An Algebra of Routing Tables

Peter Höfner

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

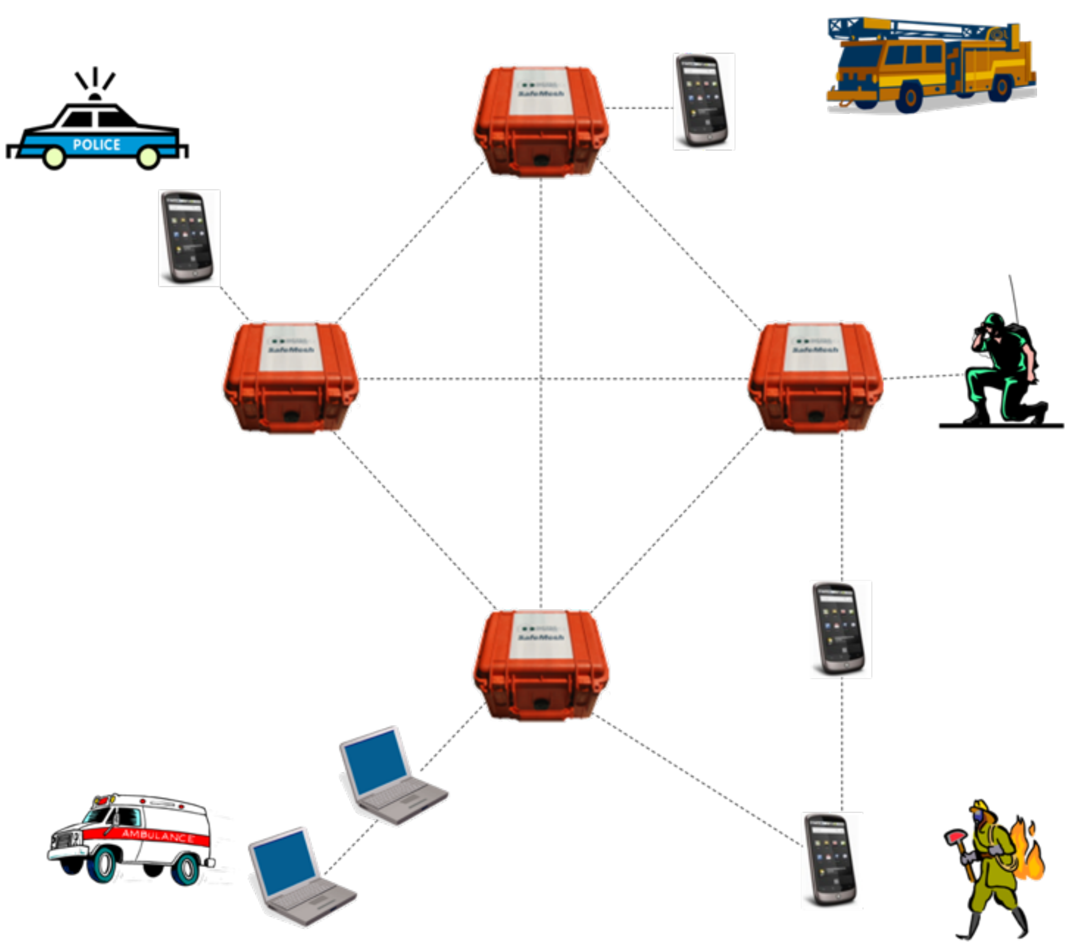
Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Members
NICTA Partners

What is the Problem?

