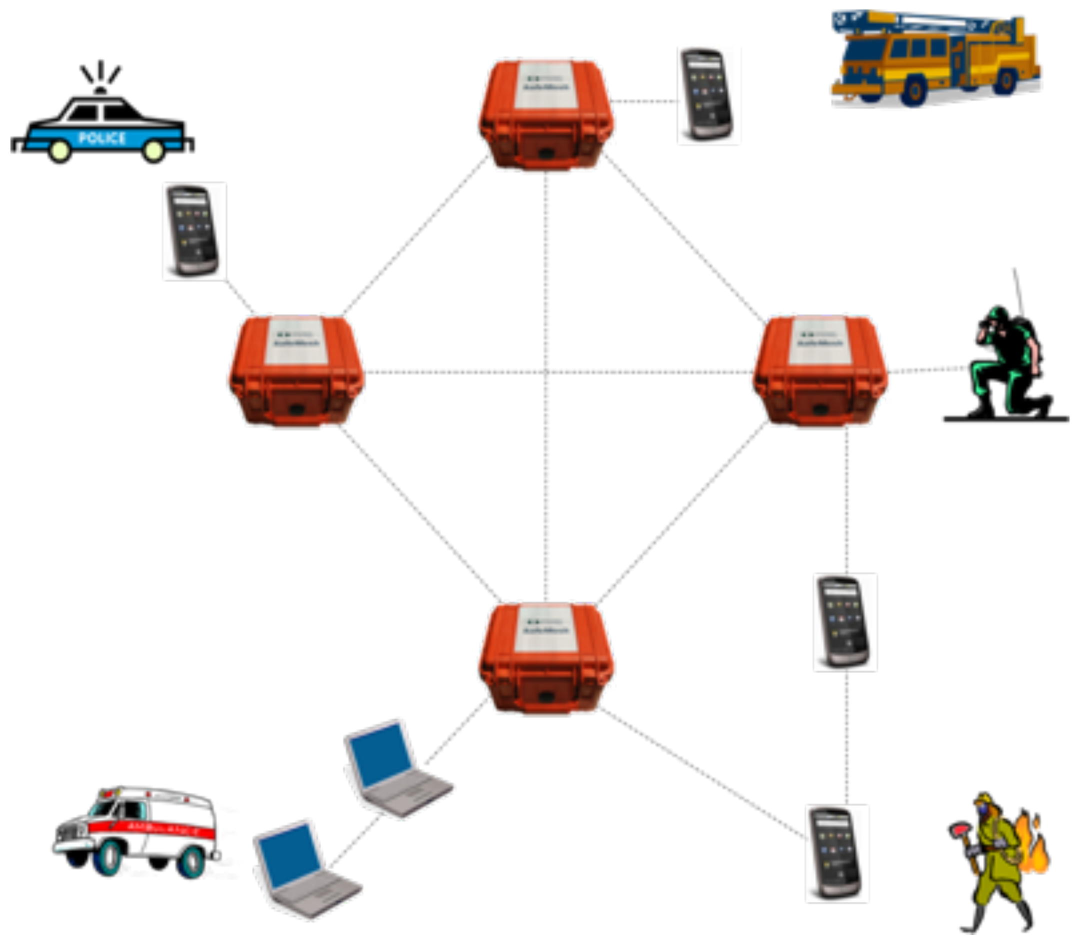
Formal Methods for Wireless Mesh Networks

Peter Höfner

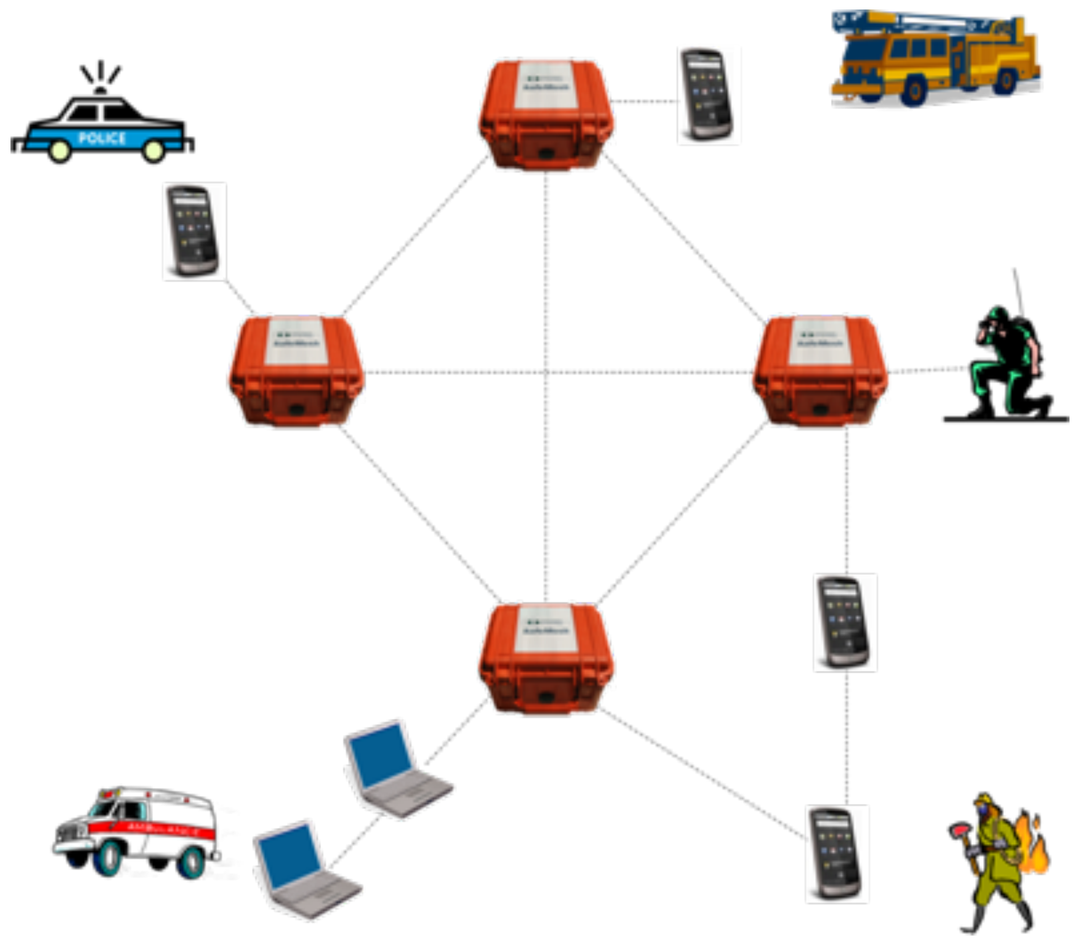
What is the Problem?

- 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
  - ...



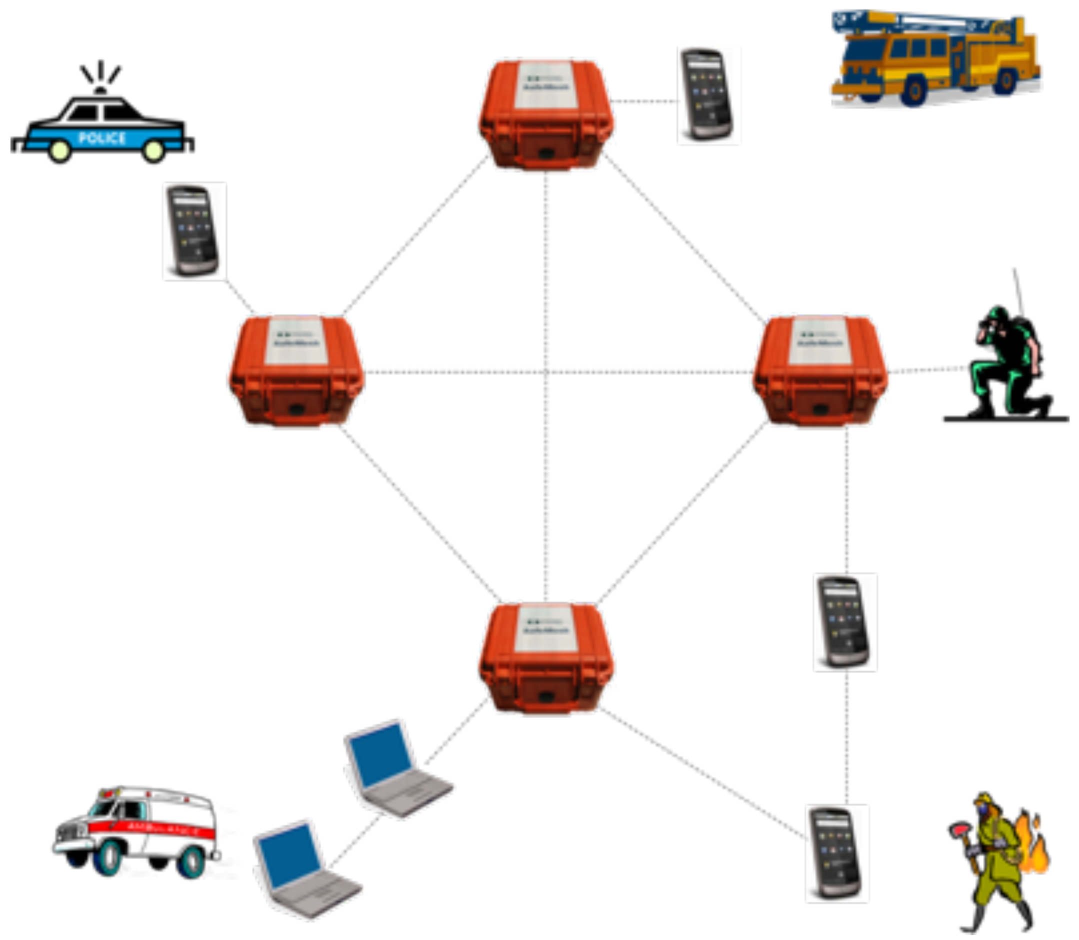
What is the Problem?

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



What is the Problem?

- 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



What is the Problem?

"Our requirement was for a system breadcrumb type deployment
over at least 4 nodes and maintain a throughput of around 5Mbps–10Mbps to enable 'good' quality video to be passed. The commercial devices failed to meet our requirements [...]"

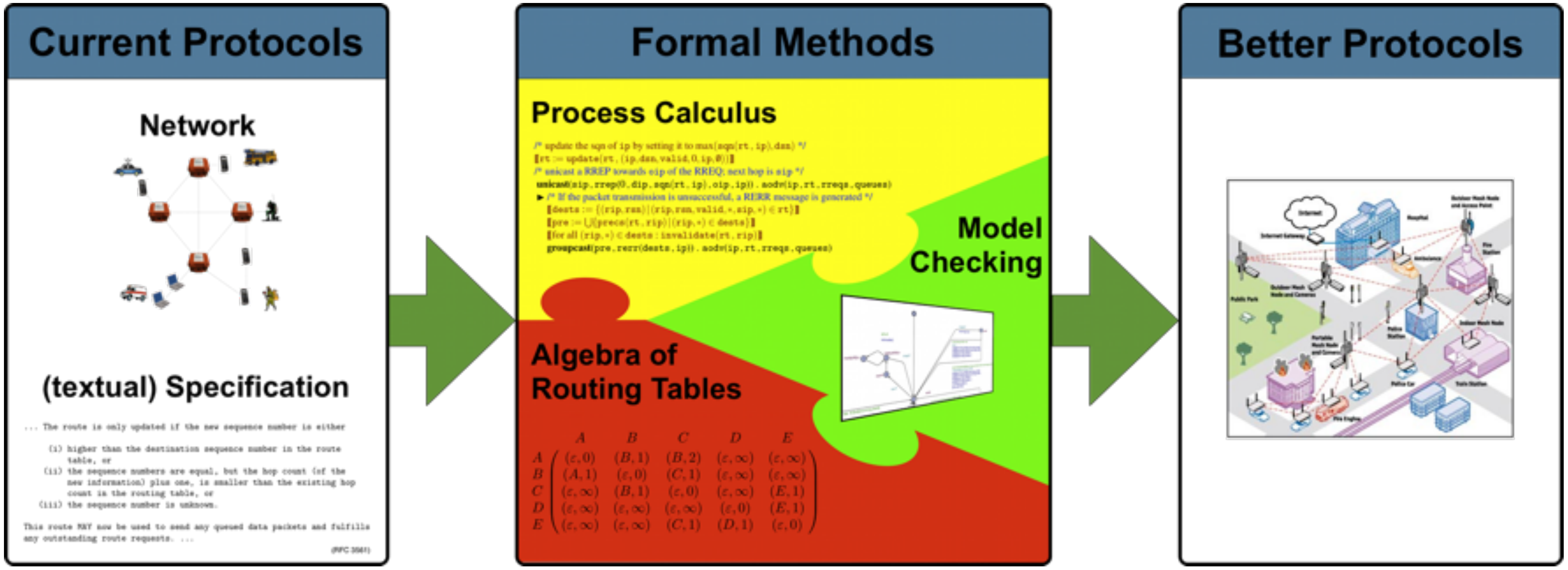
Rick Loebler, Applied Technology Manager,
NSW Police Force

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

Formal Methods for Mesh Networks

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



... 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.

This route MAY now be used to send any queued data packets and fulfills any outstanding route requests. ...

