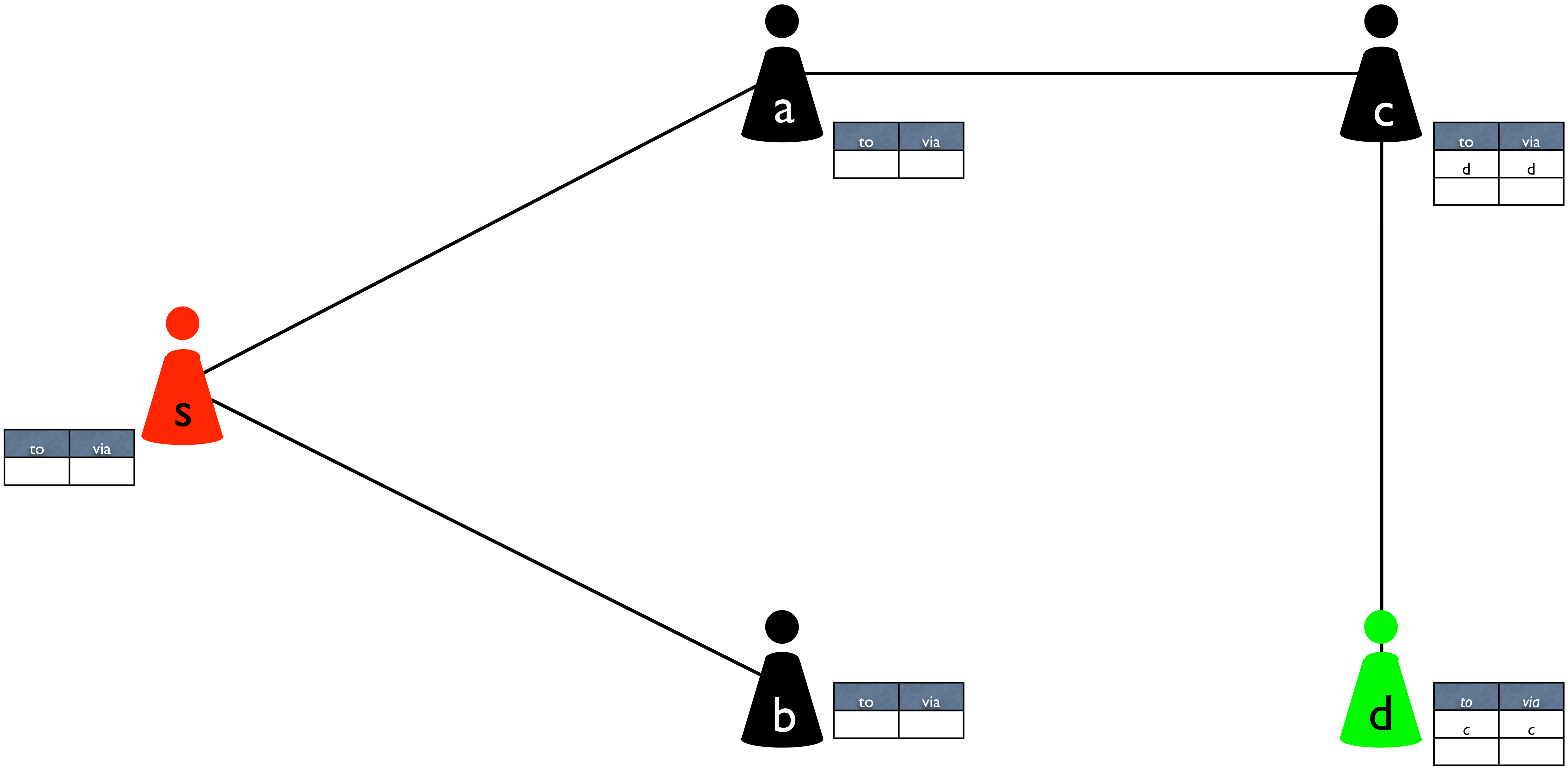
AODV – An Example

NICTA

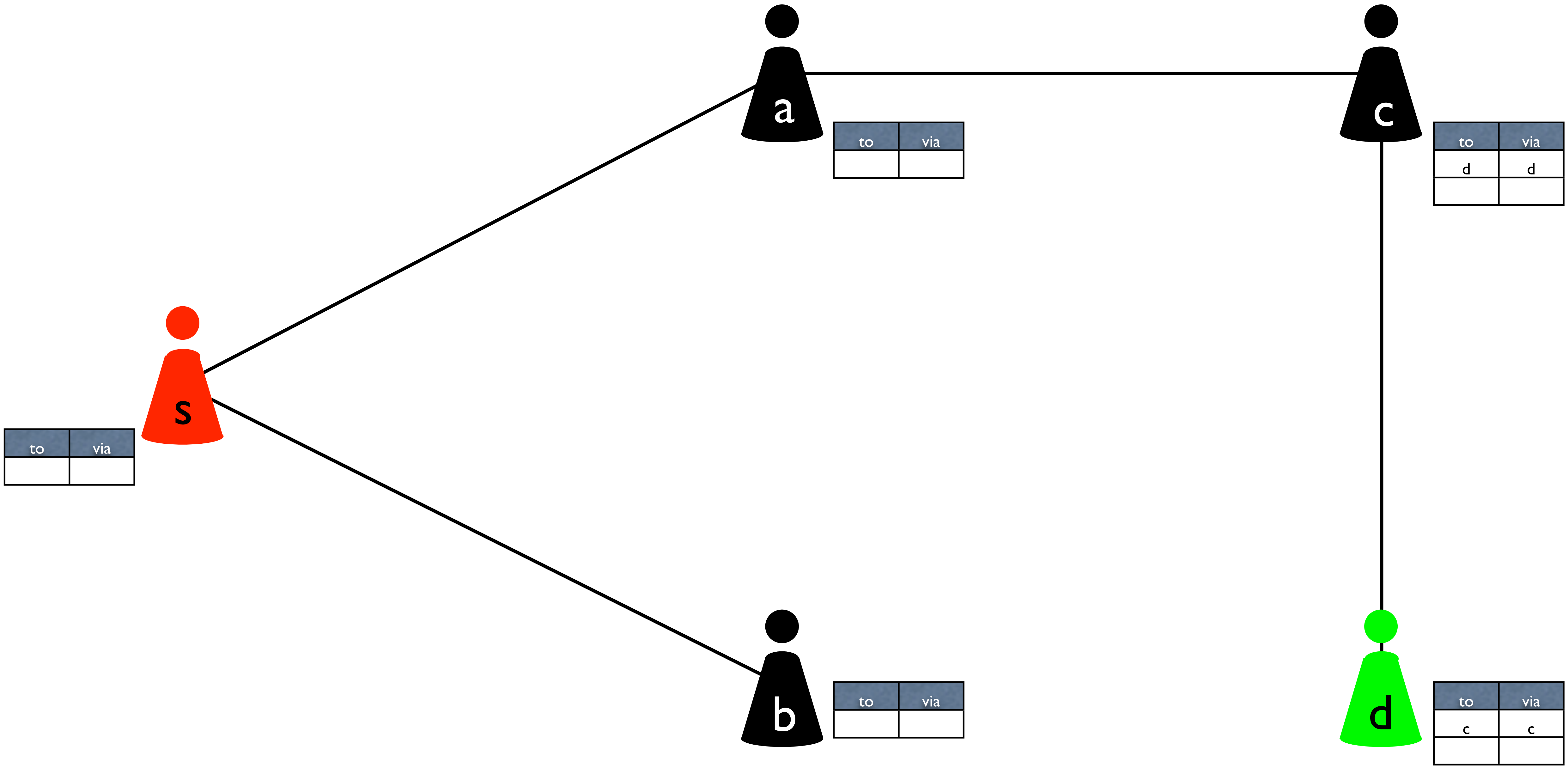


s is looking for a route to d

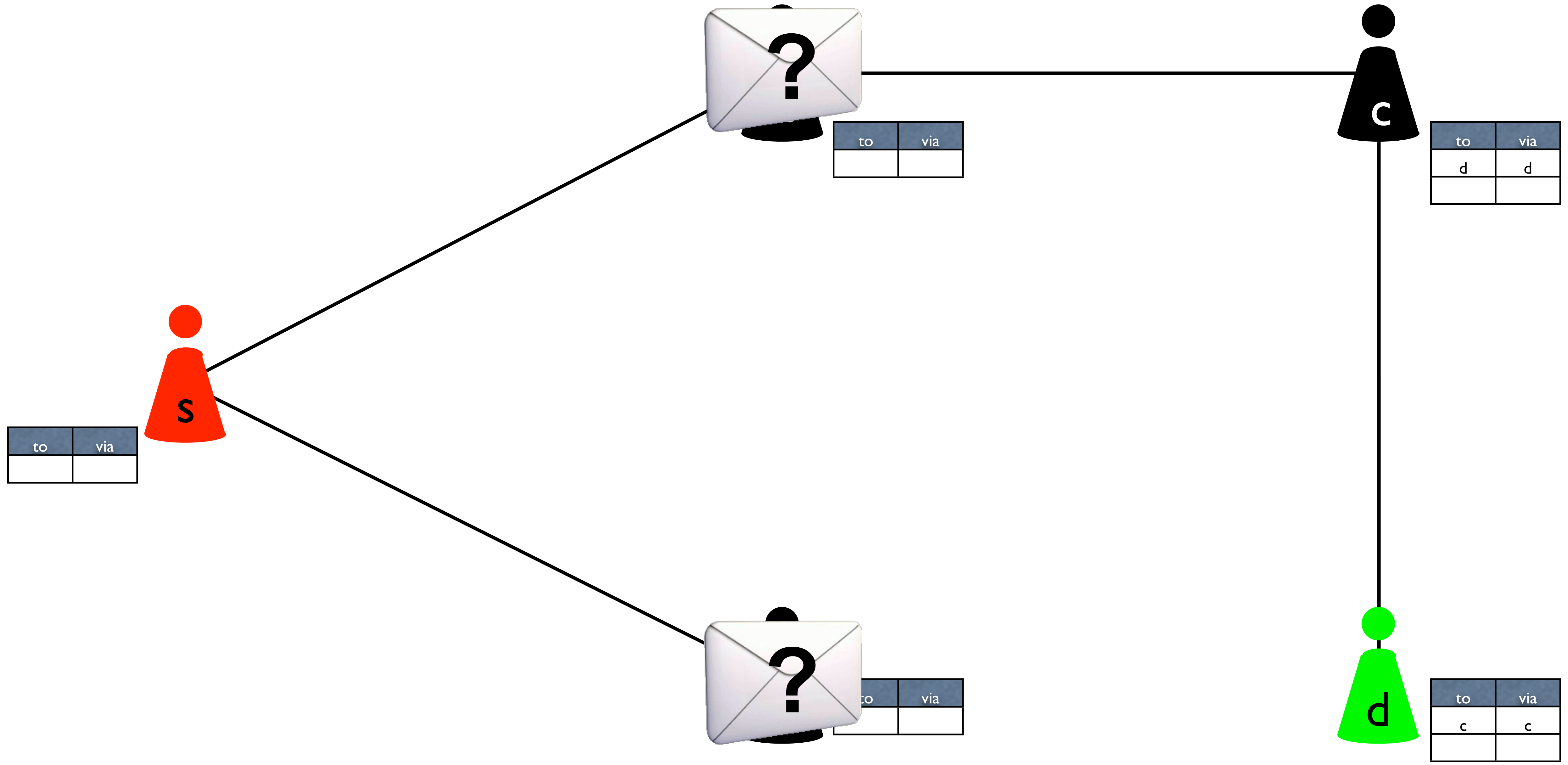
AODV – An Example



AODV - An Example