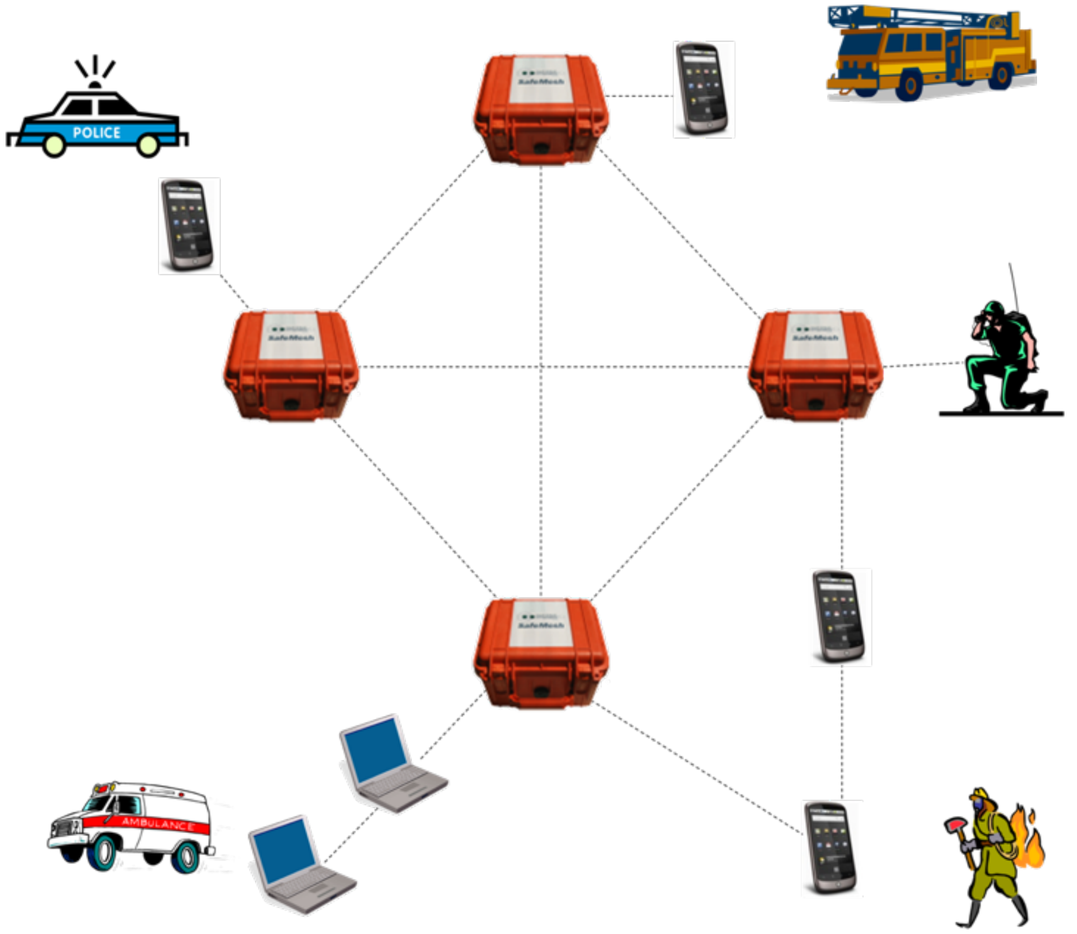
NICTA

- 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
  - ...



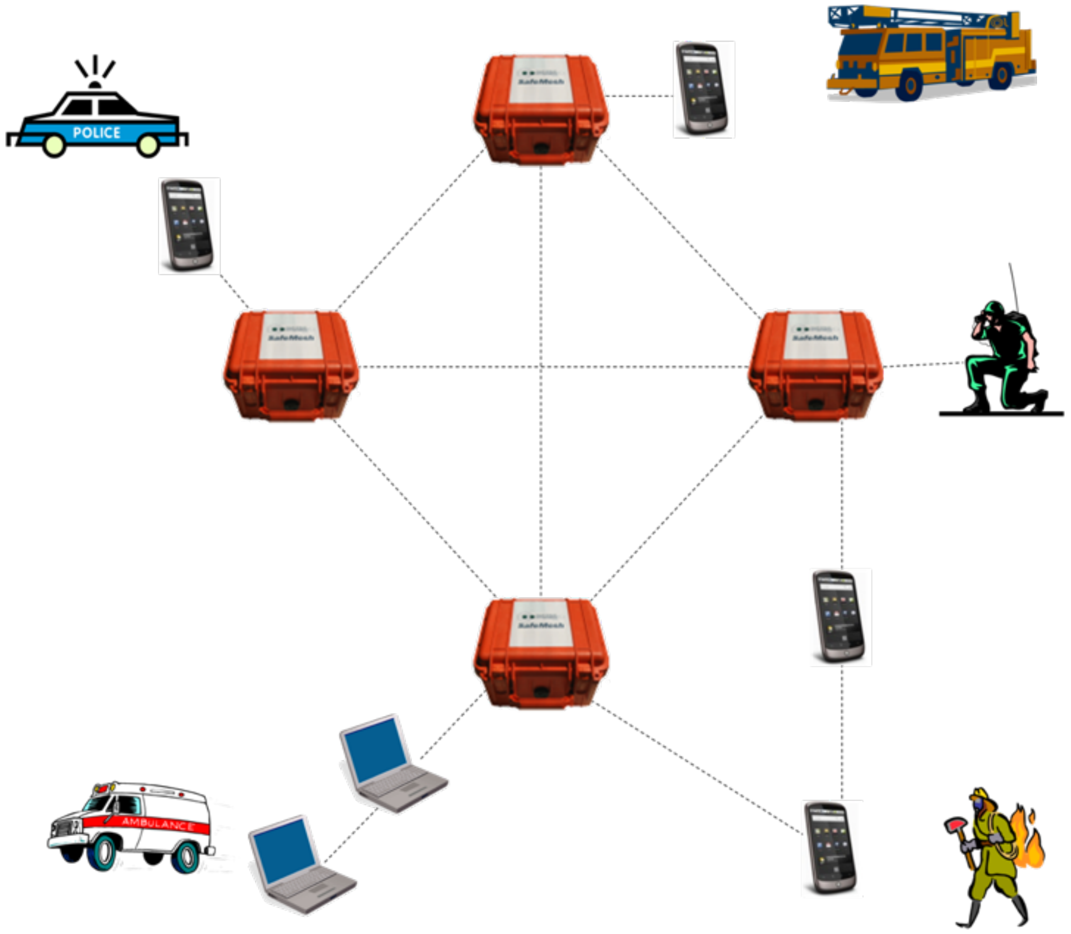
What is the Problem?