(RFC 3561)

```
/* 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)
```

$$\begin{array}{c|ccccc} & A & B & C & D & E \\ \hline A & (\varepsilon,0) & (B,1) & (B,2) & (\varepsilon,\infty) & (\varepsilon,\infty) \\ B & (A,1) & (\varepsilon,0) & (C,1) & (\varepsilon,\infty) & (\varepsilon,\infty) \\ C & (\varepsilon,\infty) & (B,1) & (\varepsilon,0) & (\varepsilon,\infty) & (E,1) \\ D & (\varepsilon,\infty) & (\varepsilon,\infty) & (\varepsilon,\infty) & (\varepsilon,0) & (E,1) \\ E & (\varepsilon,\infty) & (\varepsilon,\infty) & (C,1) & (D,1) & (\varepsilon,0) \end{array}$$

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)

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: $(\mathtt{dip}\,,\,\mathtt{dsn}\,,\,\mathtt{val}\,,\,\mathtt{hops}\,,\,\mathtt{nhip}\,,\,\mathtt{pre})$

Ad Hoc On-Demand Distance Vector Protocol

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

Ad Hoc On-Demand Distance Vector Protocol

- Properties of AODV
  - loop freedom ✓
  - route correctness ✓
  - route found ✗
  - packet delivery ✗

Ad Hoc On-Demand Distance Vector Protocol

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

Process Algebra

```
+ [ (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)⟧
```

Process Algebra

- New process algebra developed
- Language for formalising specs of network protocols
- Key features:
  - guarantee broadcast
  - prioritised unicast
  - data handling
- Achievements
  - full concise specification of AODV (RFC 3561) (no time)
  - formally verified loop-freedom (without timeouts)
    - invariant proof
  - found several ambiguities, mistakes, shortcomings
  - found solutions for some limitations

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

Routing Algebra

$$\begin{array}{c|ccccc|}
 & A & B & C & D & E \\
A & (\epsilon, 0) & (B, 1) & (B, 2) & (\epsilon, \infty) & (\epsilon, \infty) \\
B & (A, 1) & (\epsilon, 0) & (C, 1) & (\epsilon, \infty) & (\epsilon, \infty) \\
C & (\epsilon, \infty) & (B, 1) & (\epsilon, 0) & (\epsilon, \infty) & (E, 1) \\
D & (\epsilon, \infty) & (\epsilon, \infty) & (\epsilon, \infty) & (\epsilon, 0) & (E, 1) \\
E & (\epsilon, \infty) & (\epsilon, \infty) & (C, 1) & (D, 1) & (\epsilon, 0)
\end{array}$$

Routing Algebra – Elements, Operators

- Routing table entries (no sequence number so far)

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

- Choice: $(A, 5) + (B, 2) = (B, 2)$
- Multiplication: $(A, 5) \cdot (B, 2) = (A, 7)$
  - destination and source must coincide

- idea: back to Backhouse, Carré, Griffin, Sobrinho

Routing Algebra – Elements, Operators

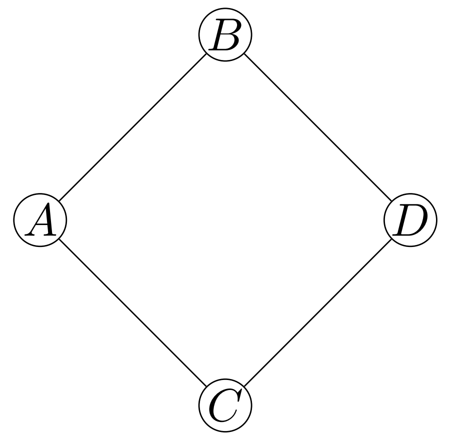
- Matrices over routing table entries

$$\begin{array}{c|ccccc|l}
 & A & B & C & D & \dots & \\
\hline
A & (\_, 0) & (B, 1) & (B, 2) & (\_, \infty) & & \text{routing table of } A \\
B & (A, 1) & (\_, 0) & (C, 1) & (\_, \infty) & \dots & \\
C & (\_, \infty) & (B, 1) & (\_, 0) & (\_, \infty) & & \\
D & (\_, \infty) & (\_, \infty) & (\_, \infty) & (\_, 0) & & \\
\vdots & & \vdots & & & \ddots & \\
\hline
 & & \text{“routes” to } B & & & &
\end{array}$$

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

Example

- A route request is broadcast



$$\underbrace{\begin{pmatrix} (\_,0) & (B,1) & (C,1) & (\_,\infty) \\ (A,1) & (\_,0) & (\_,\infty) & (D,1) \\ (A,1) & (\_,\infty) & (\_,0) & (D,1) \\ (\_,\infty) & (B,1) & (C,1) & (\_,0) \end{pmatrix}}_{\text{topology}} \bullet \underbrace{\begin{pmatrix} (\_,0) & (\_,\infty) & (\_,\infty) & (\_,\infty) \\ (\_,\infty) & (\_,\infty) & (\_,\infty) & (\_,\infty) \\ (\_,\infty) & (\_,\infty) & (\_,\infty) & (\_,\infty) \\ (\_,\infty) & (\_,\infty) & (\_,\infty) & (\_,\infty) \end{pmatrix}}_{\text{sender}} \bullet \underbrace{\begin{pmatrix} (\_,0) & (B,1) & (\_,\infty) & (\_,\infty) \\ (\mathbf{D},\mathbf{3}) & (\_,0) & (\_,\infty) & (\_,\infty) \\ (A,1) & (\_,\infty) & (\_,0) & (D,1) \\ (C,2) & (\_,\infty) & (C,1) & (\_,0) \end{pmatrix}}_{\text{routing table}}$$

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

Sent Messages

- Achievements
  - sending messages

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

  - broadcast, unicast, groupcast are the same (modelled by different topologies)
  - Kleene star models flooding the network (modal operators terminate flooding)
  - great potential for automation (Prover9, Isabelle, ...)

Conclusion/Future Work

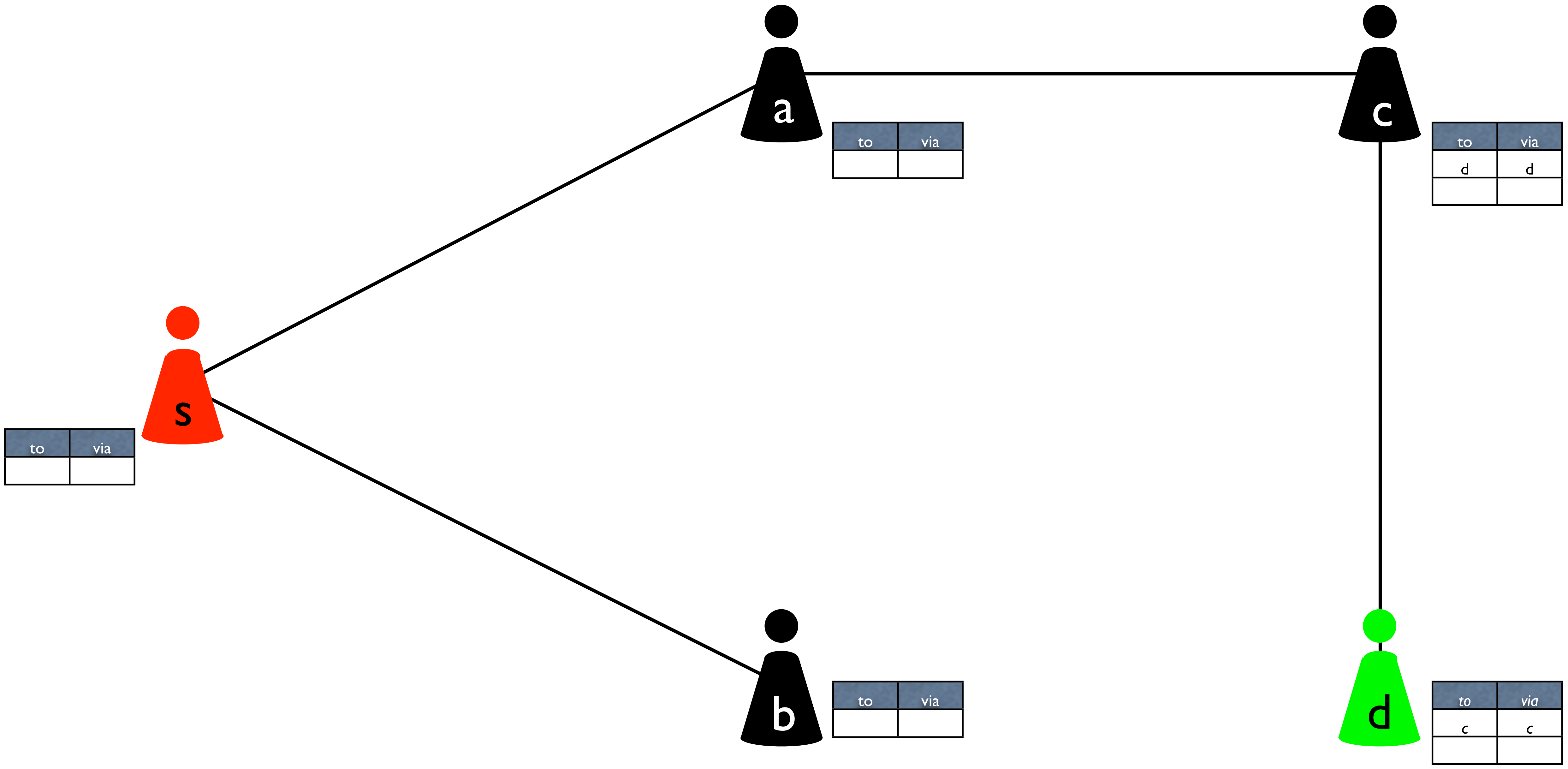
- So far concentrated on AODV
  - well known
  - IETF standard
  - known limitations
- Extend formal methods to other protocols
  - OSLR, DYMO, ...
- Add further necessary concepts
  - time
  - probability

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

NICTA

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

AODV – An Example



Routing table at s:

to	via

Routing table at a:

to	via

Routing table at b:

to	via

Routing table at c:

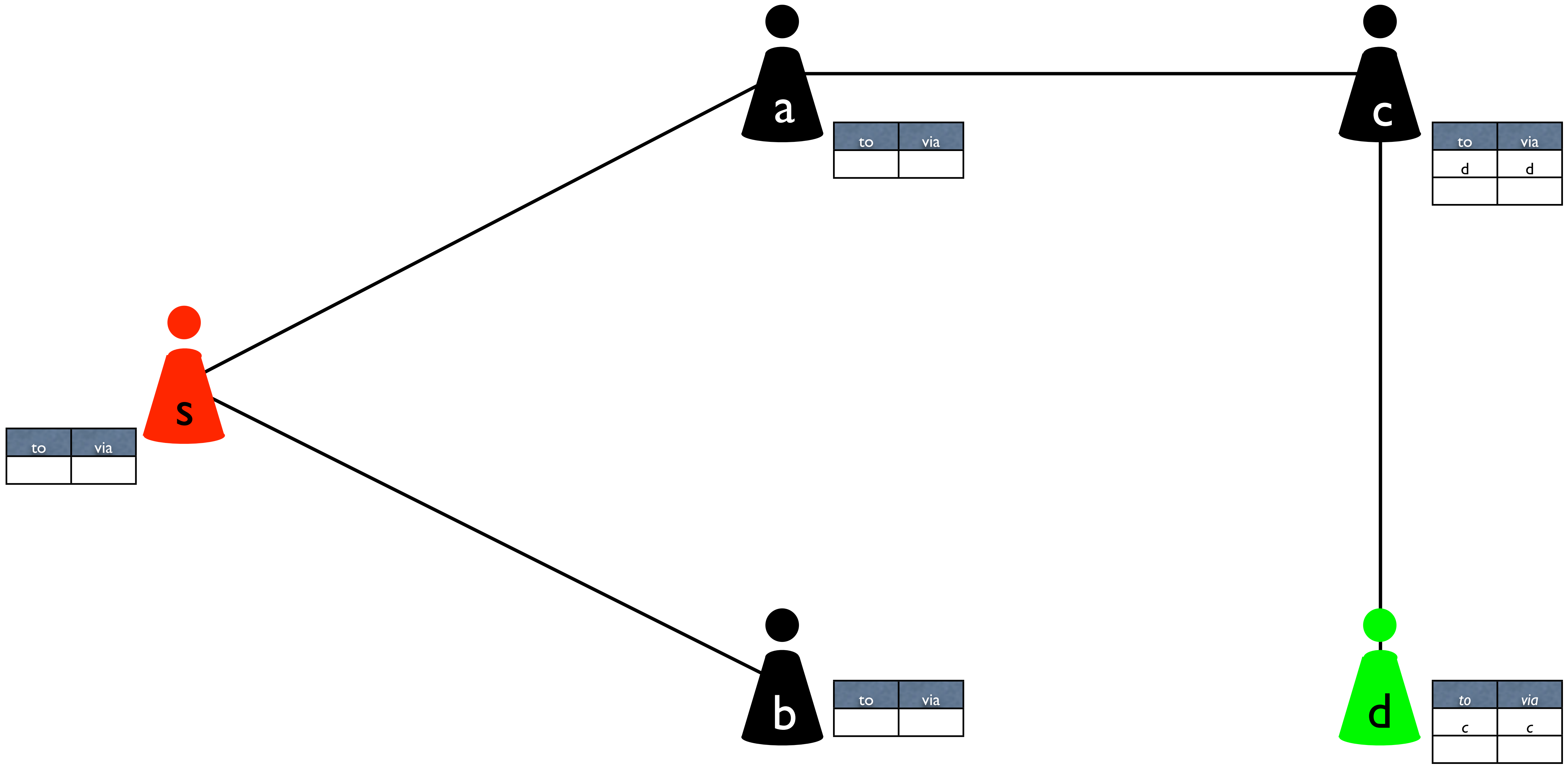
to	via
d	d

Routing table at d:

to	via
c	c

s is looking for a route to d

AODV – An Example



s:

to	via

a:

to	via

b:

to	via

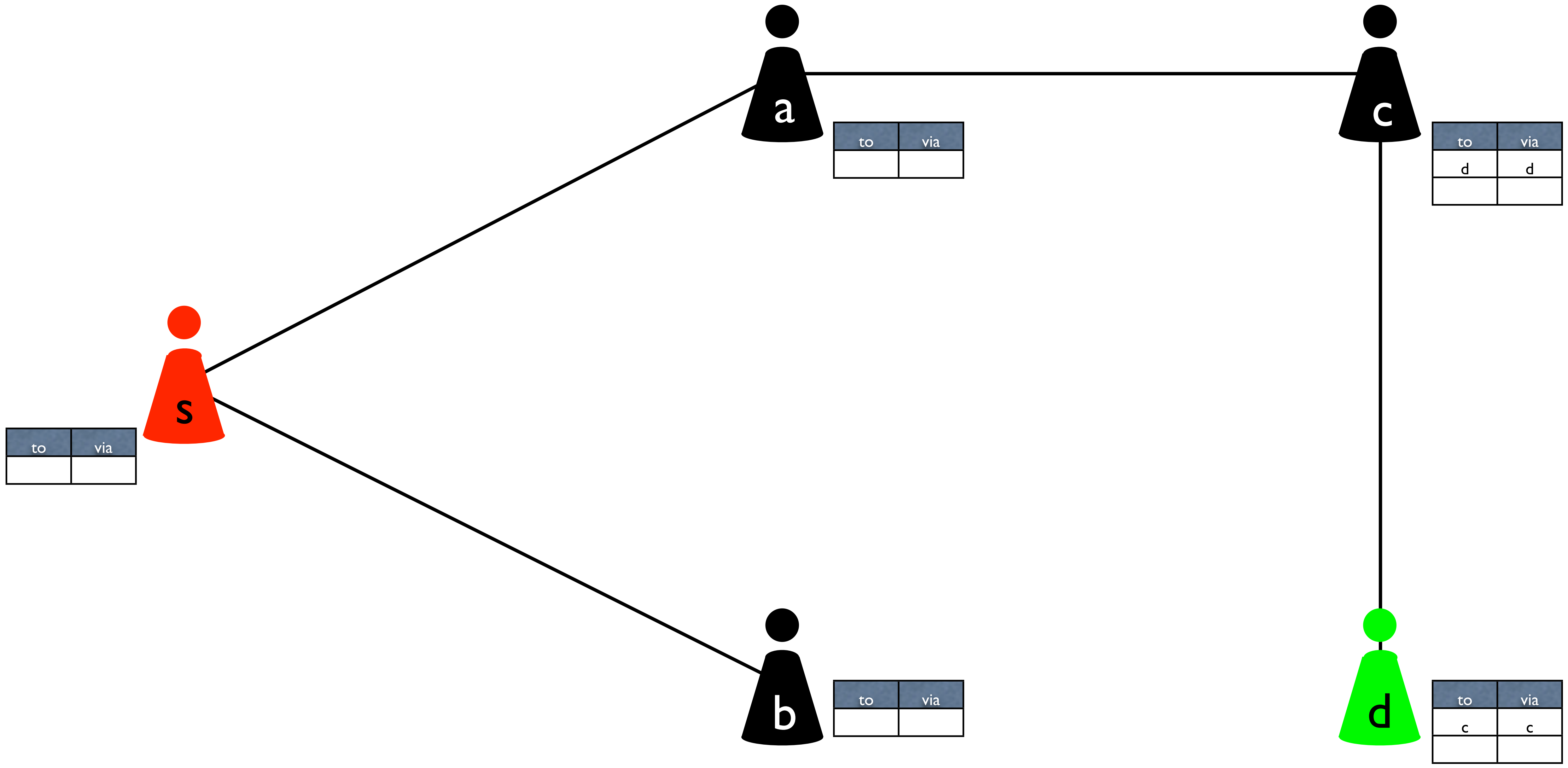
c:

to	via
d	d

d:

to	via
c	c

AODV – An Example



to	via

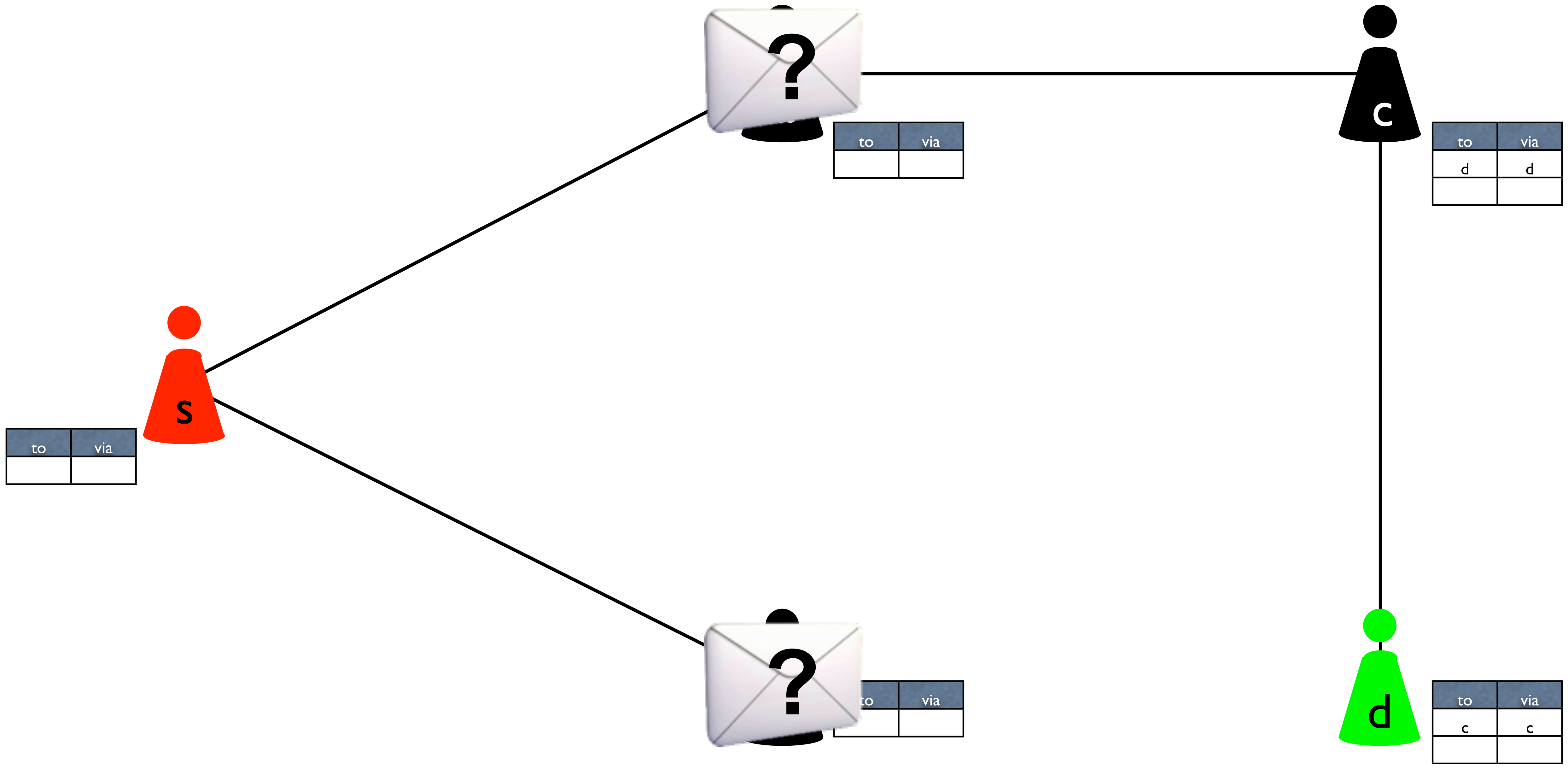
to	via
d	d

to	via

to	via

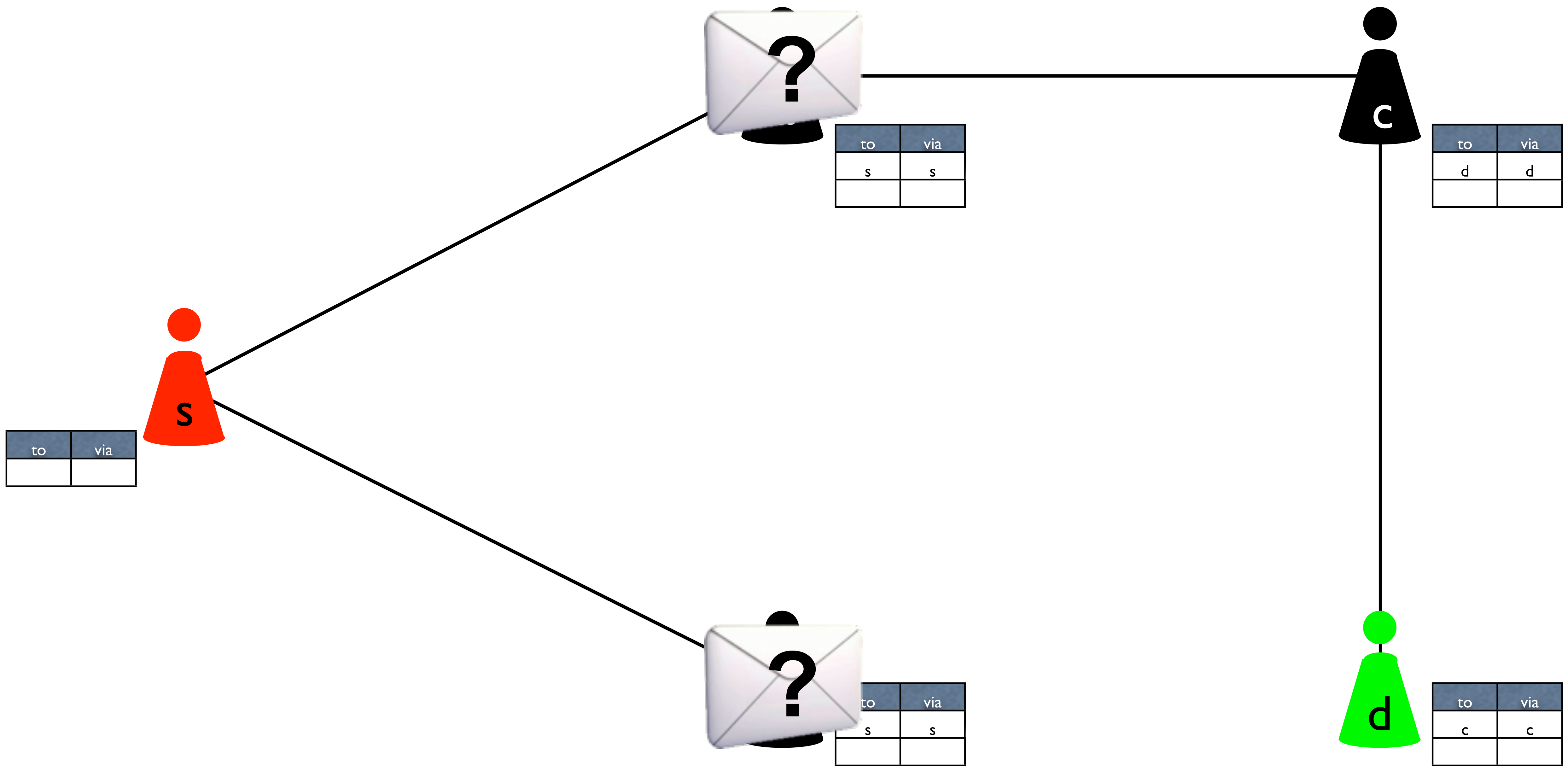
to	via
c	c

AODV – An Example



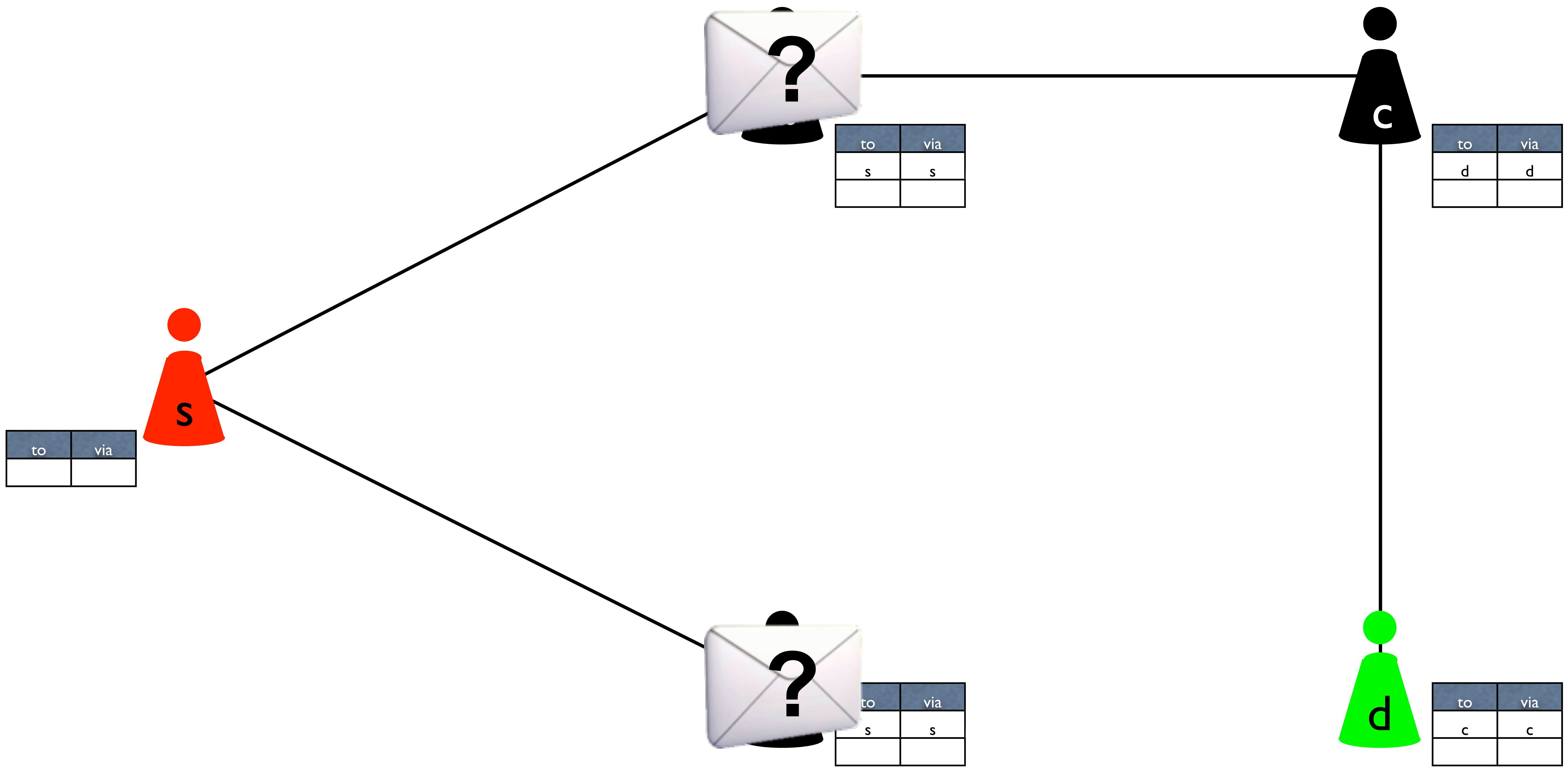
s broadcasts a route request

AODV – An Example

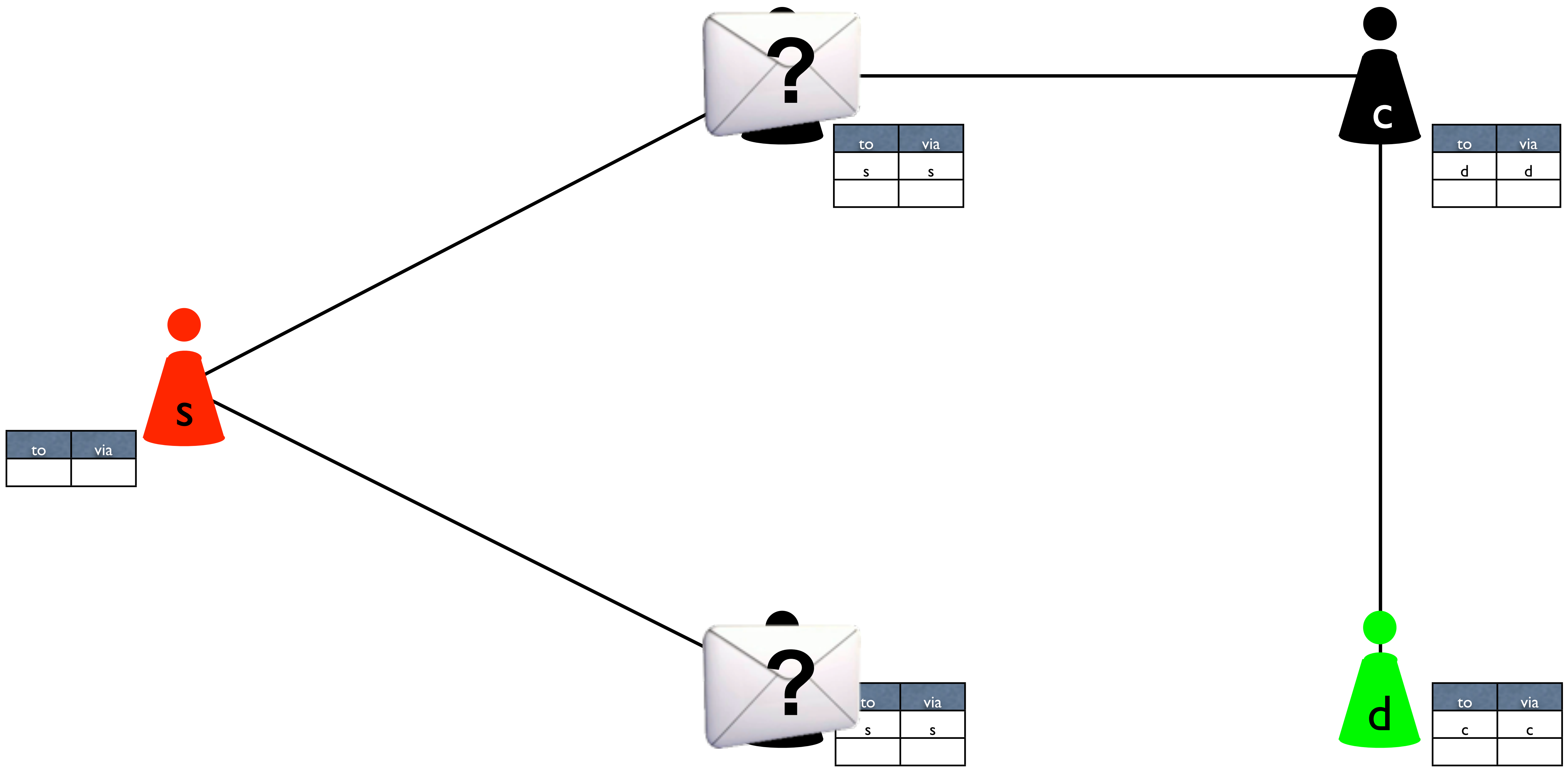


s broadcasts a route request

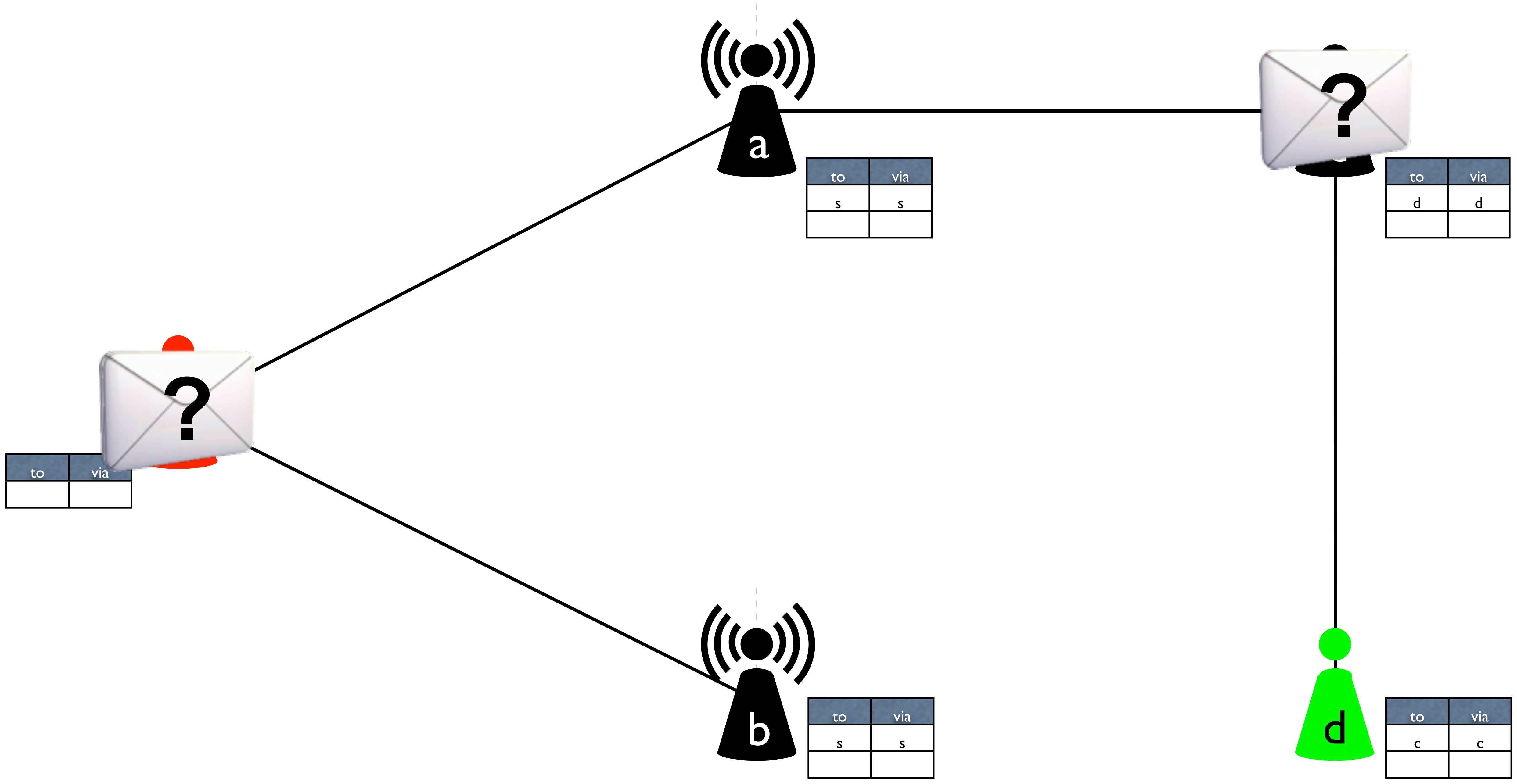
AODV – An Example



AODV – An Example

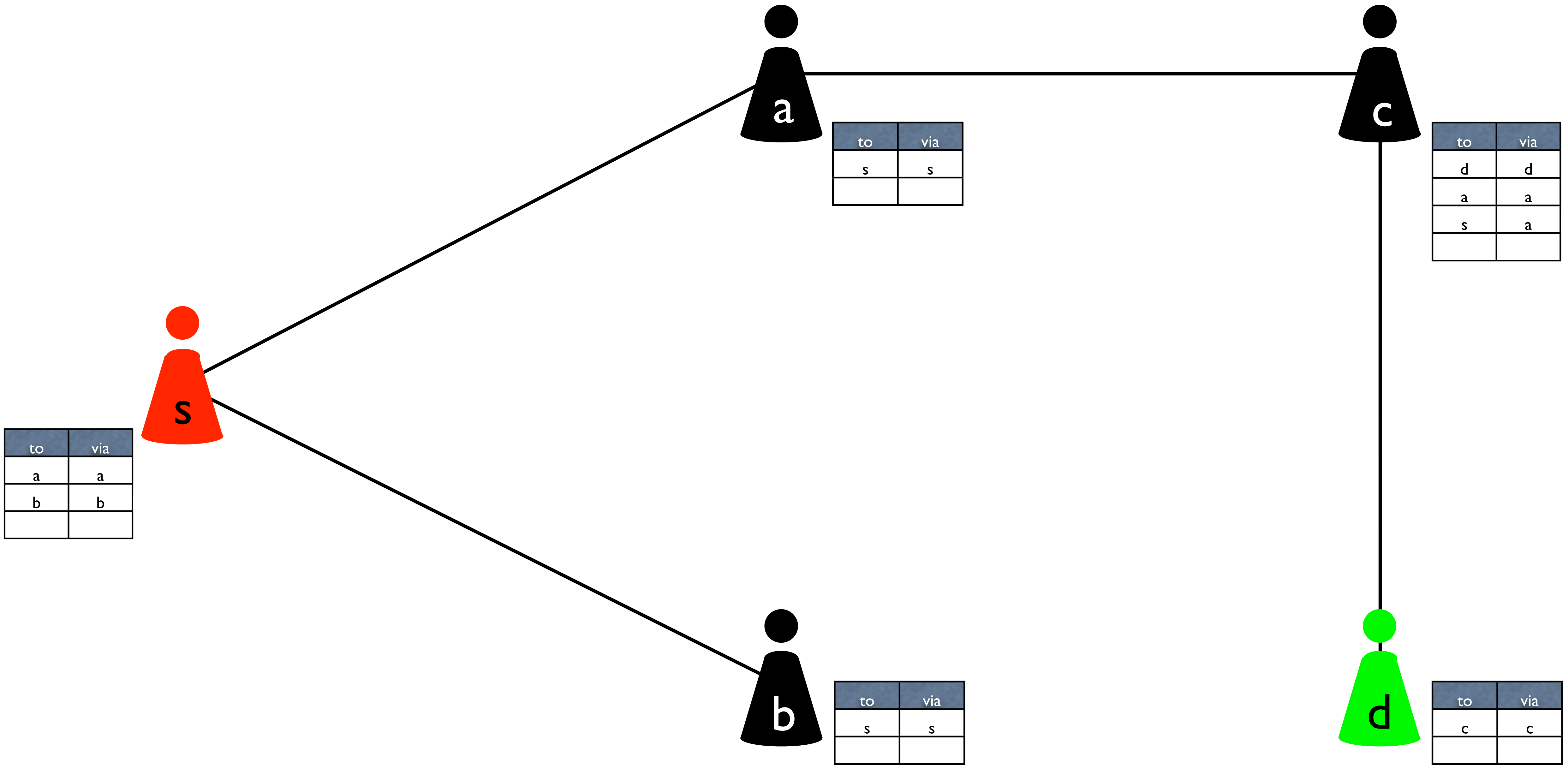


AODV – An Example