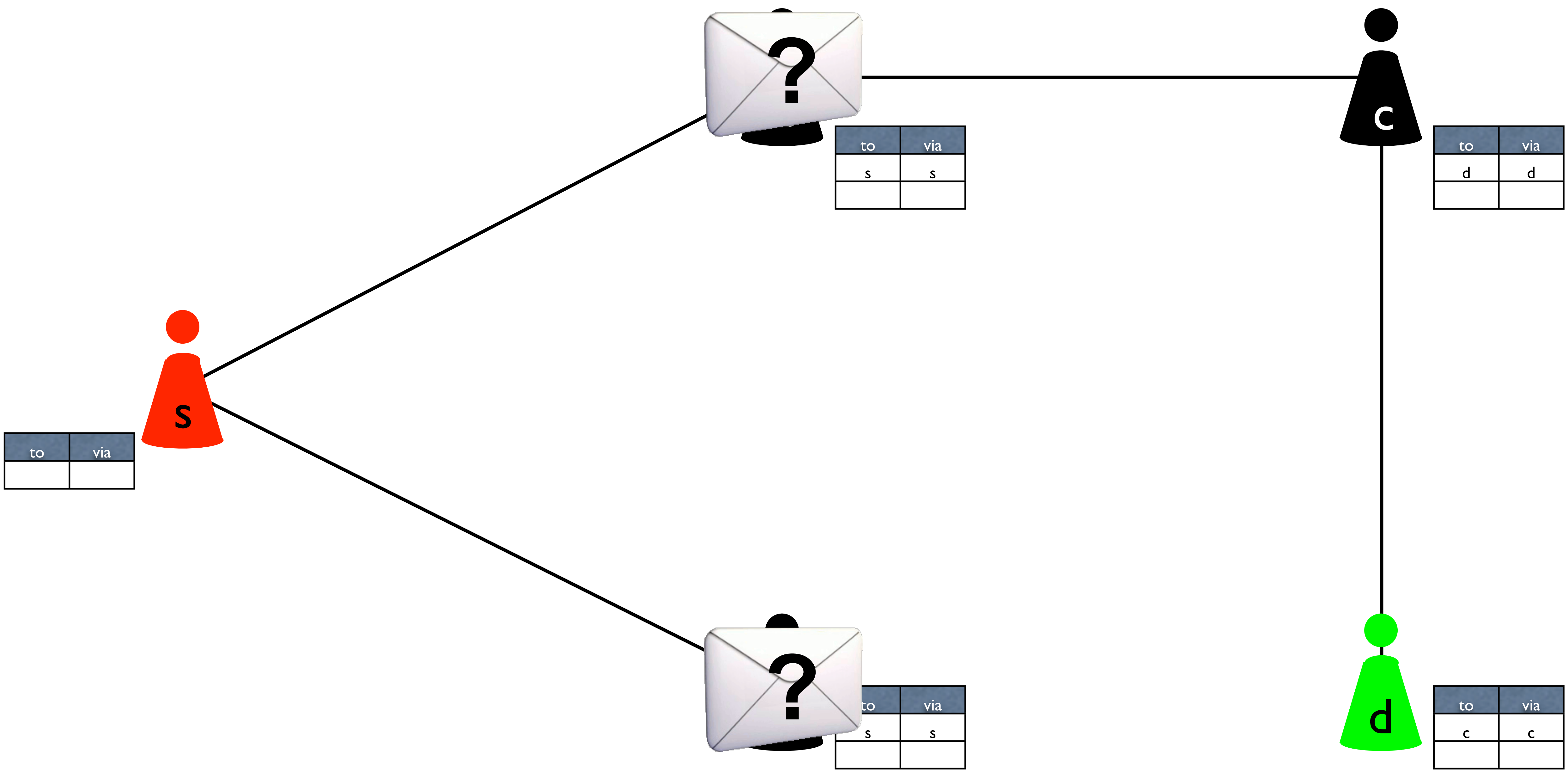
AODV – An Example



s broadcasts a route request

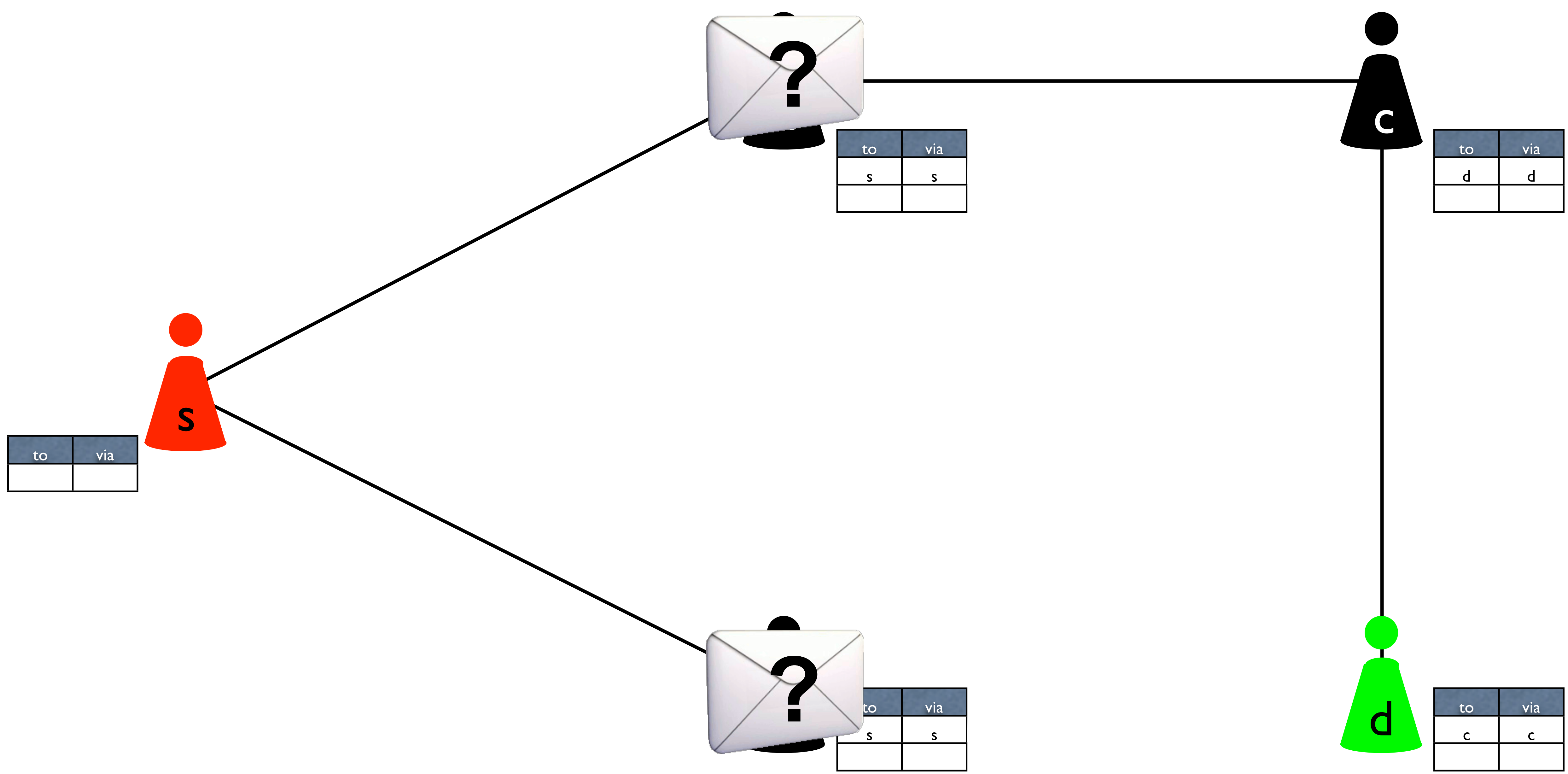
AODV – An Example

NICTA

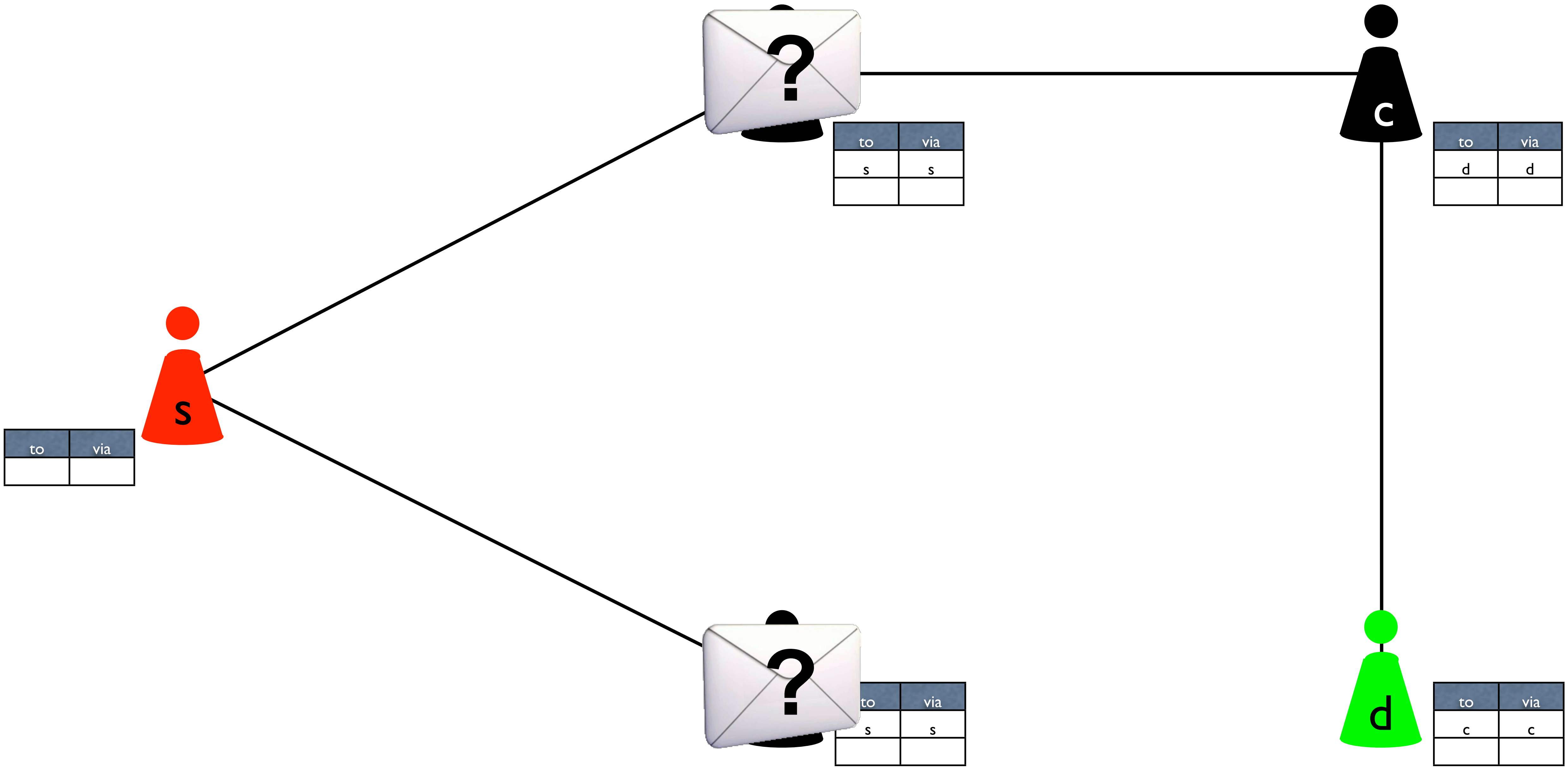