- 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

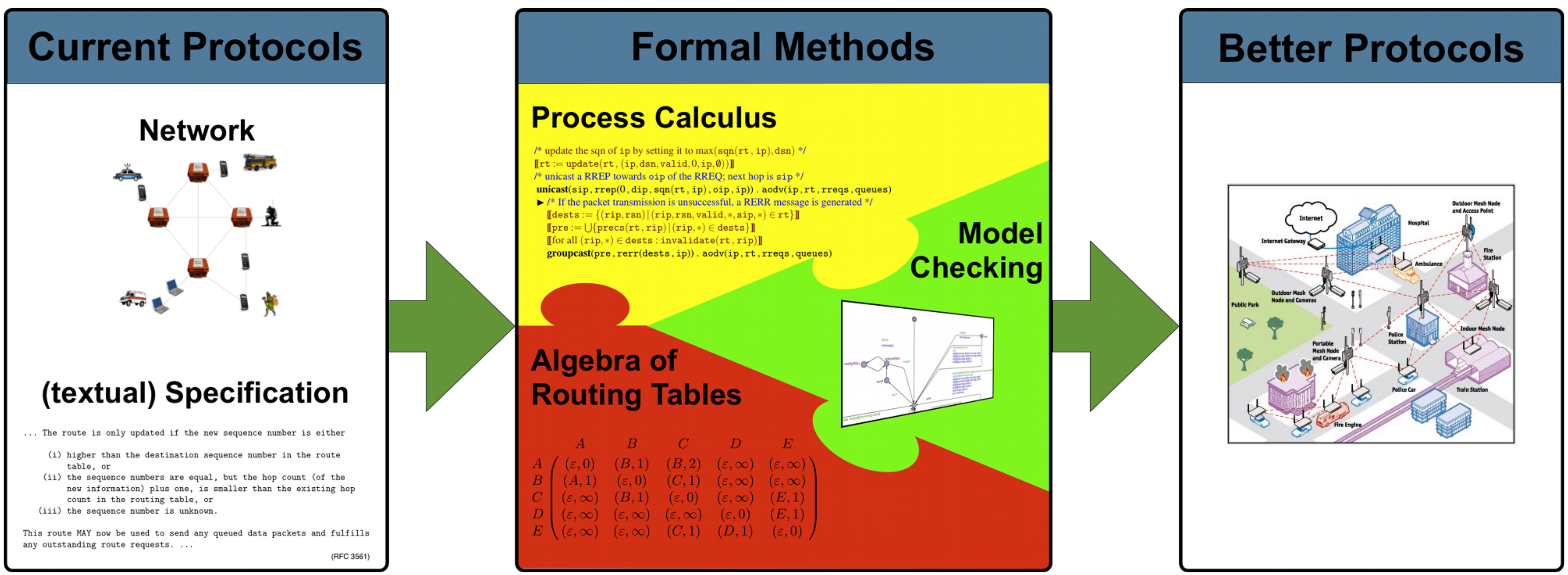
- Goal
  - model, analyse, verify and increase the performance of wireless mesh protocols