a,b forward the route request

AODV – An Example



to	via
s	s

to	via
d	d
a	a
s	a

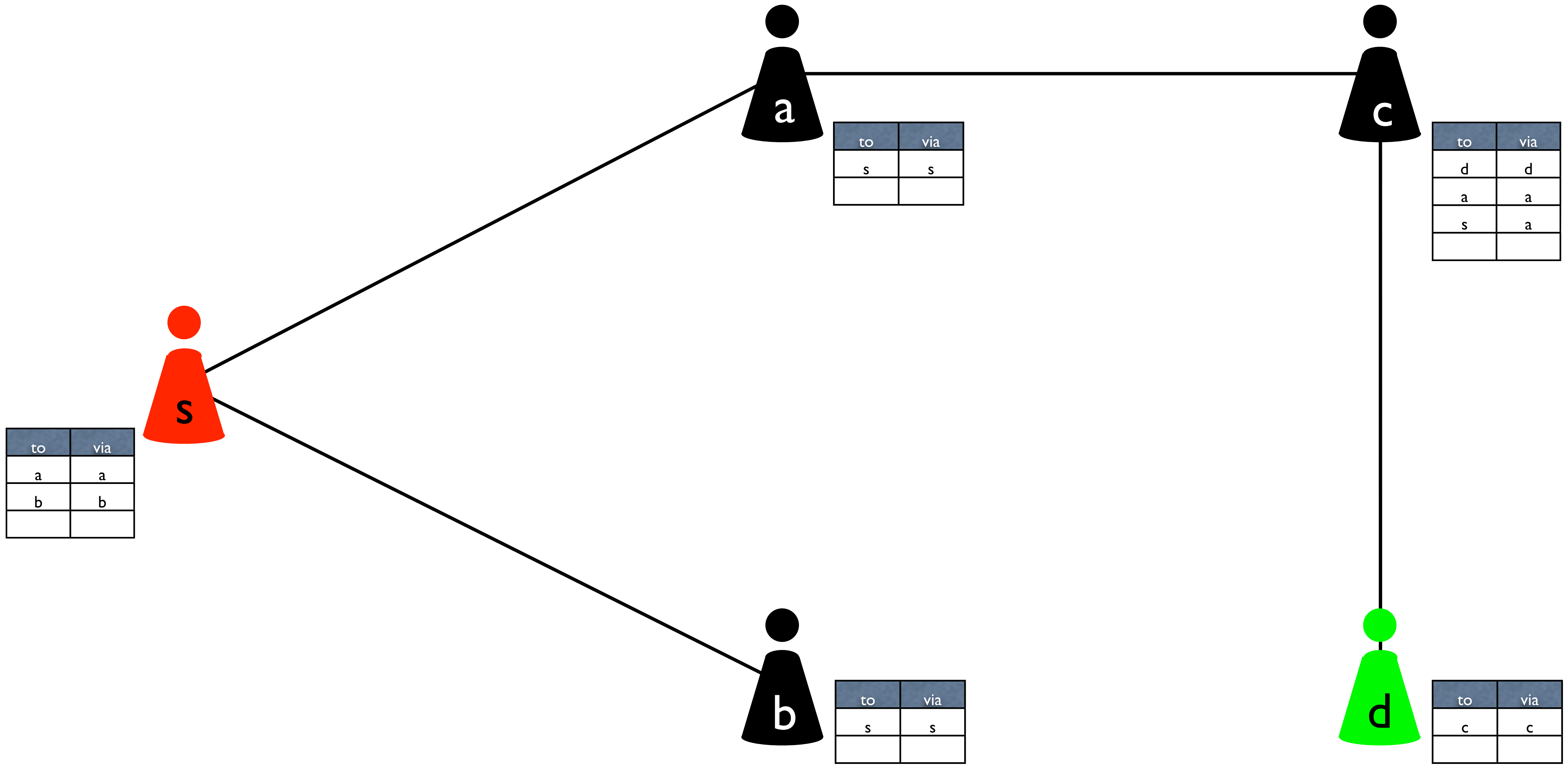
to	via
a	a
b	b

to	via
s	s

to	via
c	c

a,b forward the route request

AODV – An Example



to	via
s	s

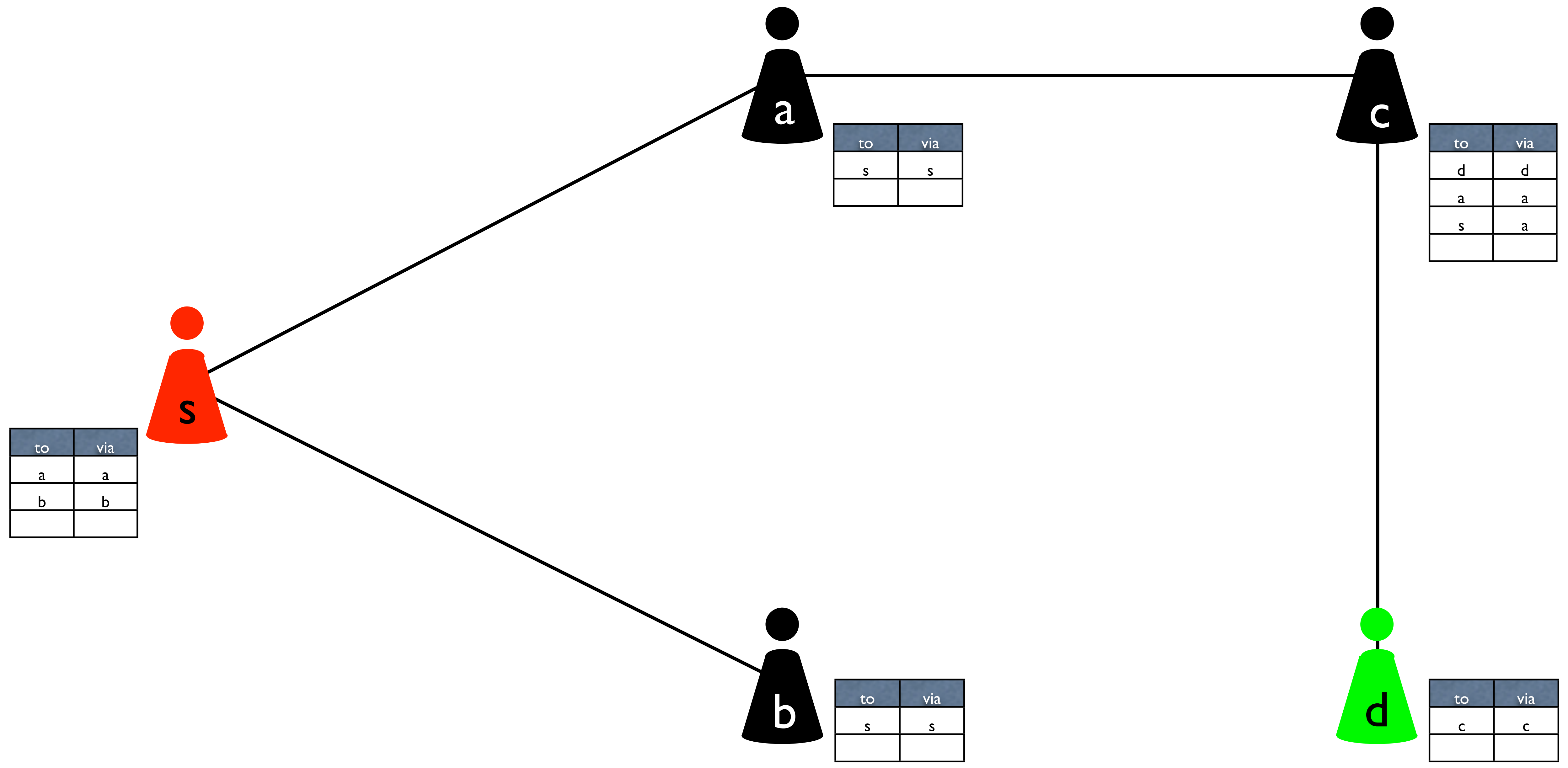
to	via
d	d
a	a
s	a

to	via
a	a
b	b

to	via
s	s

to	via
c	c

AODV – An Example



s:

to	via
a	a
b	b

a:

to	via
s	s

b:

to	via
s	s

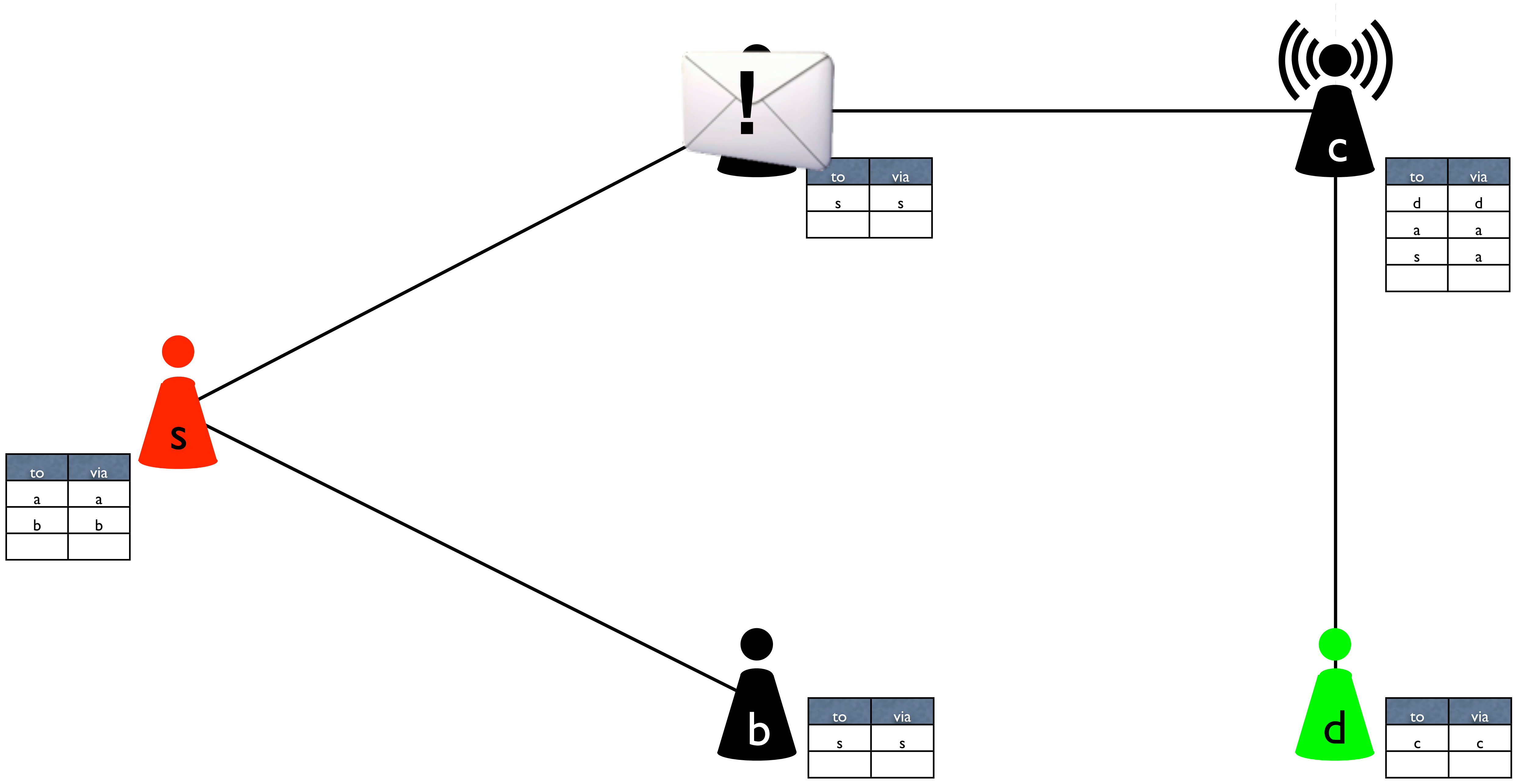
c:

to	via
d	d
a	a
s	a

d:

to	via
c	c

AODV – An Example



to	via
a	a
b	b

to	via
s	s

to	via
d	d
a	a
s	a

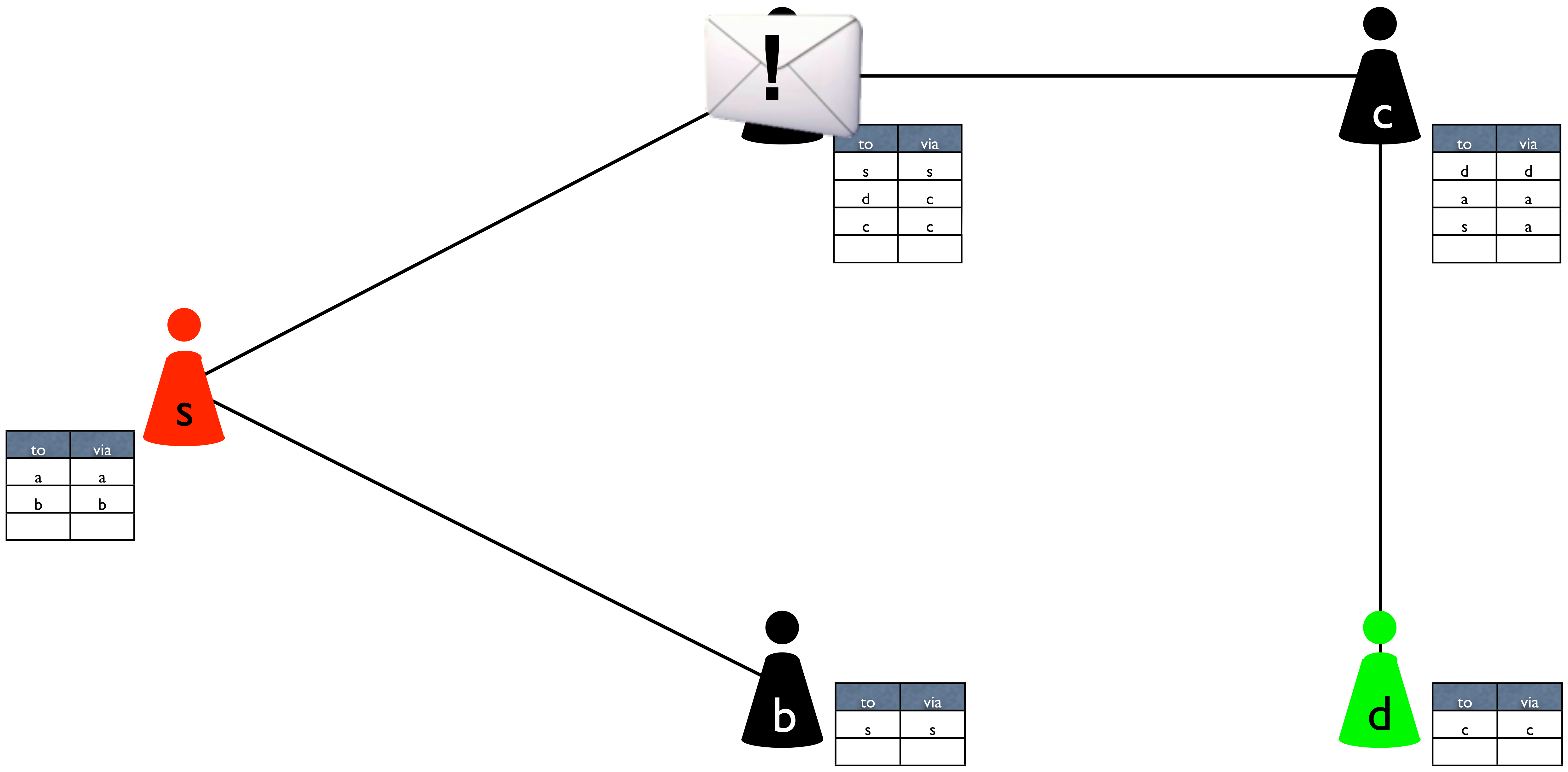
to	via
s	s

to	via
c	c

c has information about d

c answers route request and sends reply

AODV – An Example



to	via
s	s
d	c
c	c

to	via
d	d
a	a
s	a

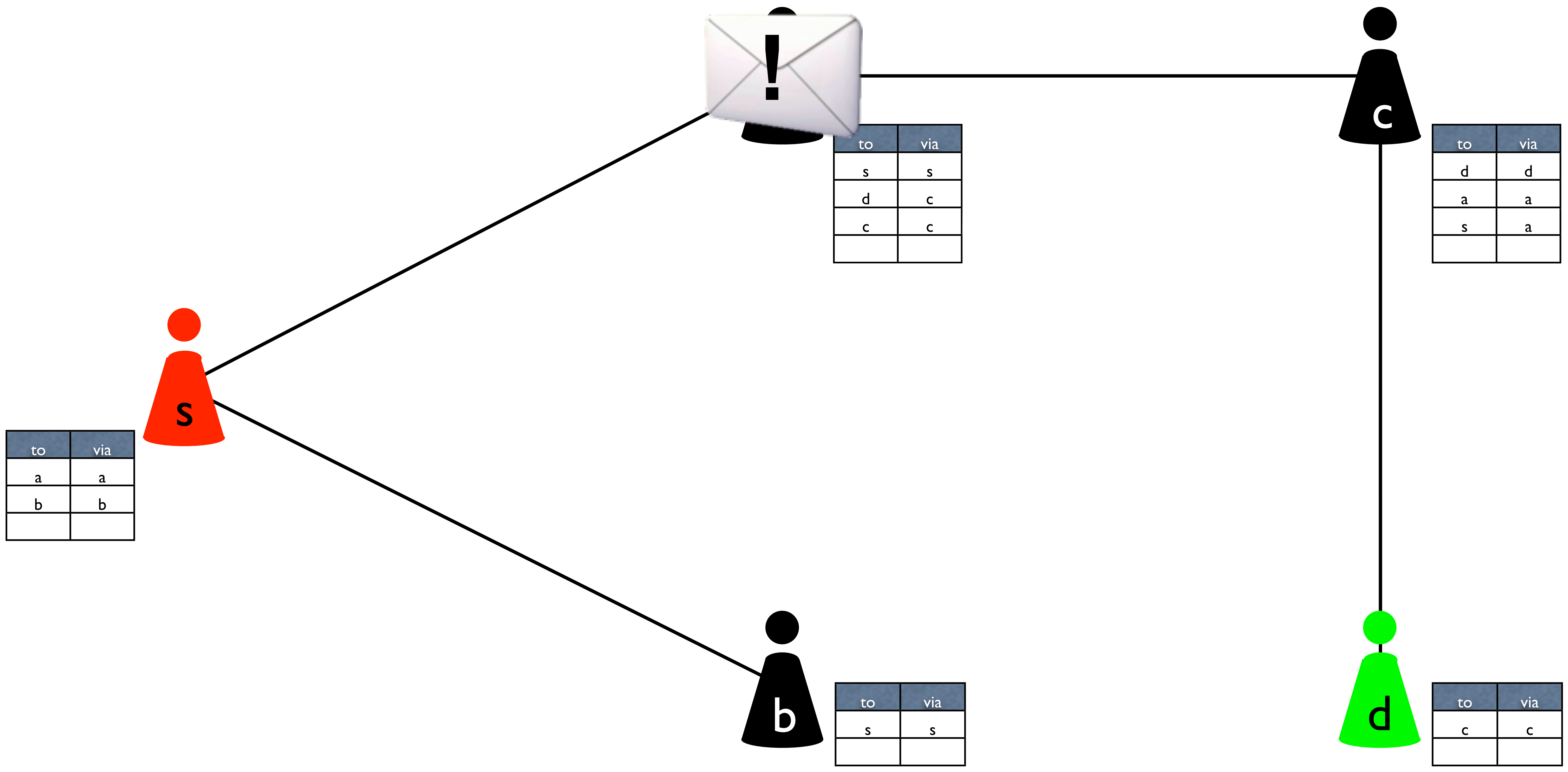
to	via
a	a
b	b

to	via
s	s

to	via
c	c

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

AODV – An Example



to	via
s	s
d	c
c	c

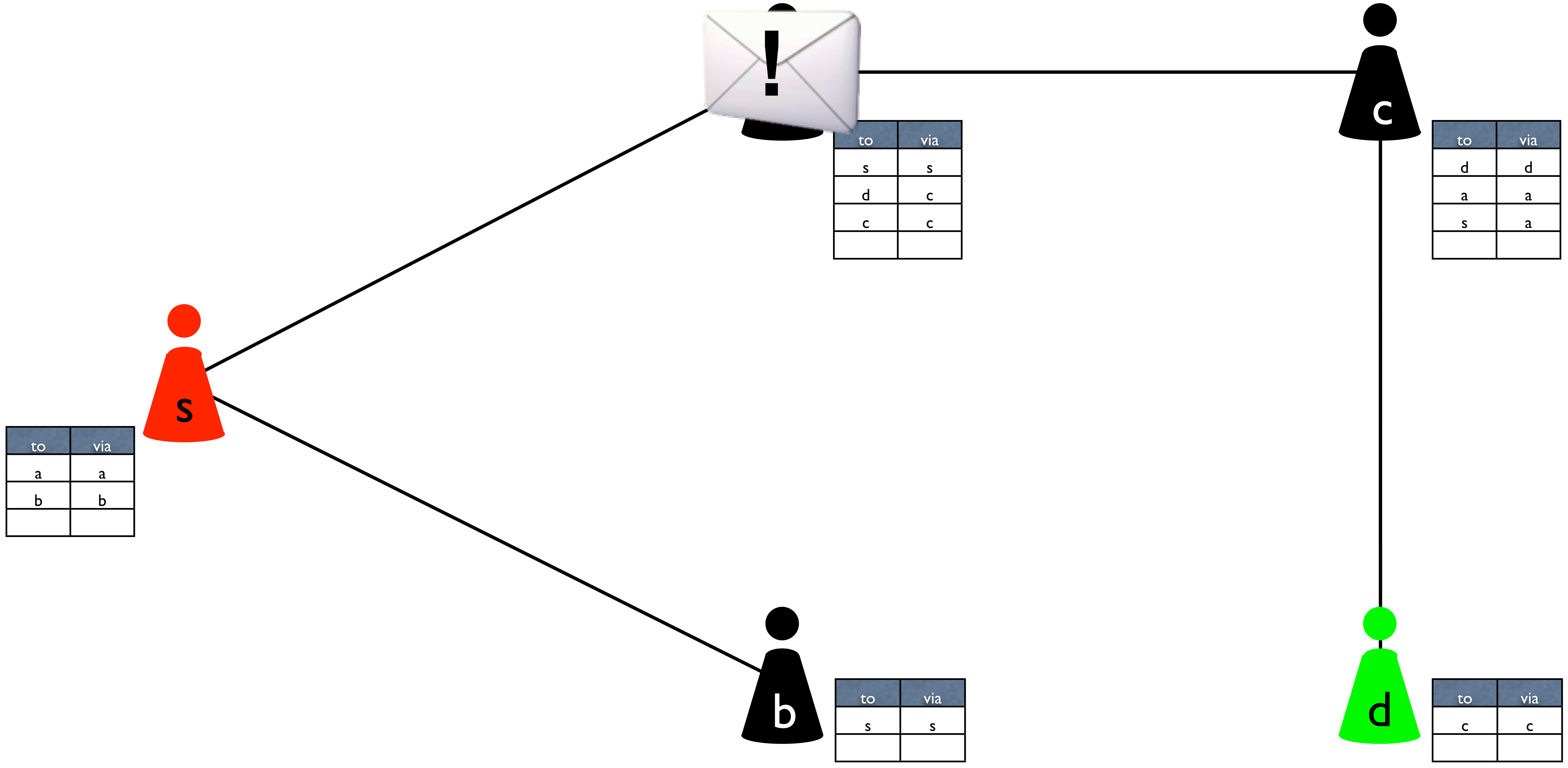
to	via
d	d
a	a
s	a

to	via
a	a
b	b

to	via
s	s

to	via
c	c

AODV – An Example



to	via
a	a
b	b