s broadcasts a route request

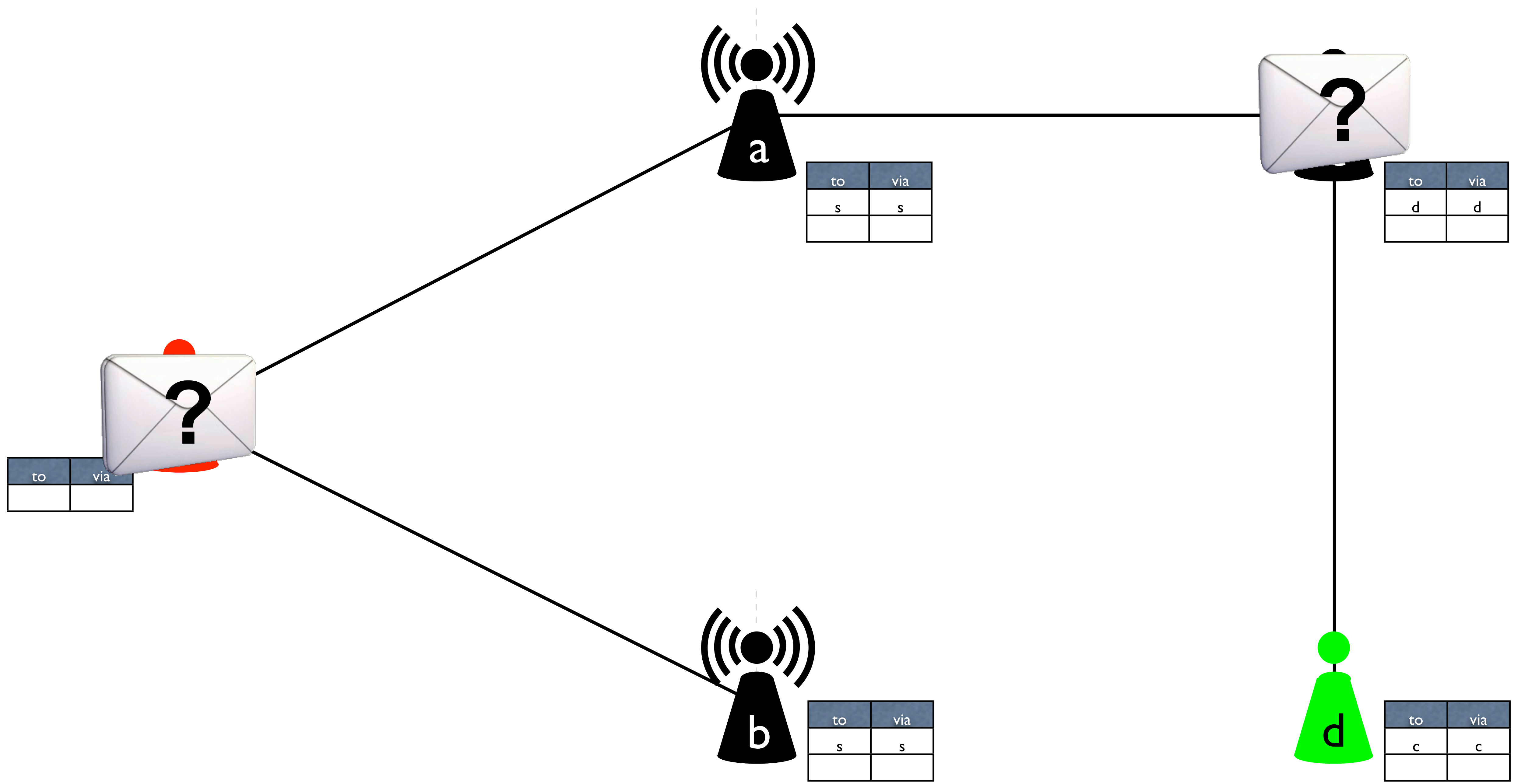
AODV - An Example



AODV – An Example

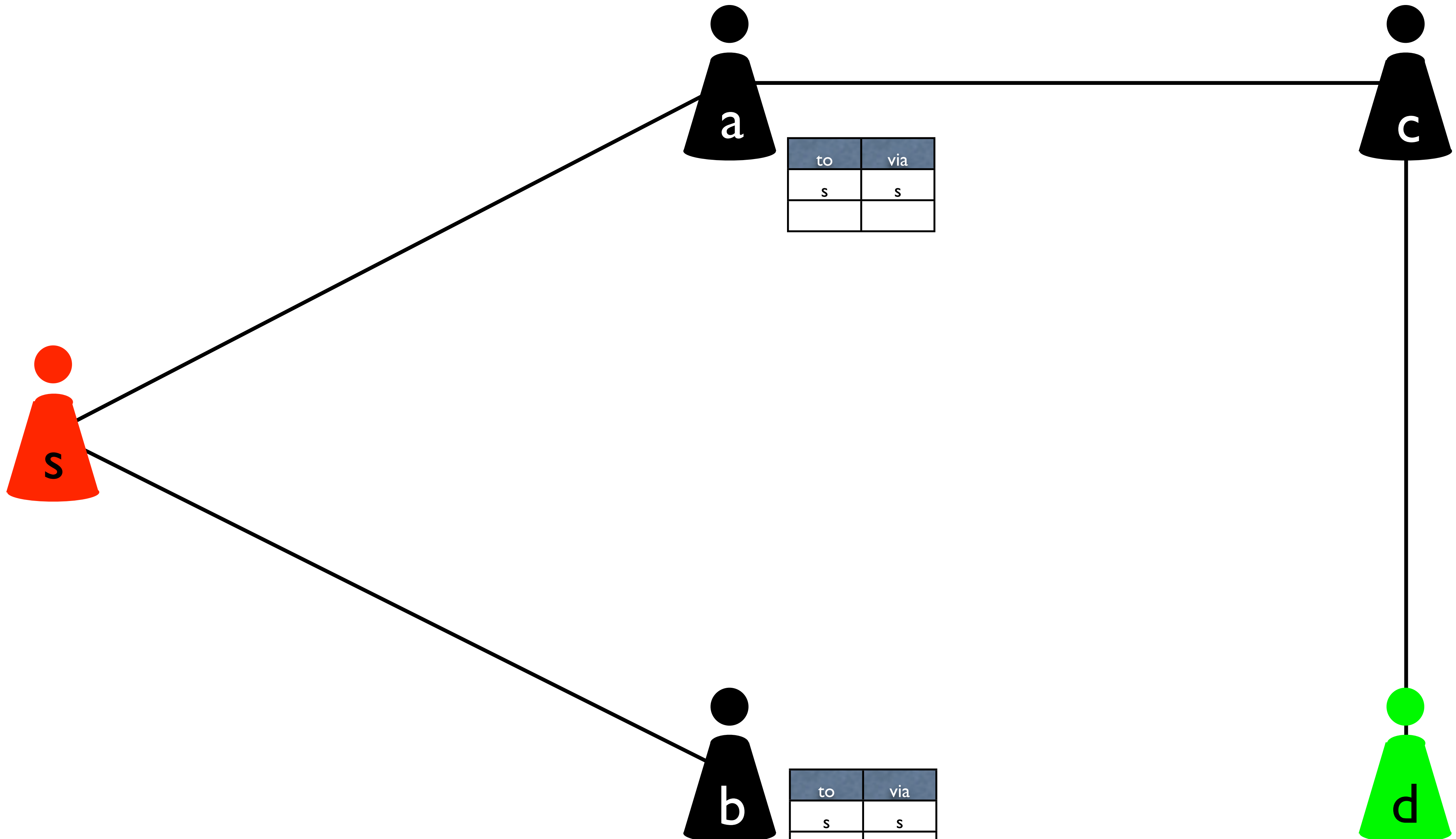


AODV – An Example



a,b forward the route request

AODV – An Example