- 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



Towards an Algebra of Routing Tables

- Routing protocols
  - find a 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

Ad Hoc On-Demand Distance Vector Protocol

- Routing protocol for WMNs

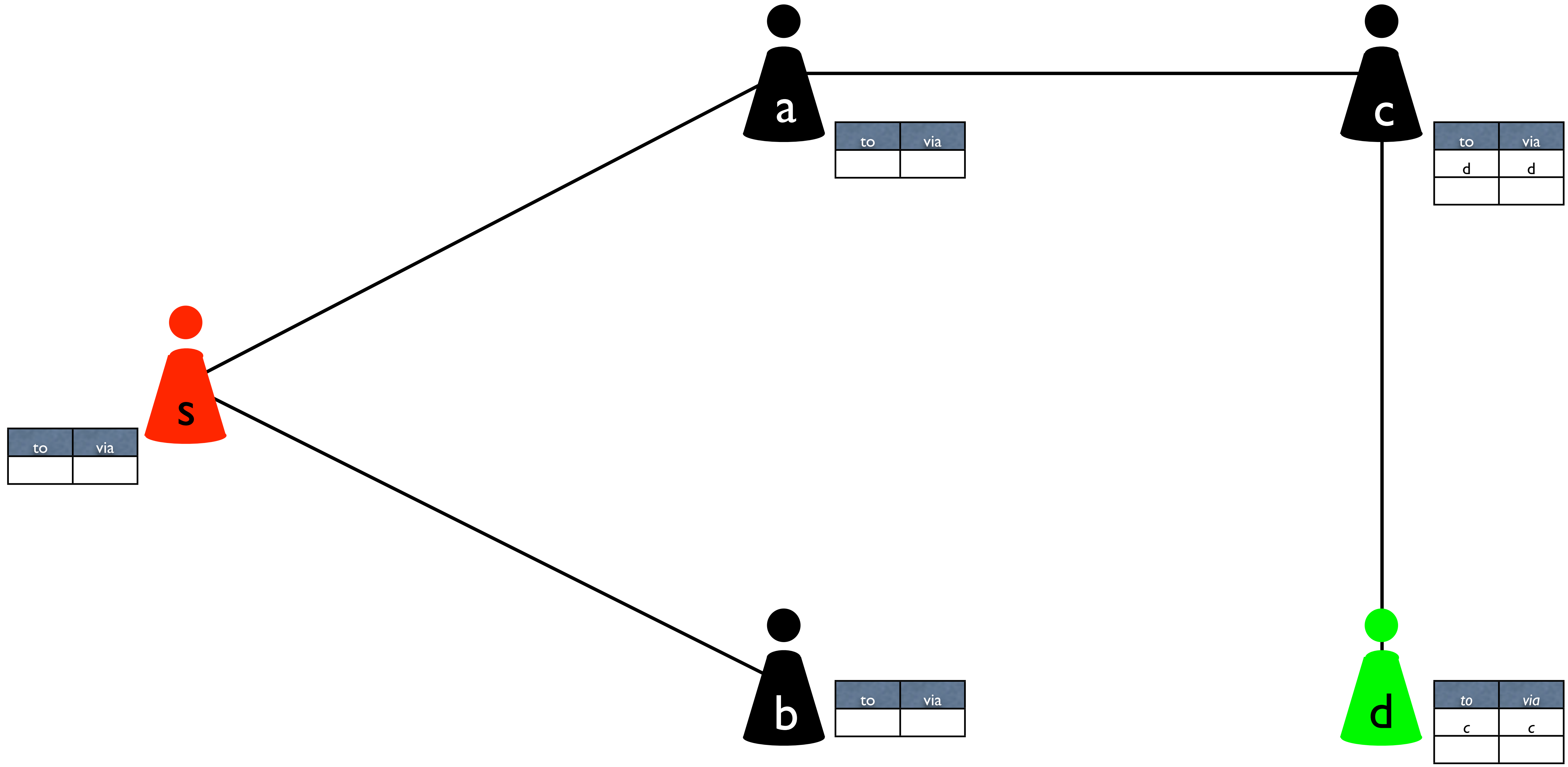
- Ad hoc (network is not static)
- On-Demand (routes are established when needed)
- Distance (metric is hop count)
- Vector (routing table has the form of a vector)