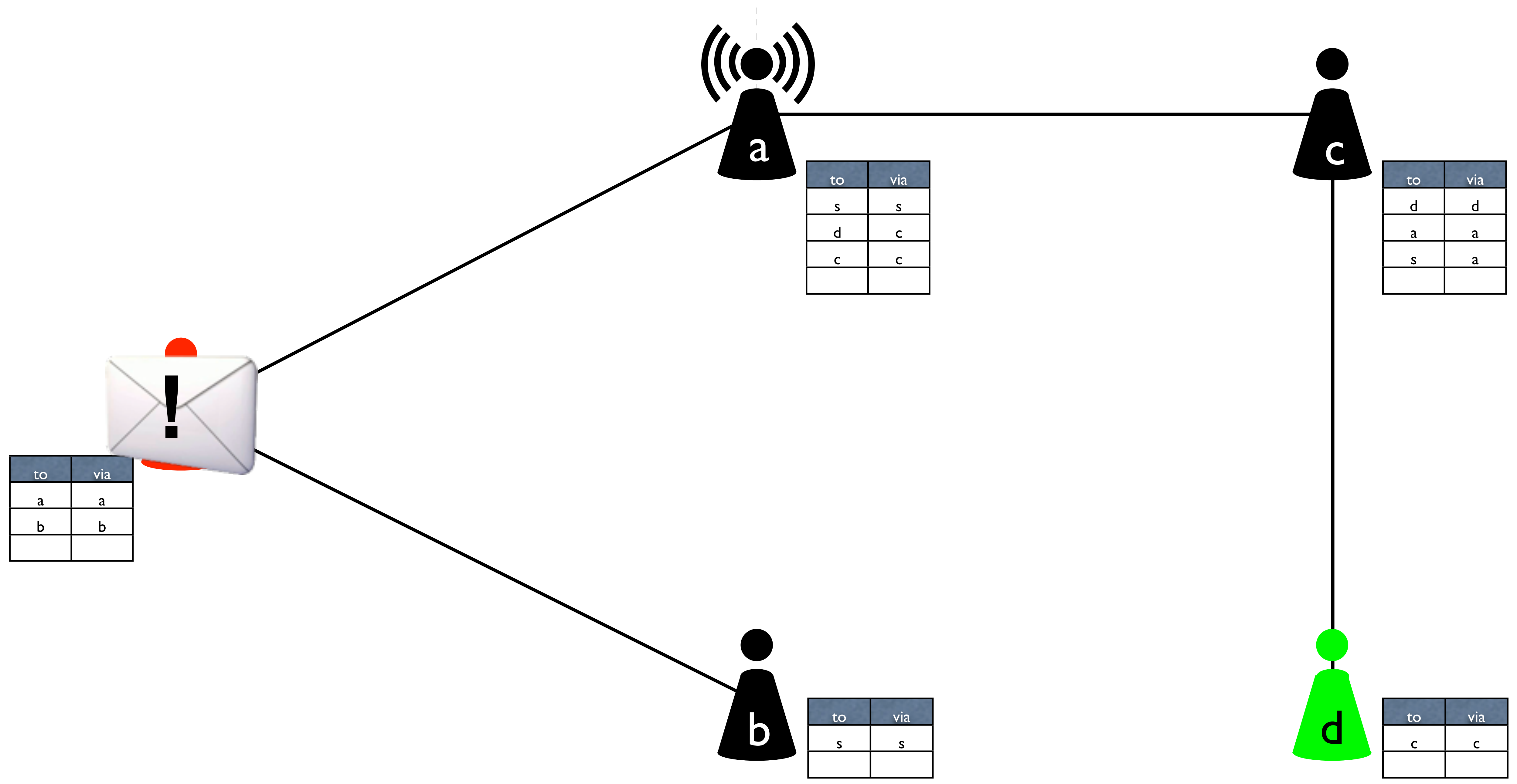
to	via
s	s
d	c
c	c

to	via
d	d
a	a
s	a

to	via
s	s

to	via
c	c

AODV – An Example



to	via
a	a
b	b

to	via
s	s
d	c
c	c

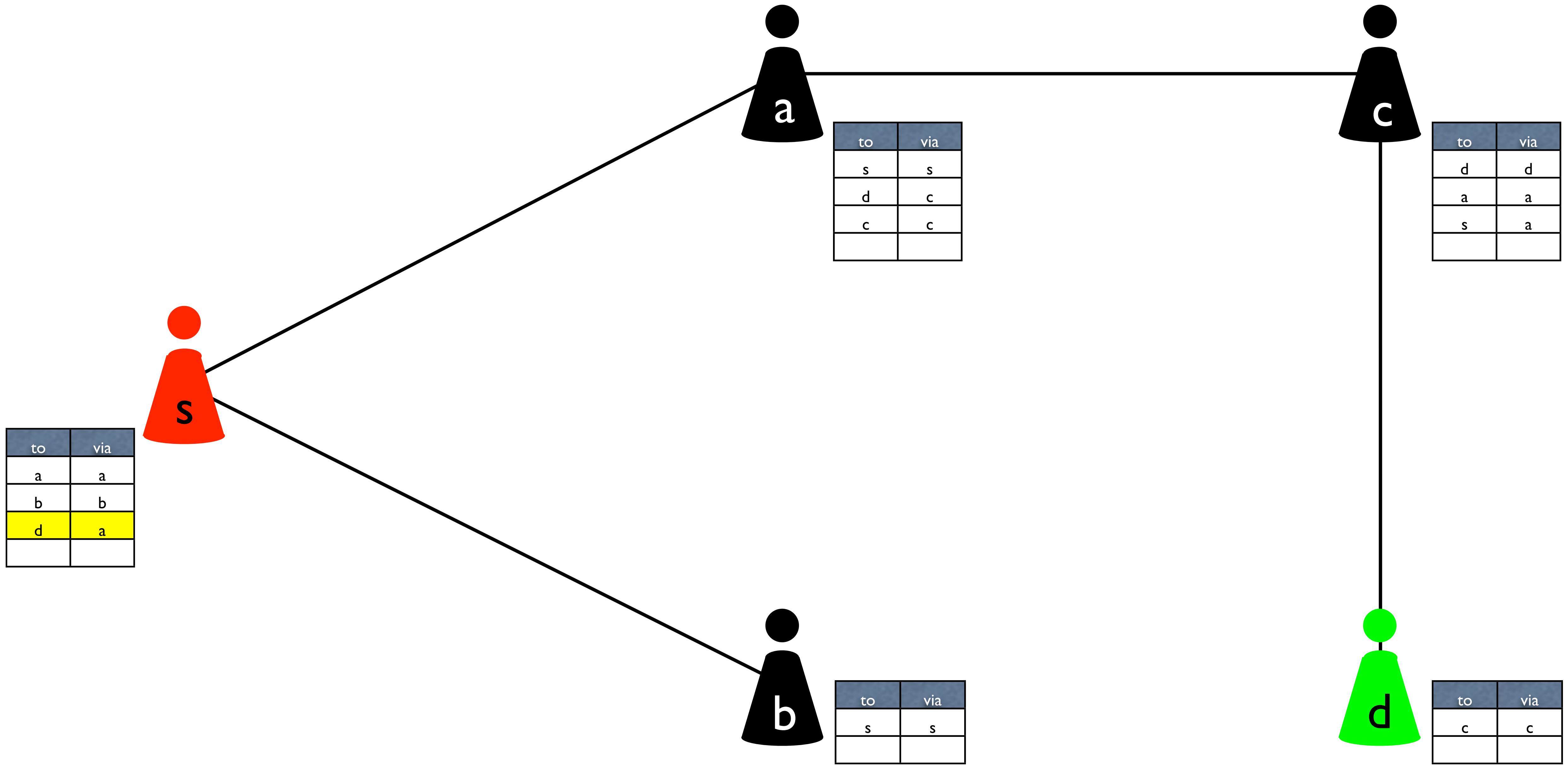
to	via
d	d
a	a
s	a

to	via
s	s

to	via
c	c

a forwards route reply

AODV – An Example



to	via
s	s
d	c
c	c

to	via
d	d
a	a
s	a

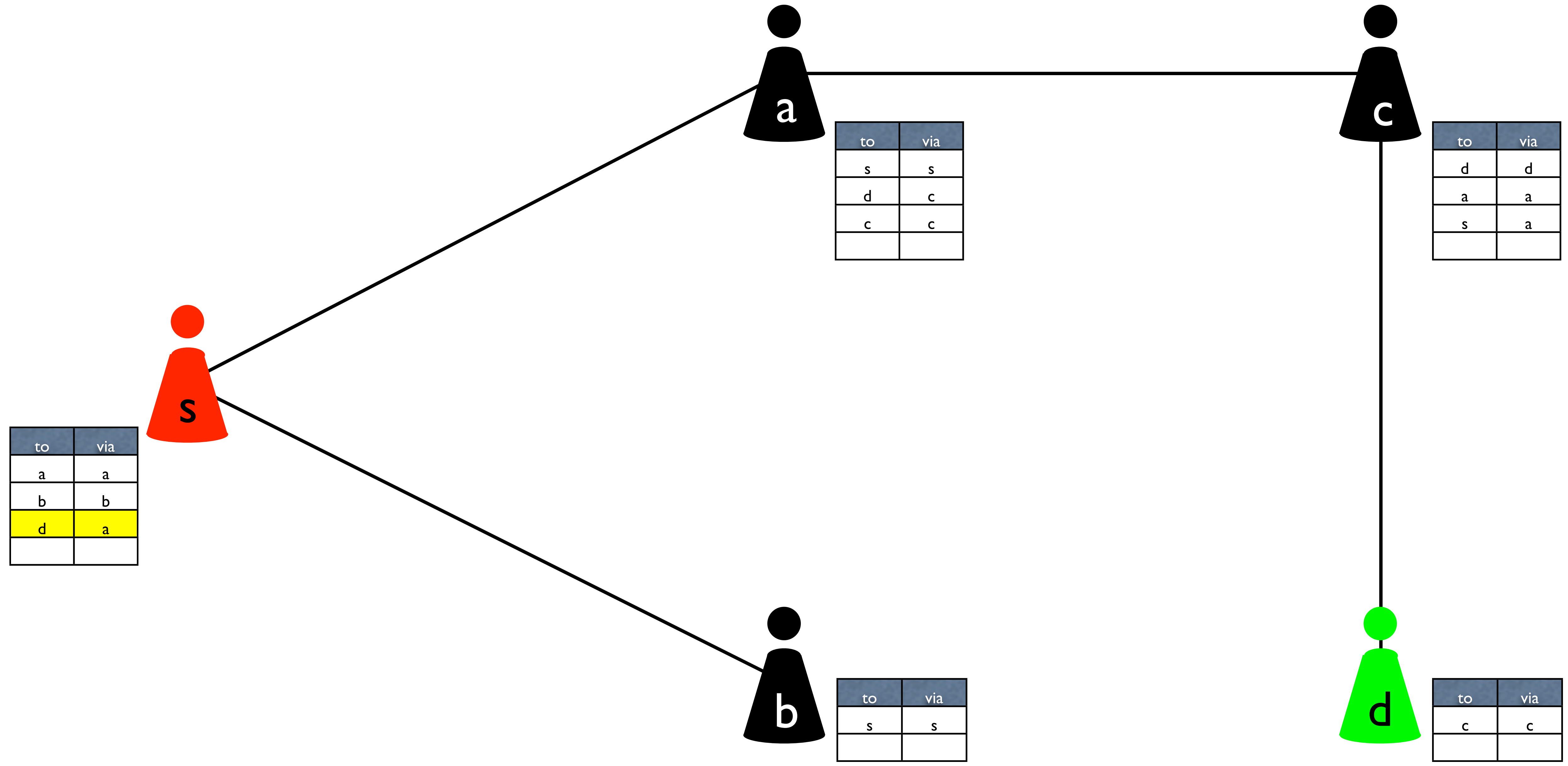
to	via
a	a
b	b
d	a

to	via
s	s

to	via
c	c

a forwards route reply

AODV – An Example



to	via
a	a
b	b
d	a

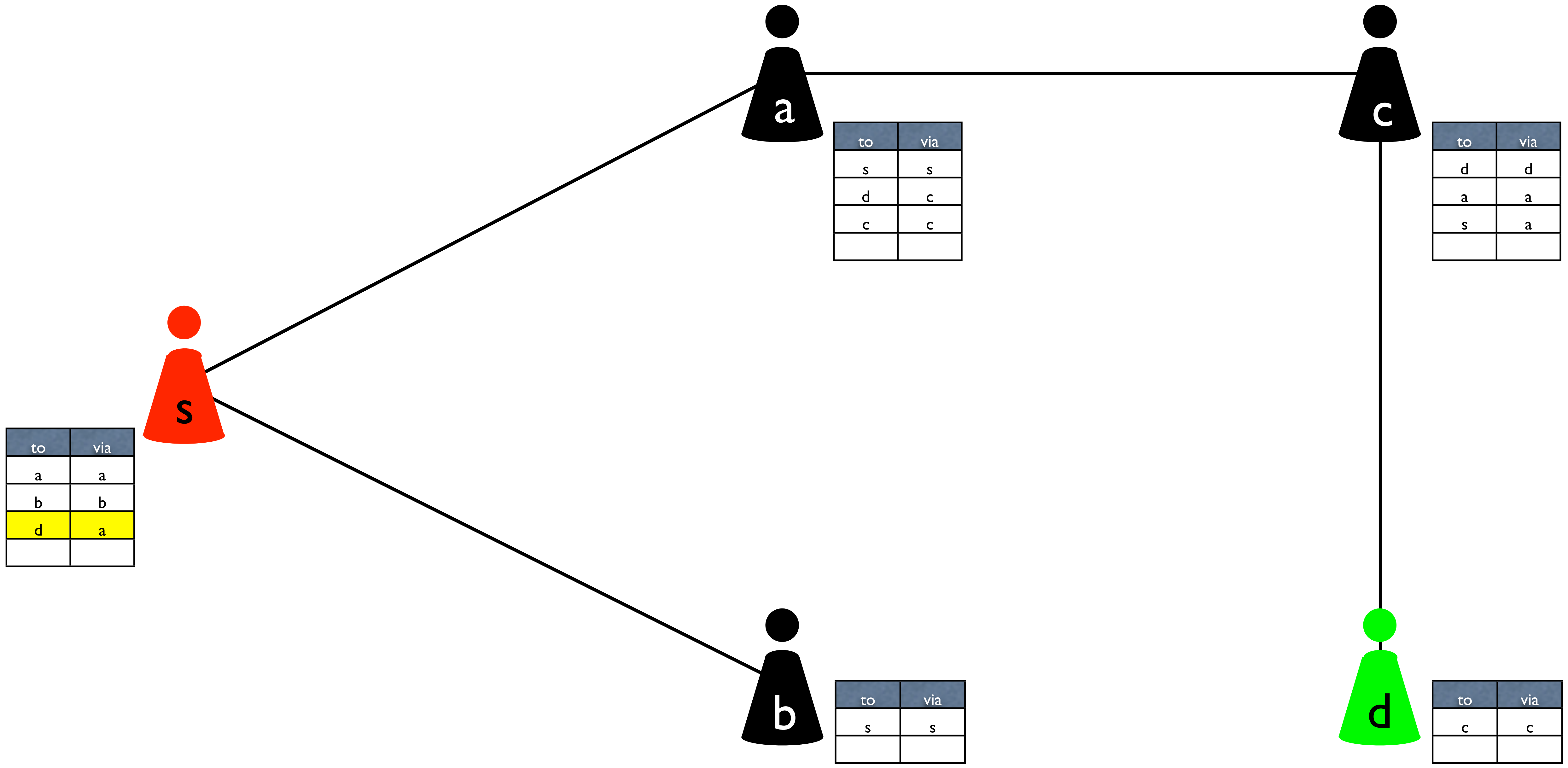
to	via
s	s
d	c
c	c

to	via
d	d
a	a
s	a

to	via
s	s

to	via
c	c

AODV – An Example



s:

to	via
a	a
b	b
d	a

a:

to	via
s	s
d	c
c	c

c:

to	via
d	d
a	a
s	a

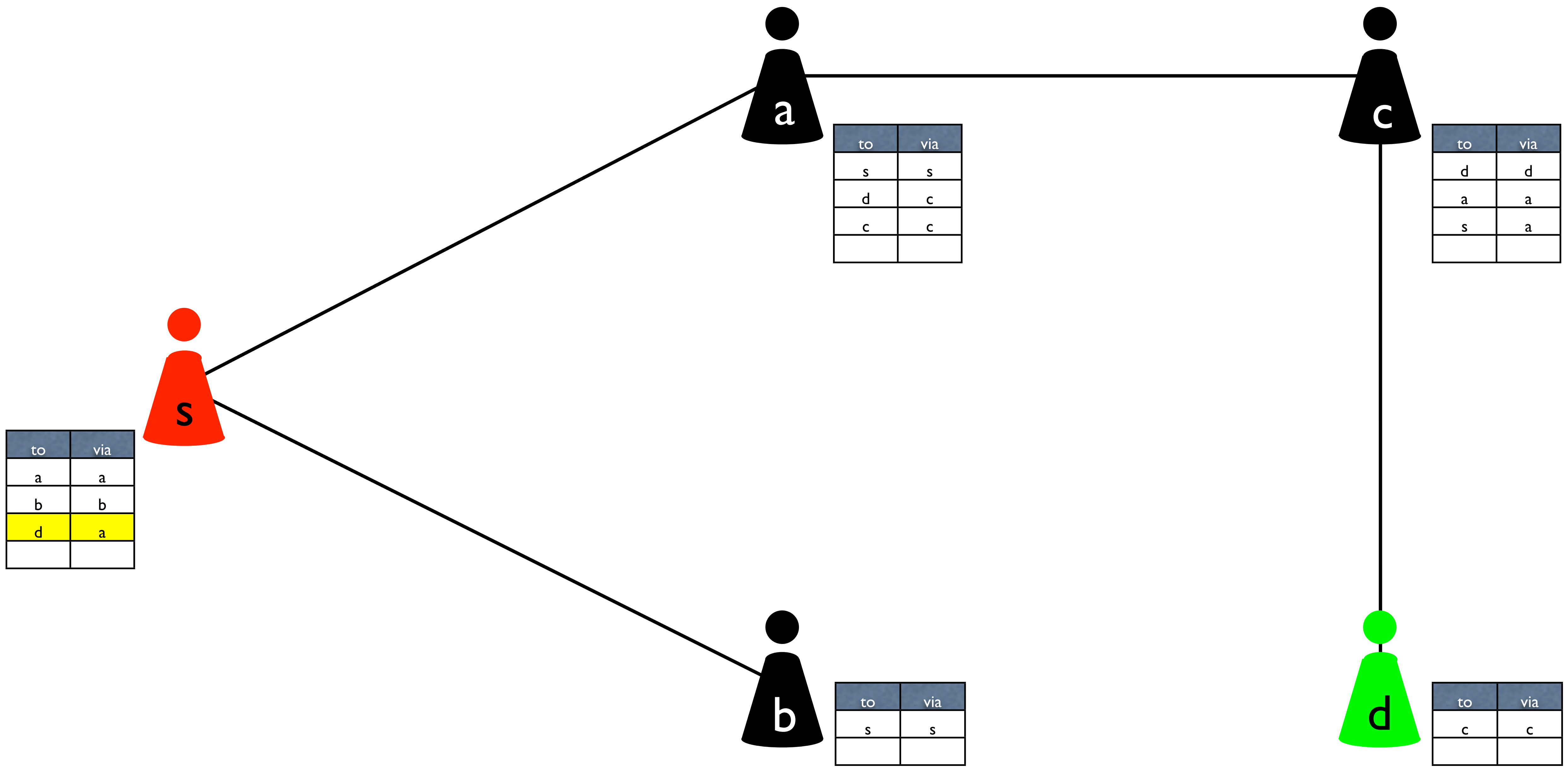
b:

to	via
s	s

d:

to	via
c	c

AODV – An Example



to	via
s	s
d	c
c	c

to	via
d	d
a	a
s	a

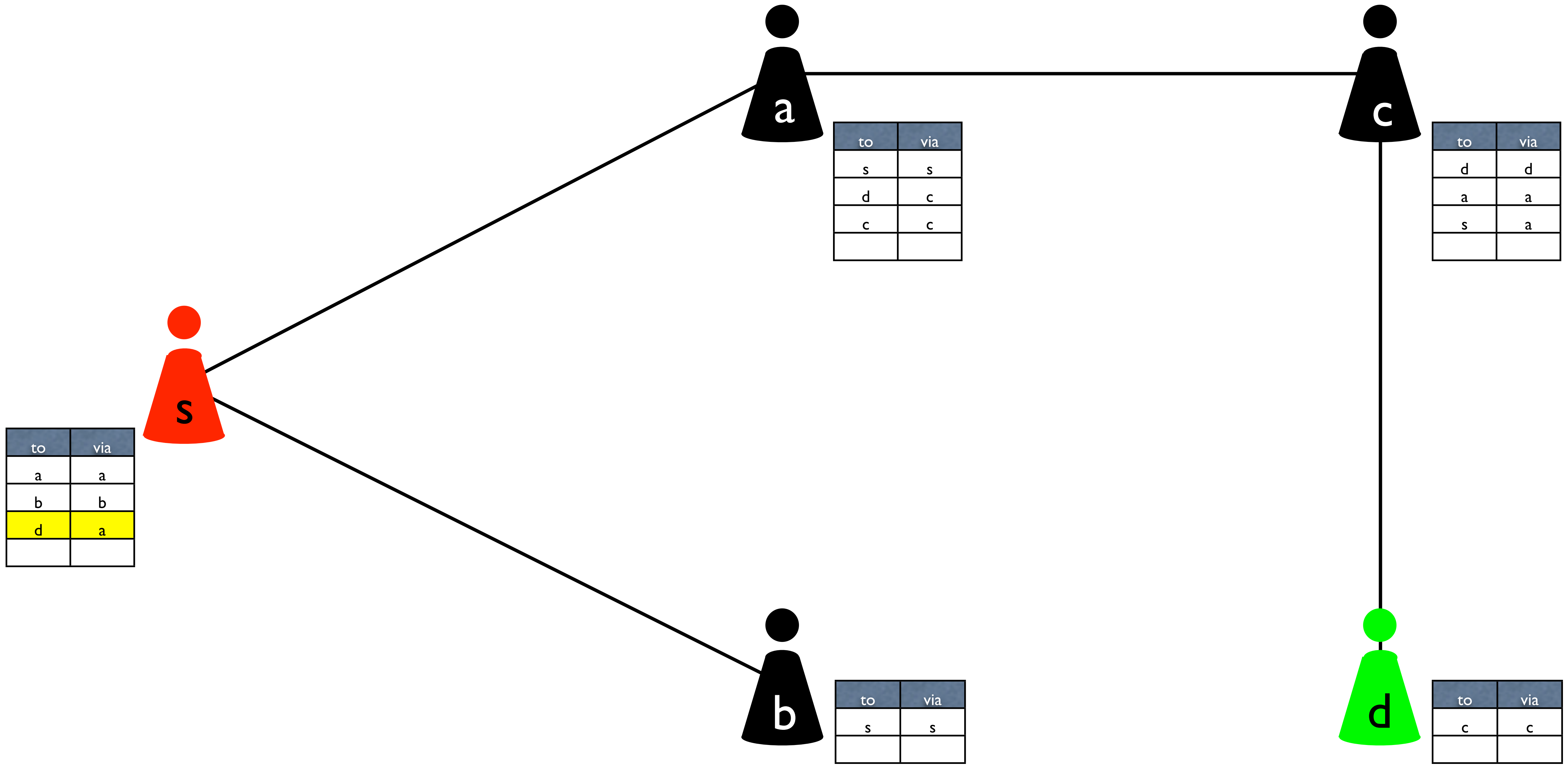
to	via
a	a
b	b
d	a

to	via
s	s

to	via
c	c

s has found a route to d

AODV – An Example



s

to	via
a	a
b	b
d	a

a

to	via
s	s
d	c
c	c

c

to	via
d	d
a	a
s	a

b

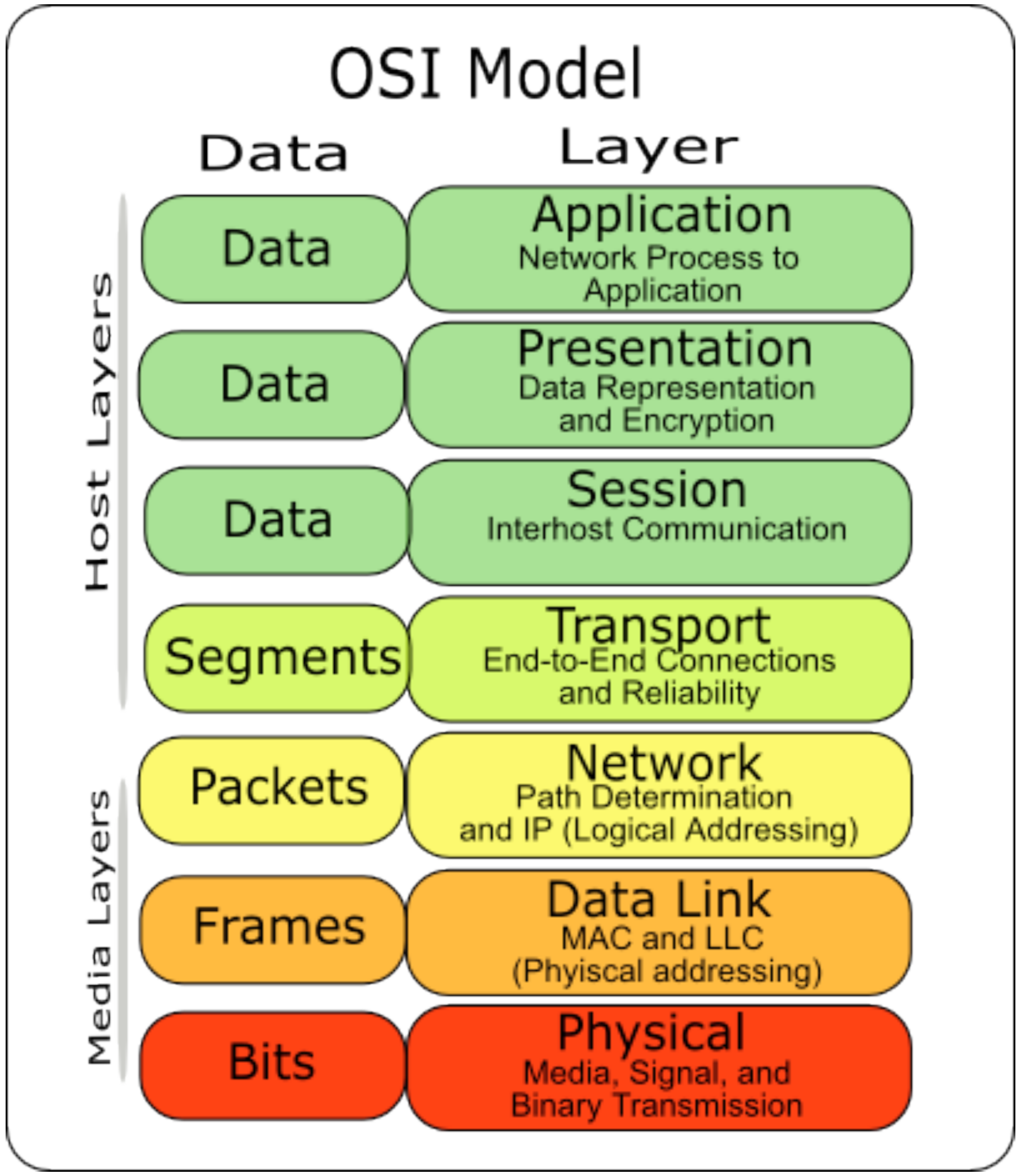
to	via
s	s

d

to	via
c	c

s has found a route to d

Different Network Layers


OSI Model
Data
Layer
Host Layers
Media Layers
Data
Application
Network Process to Application
Data
Presentation
Data Representation and Encryption
Data
Session
Interhost Communication
Segments
Transport
End-to-End Connections and Reliability
Packets
Network
Path Determination and IP (Logical Addressing)
Frames
Data Link
MAC and LLC (Phyiscal addressing)
Bits
Physical
Media, Signal, and Binary Transmission

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