to	via
a	a
b	b

to	via
s	s

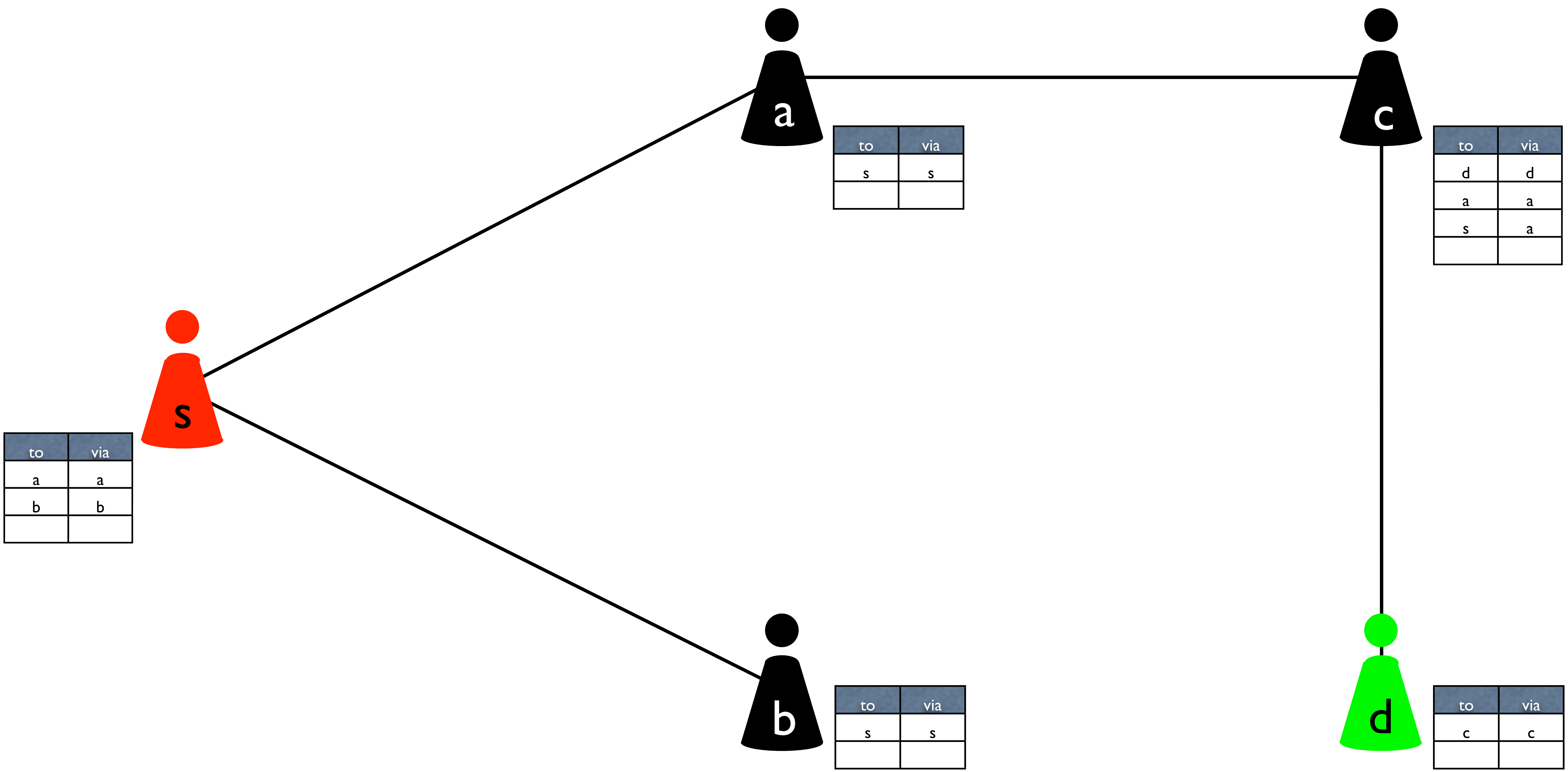
to	via
d	d
a	a
s	a

to	via
s	s

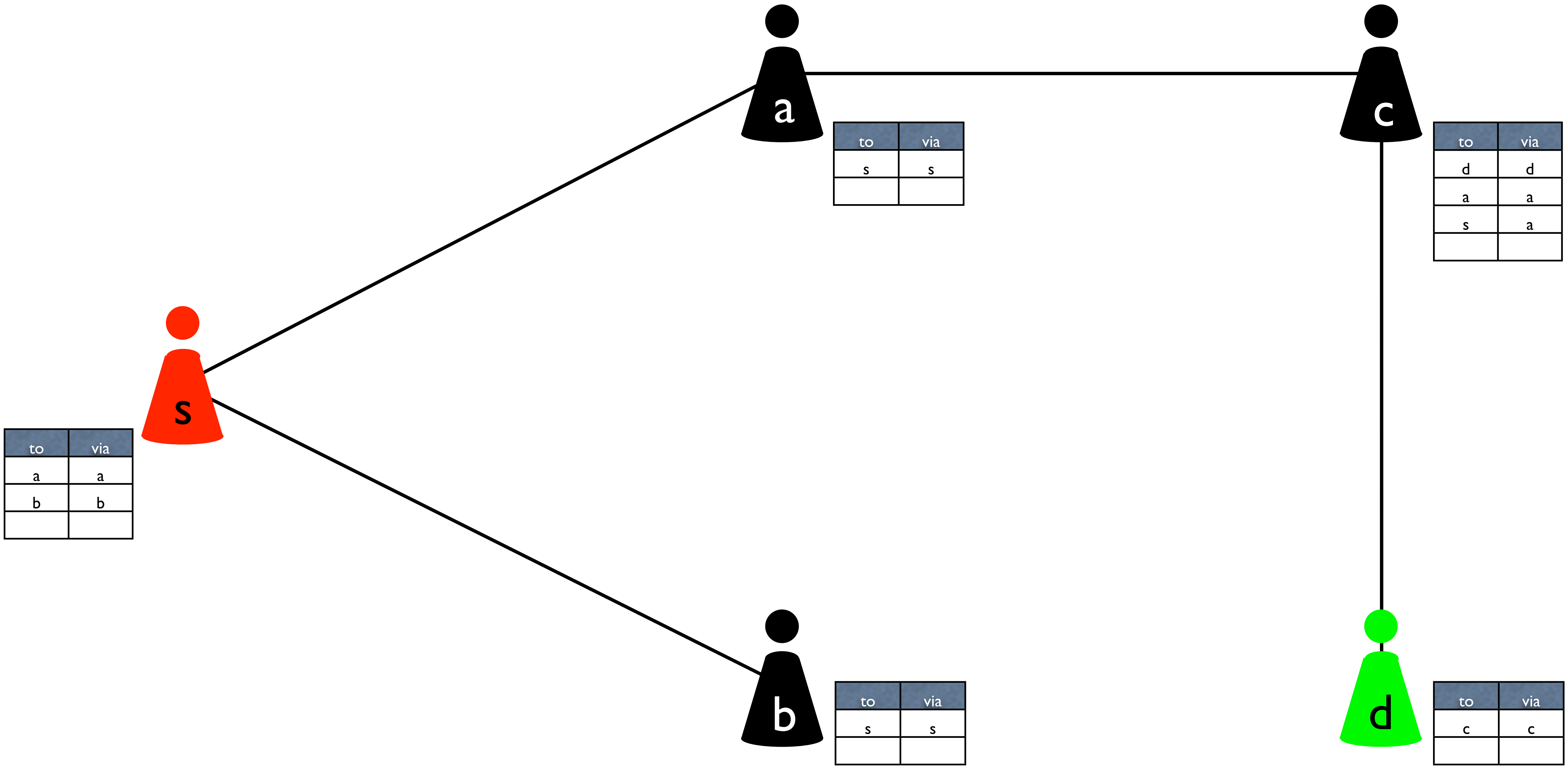
to	via
c	c

a,b forward the route request

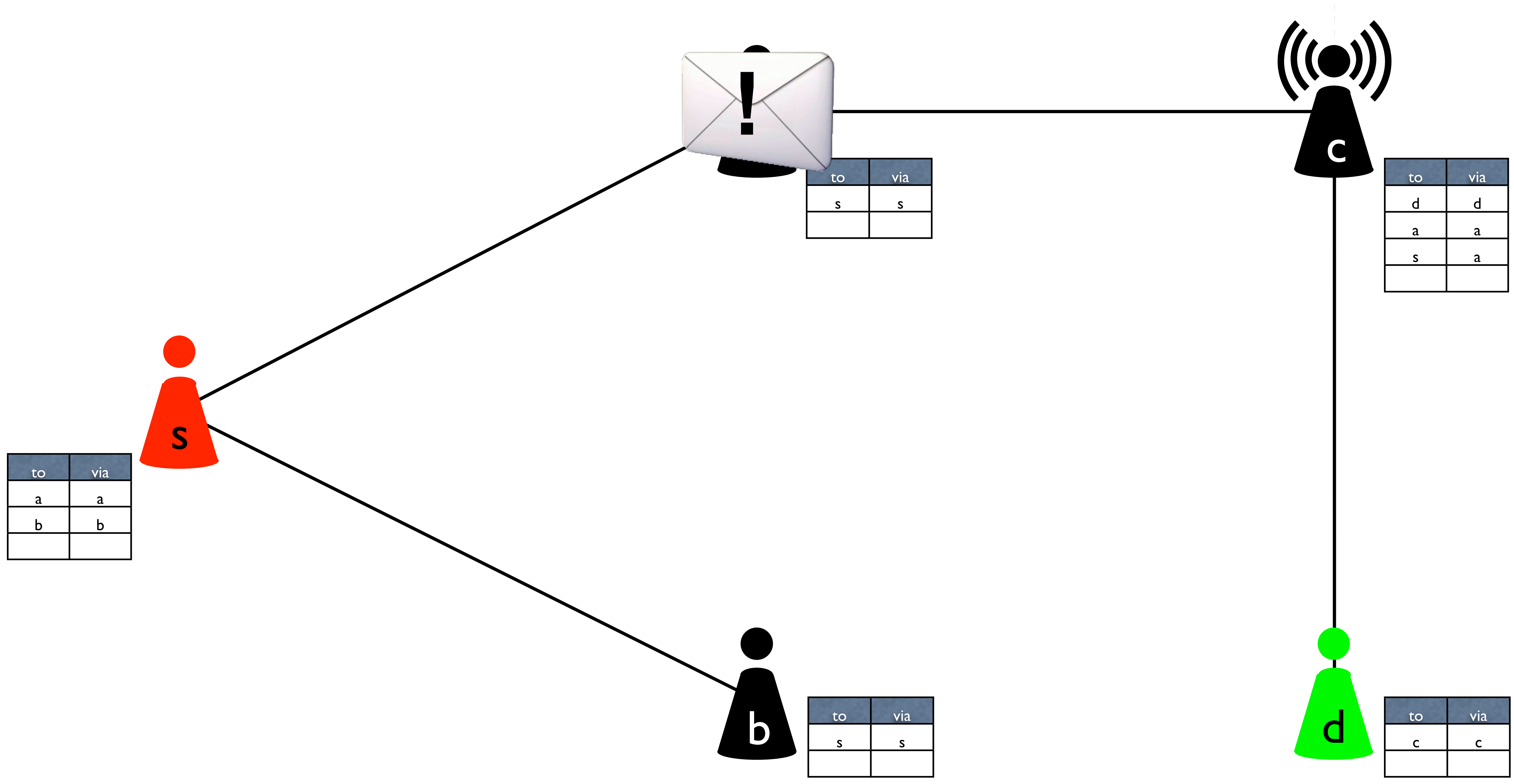