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

AODV – An Example

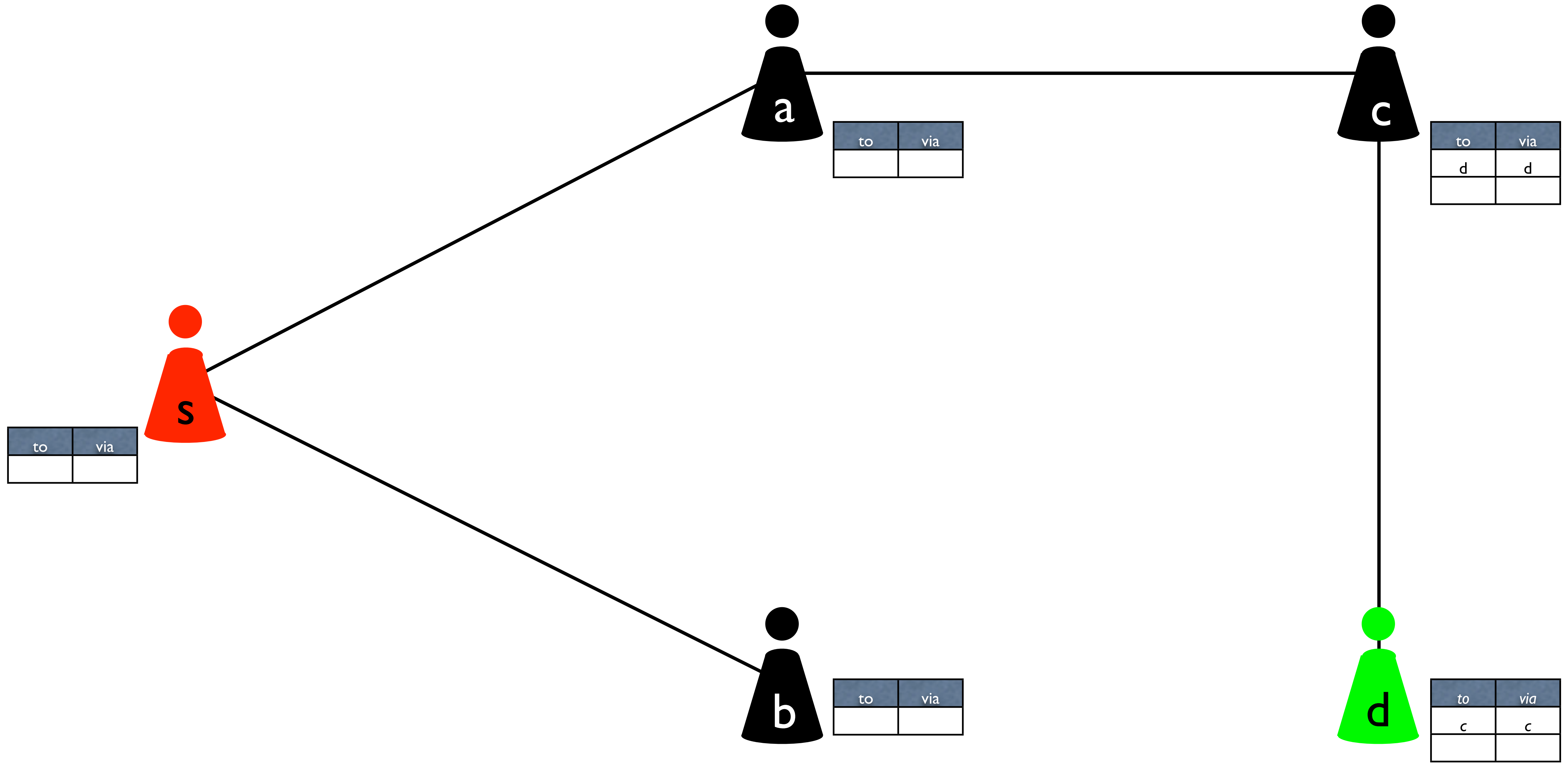
- AODV control messages
  - route request (RREQ)
  - route reply (RREP)
  - route error message (RERR)
  - (Hello messages)
- Information at nodes
  - own IP address
  - a local sequence number (freshness/timer)
  - a routing table
    - local knowledge
    - entries: $(\mathtt{dip}\,,\,\mathtt{dsn}\,,\,\mathtt{val}\,,\,\mathtt{hops}\,,\,\mathtt{nhip}\,,\,\mathtt{pre})$
    - special route: $(\mathtt{ip}\,,\,\mathtt{sn}\,,\,\mathtt{val}\,,\,0\,,\,\mathtt{ip}\,,\,\emptyset)$