AODV – An Example



AODV – An Example

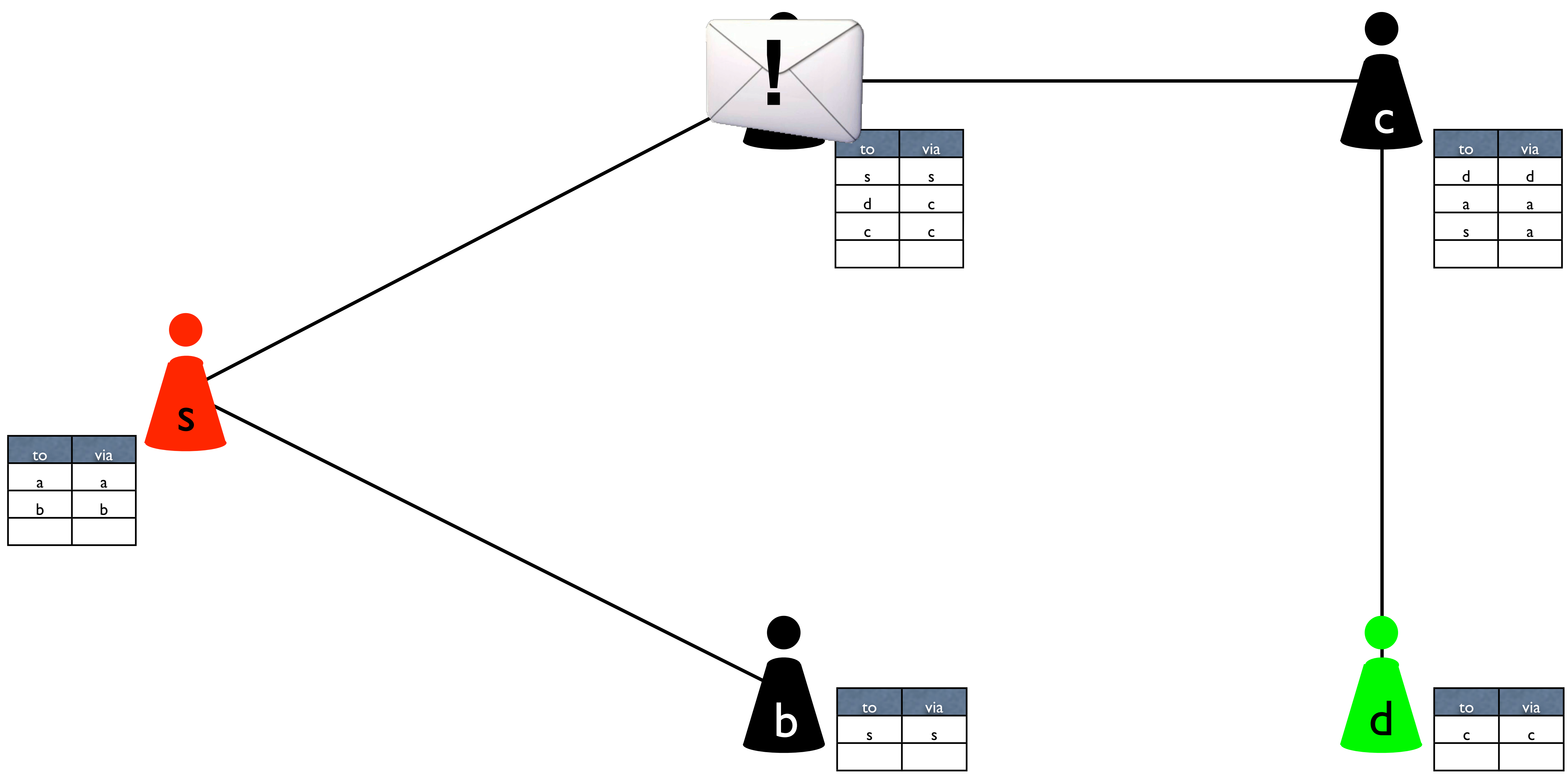


AODV – An Example



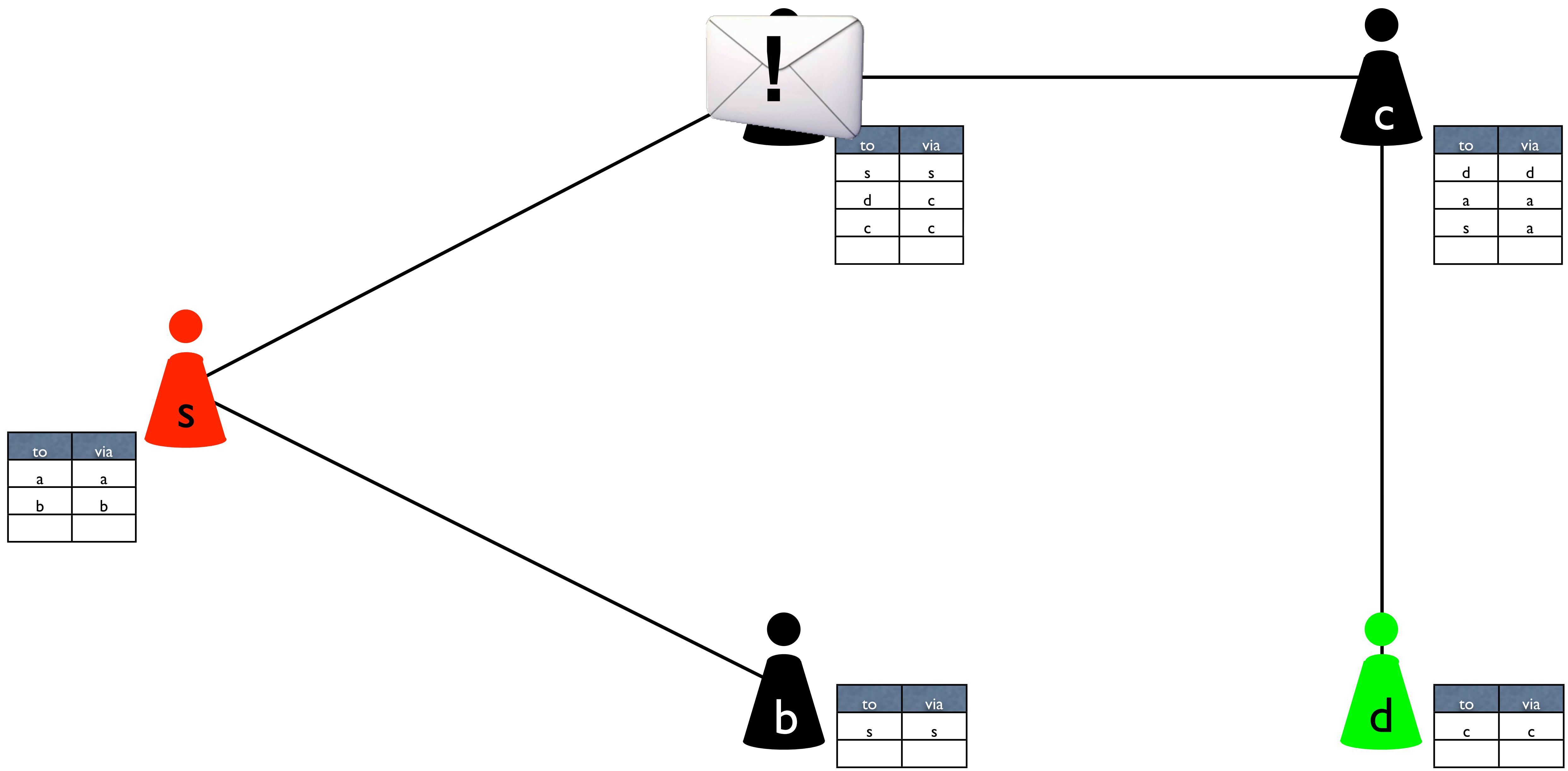
c has information about d
c answers route request and sends reply

AODV – An Example

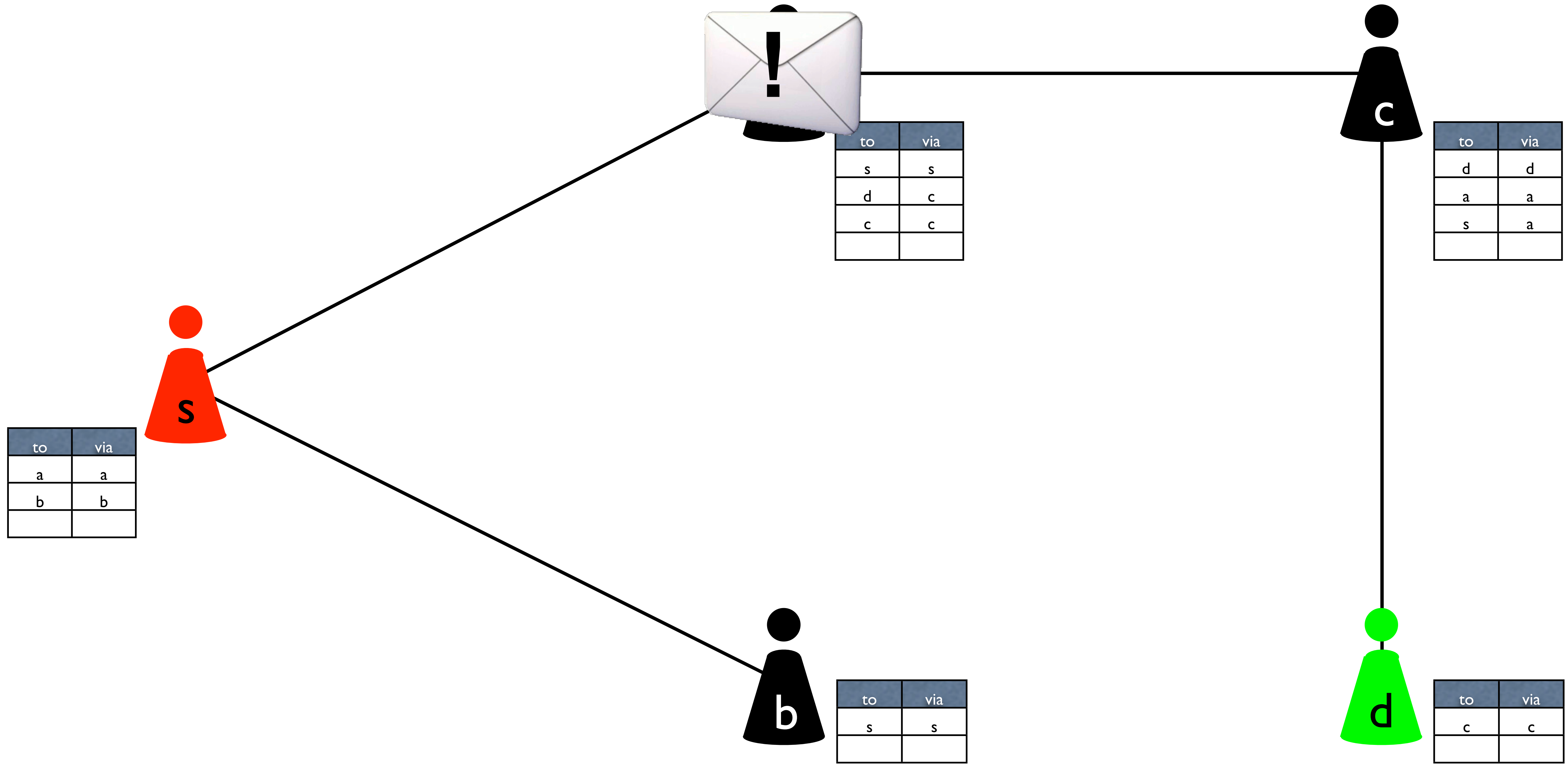


c has information about d
c answers route request and sends reply

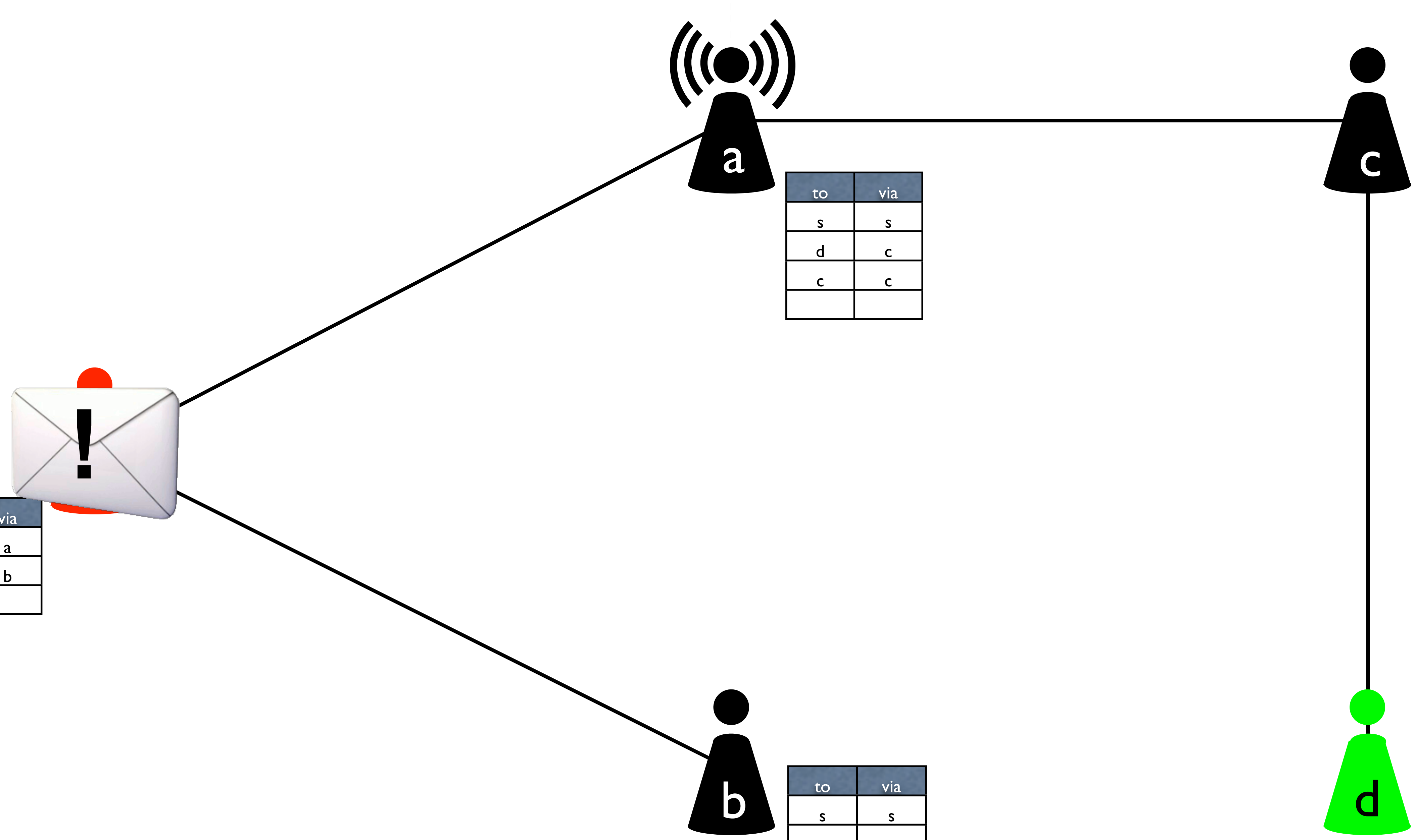
AODV – An Example



AODV – An Example



AODV – An Example



to	via
a	a
b	b

to	via
s	s
d	c
c	c

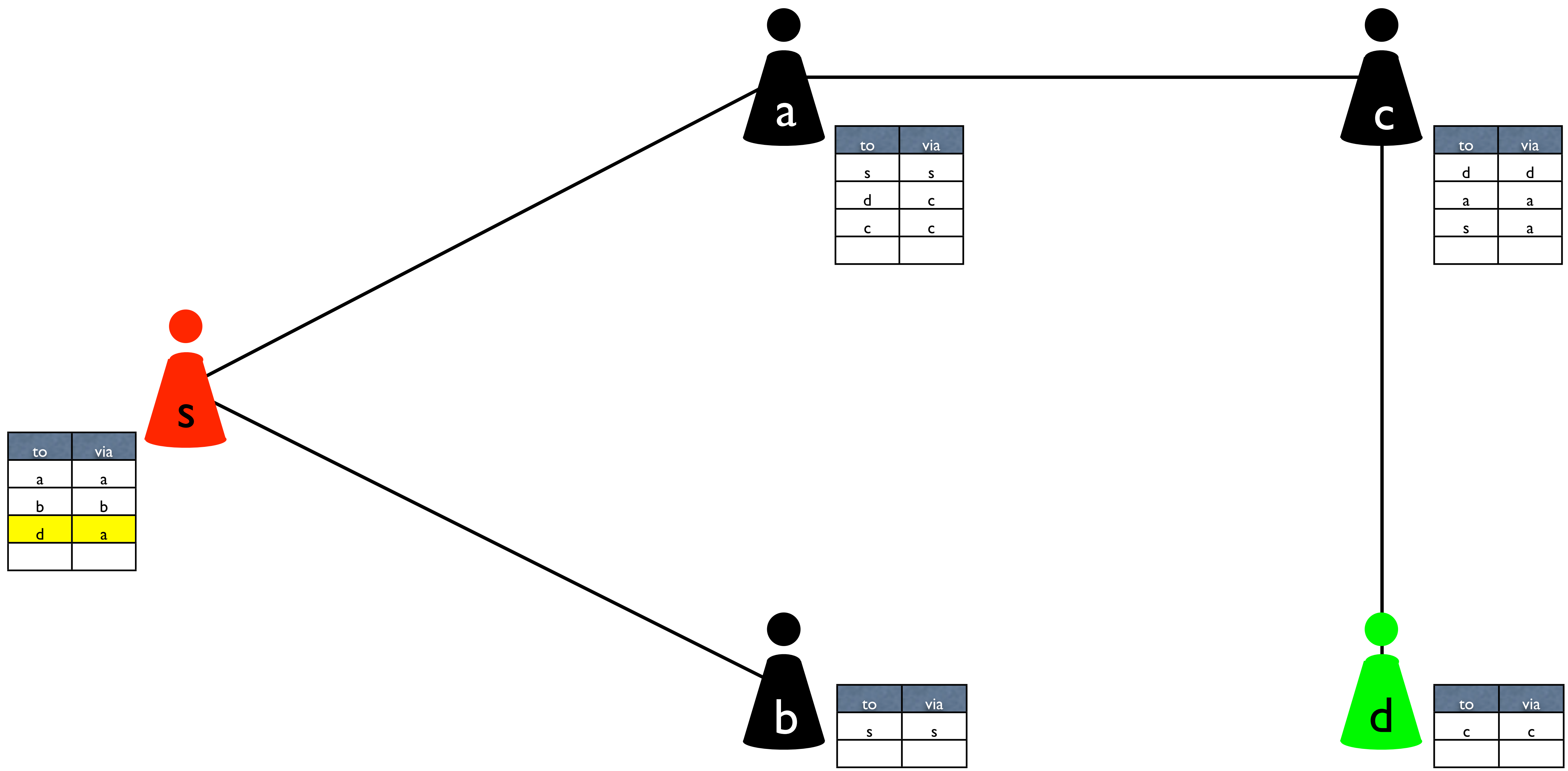
to	via
d	d
a	a
s	a

to	via
s	s

to	via
c	c

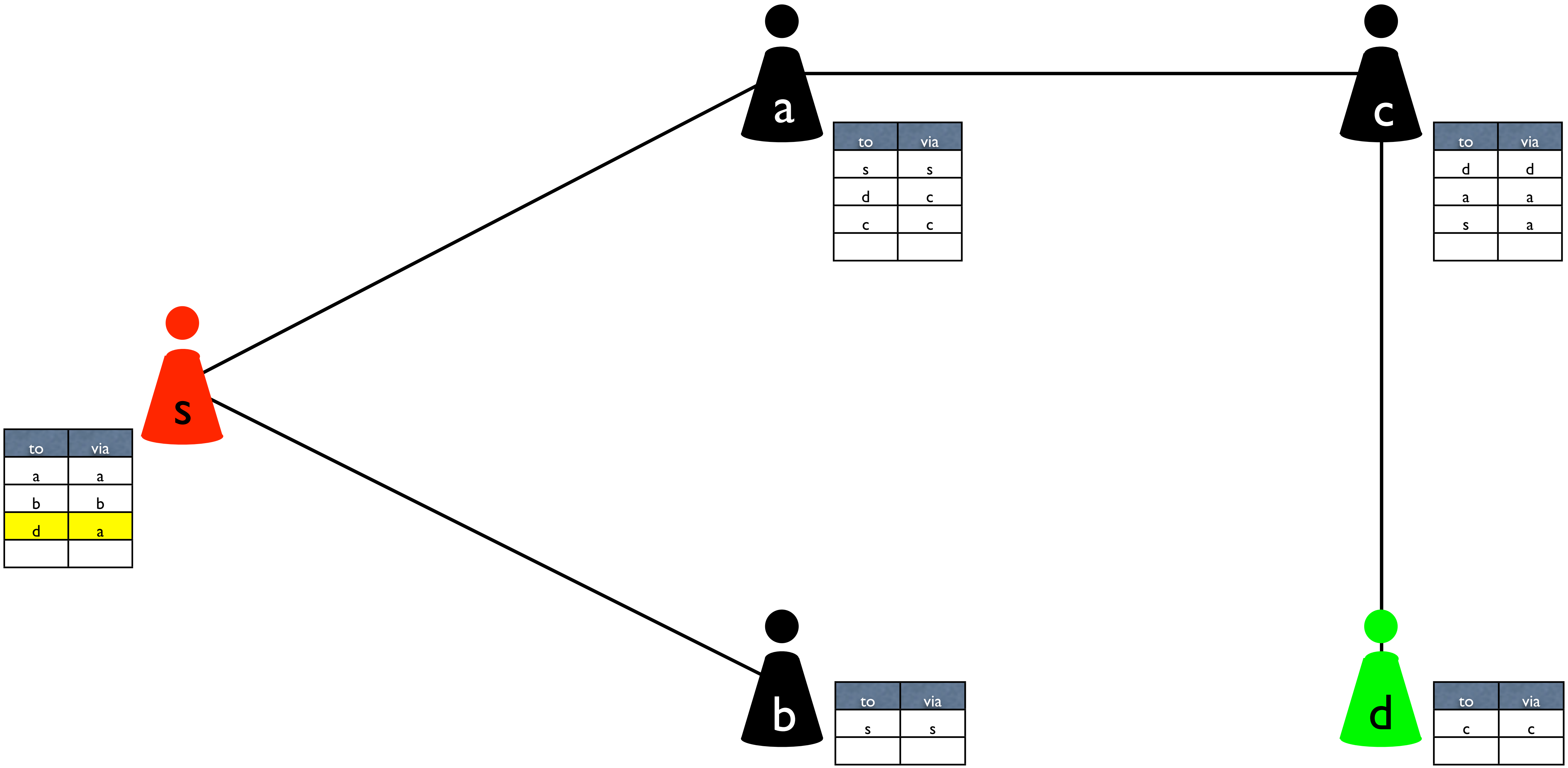
a forwards route reply

AODV - An Example

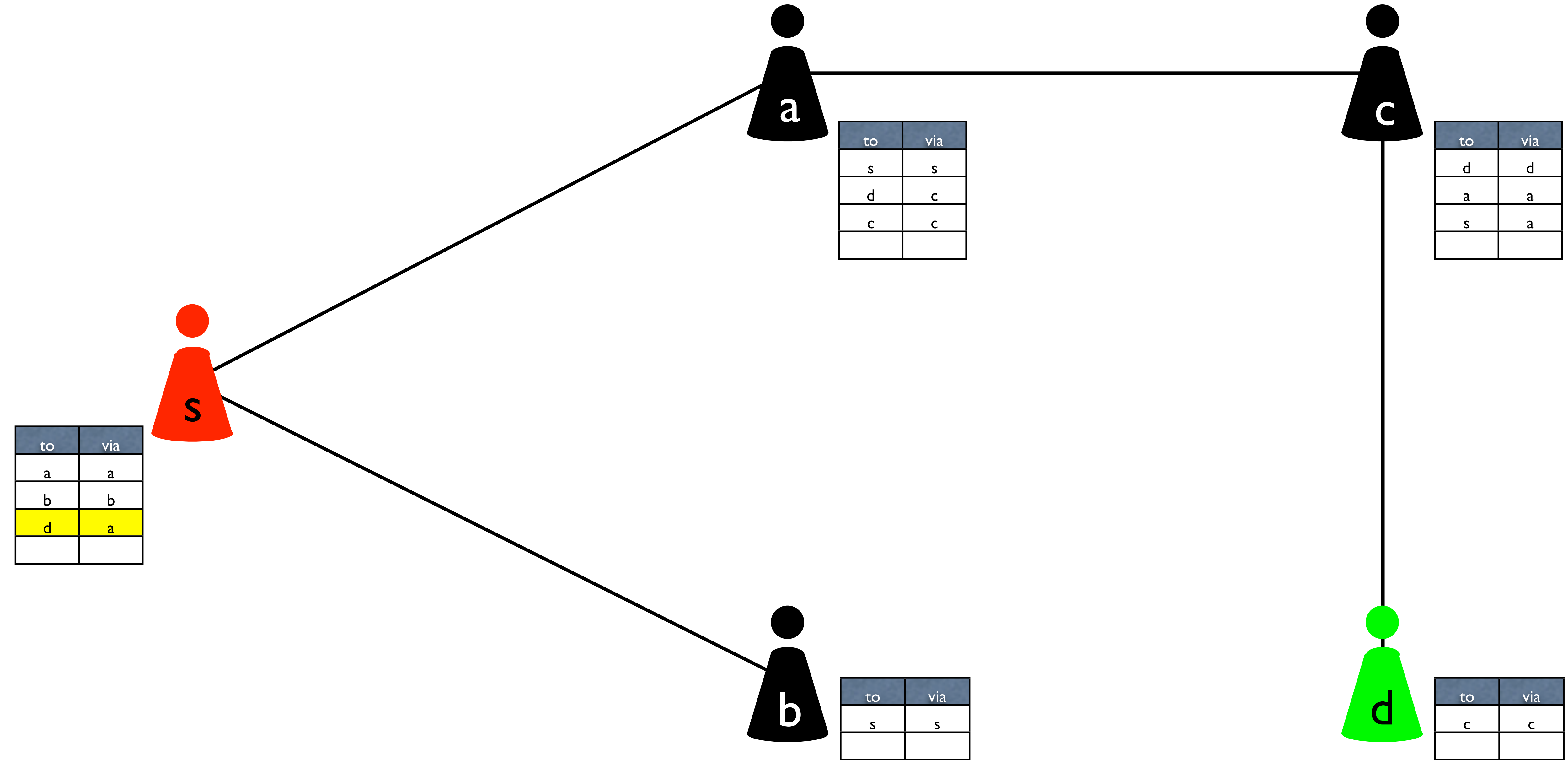


a forwards route reply

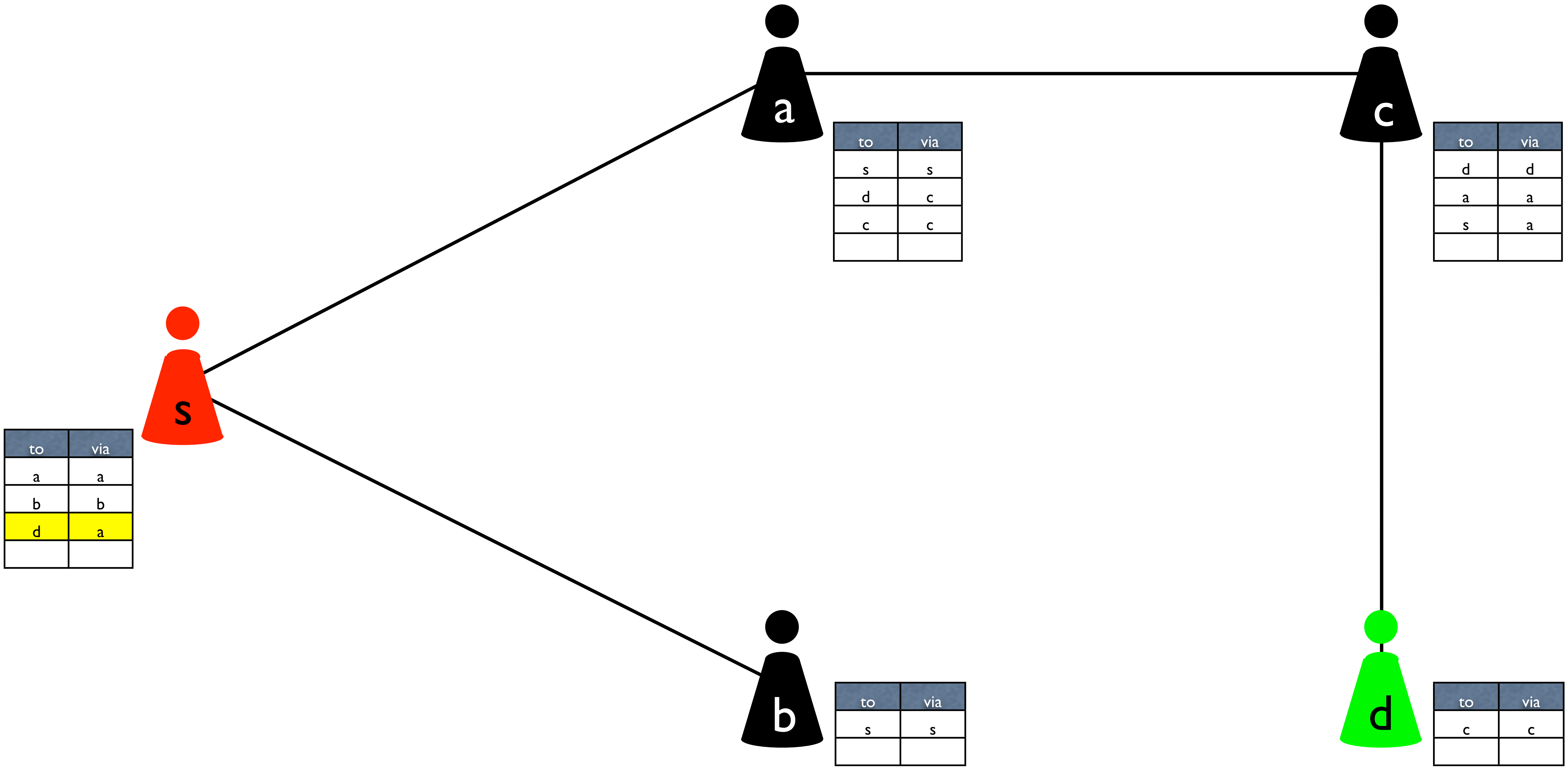
AODV – An Example



AODV – An Example

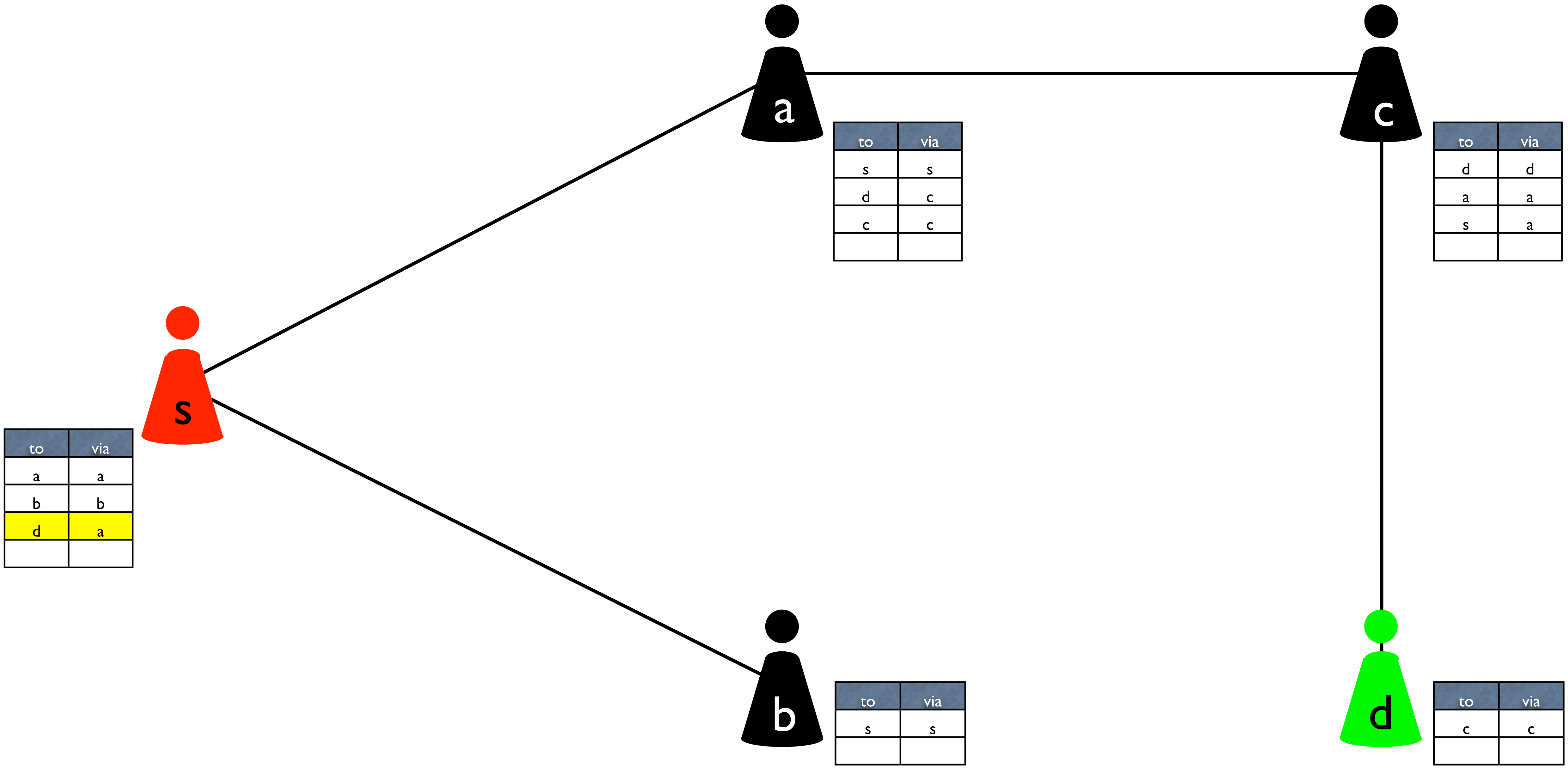


AODV – An Example



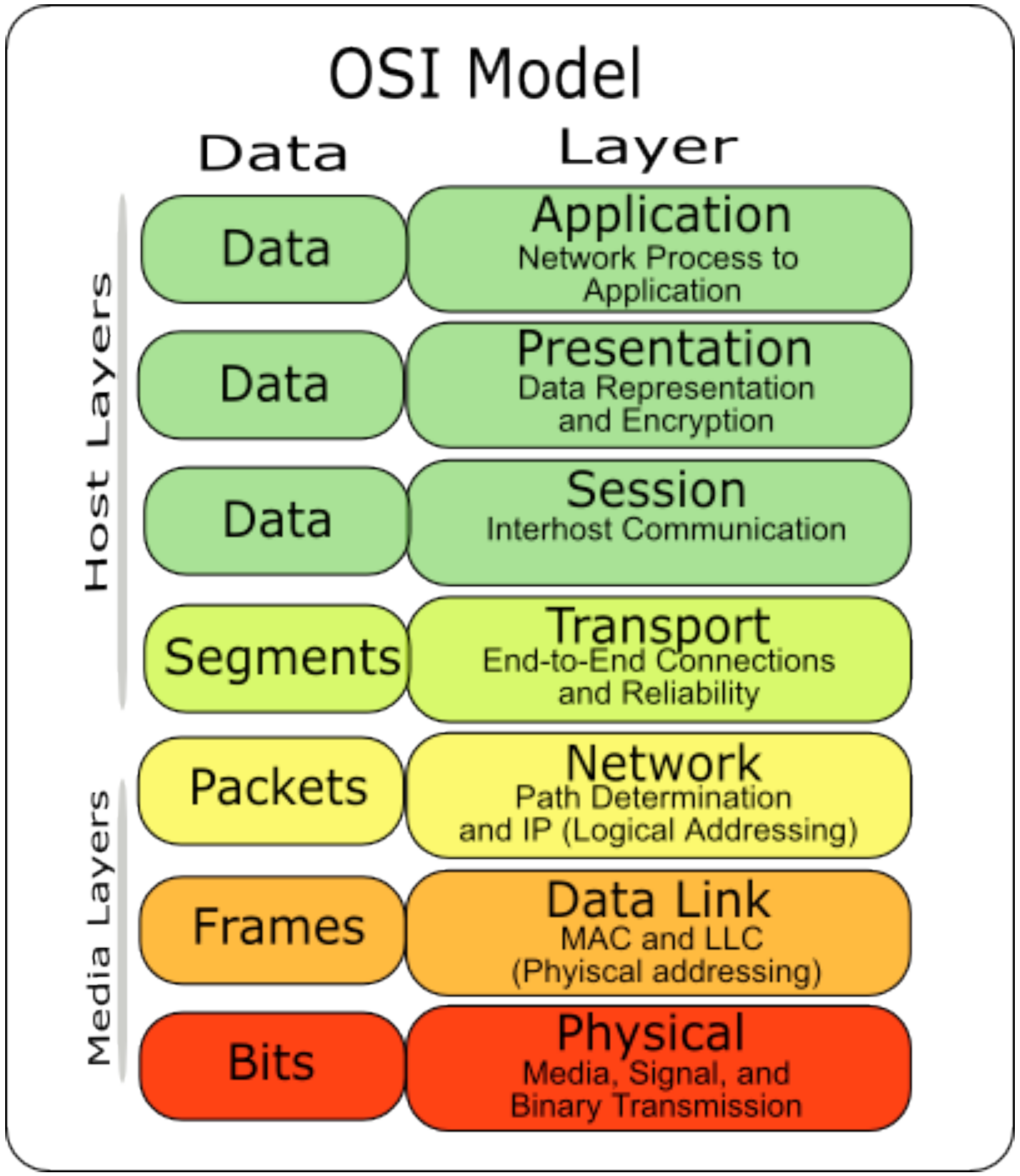
s has found a route to d

AODV – An Example



s has found a route to d

Different Network Layers



© NICTA 2011

Routing Protocols and Routing Tables

- Routing protocols
  - find (optimal) route
  - properties
    - loop freedom (no packet travels in loops)
    - route correctness (if a route is found, the route is valid)
    - route found (if a route exists, at least one route is found)
    - packet delivery

- Routing tables
  - data structure
  - belongs to client/router
  - lists destinations
  - sometimes metrics

© NICTA 2011