AODV – An Example

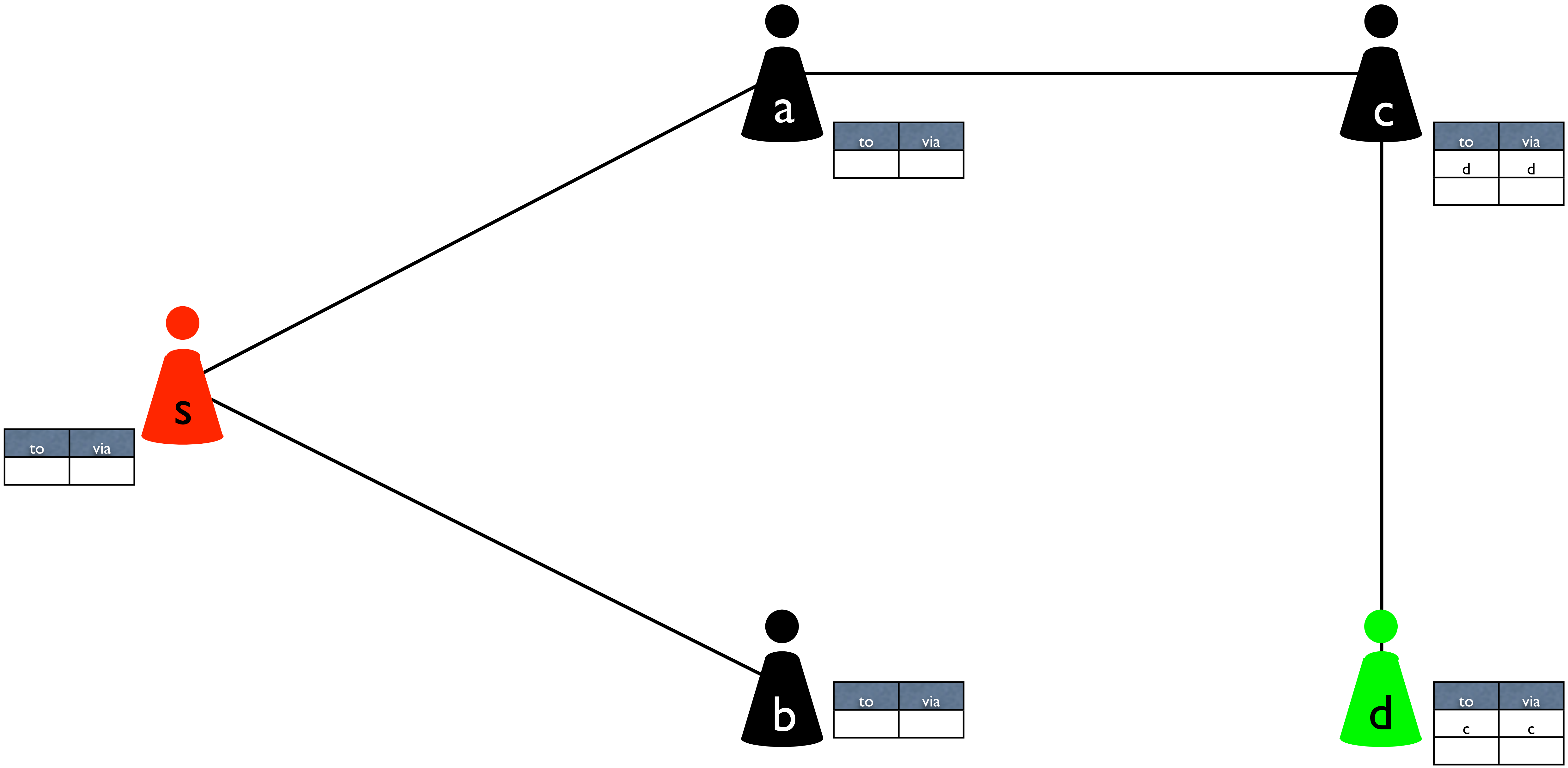


s is looking for a route to d

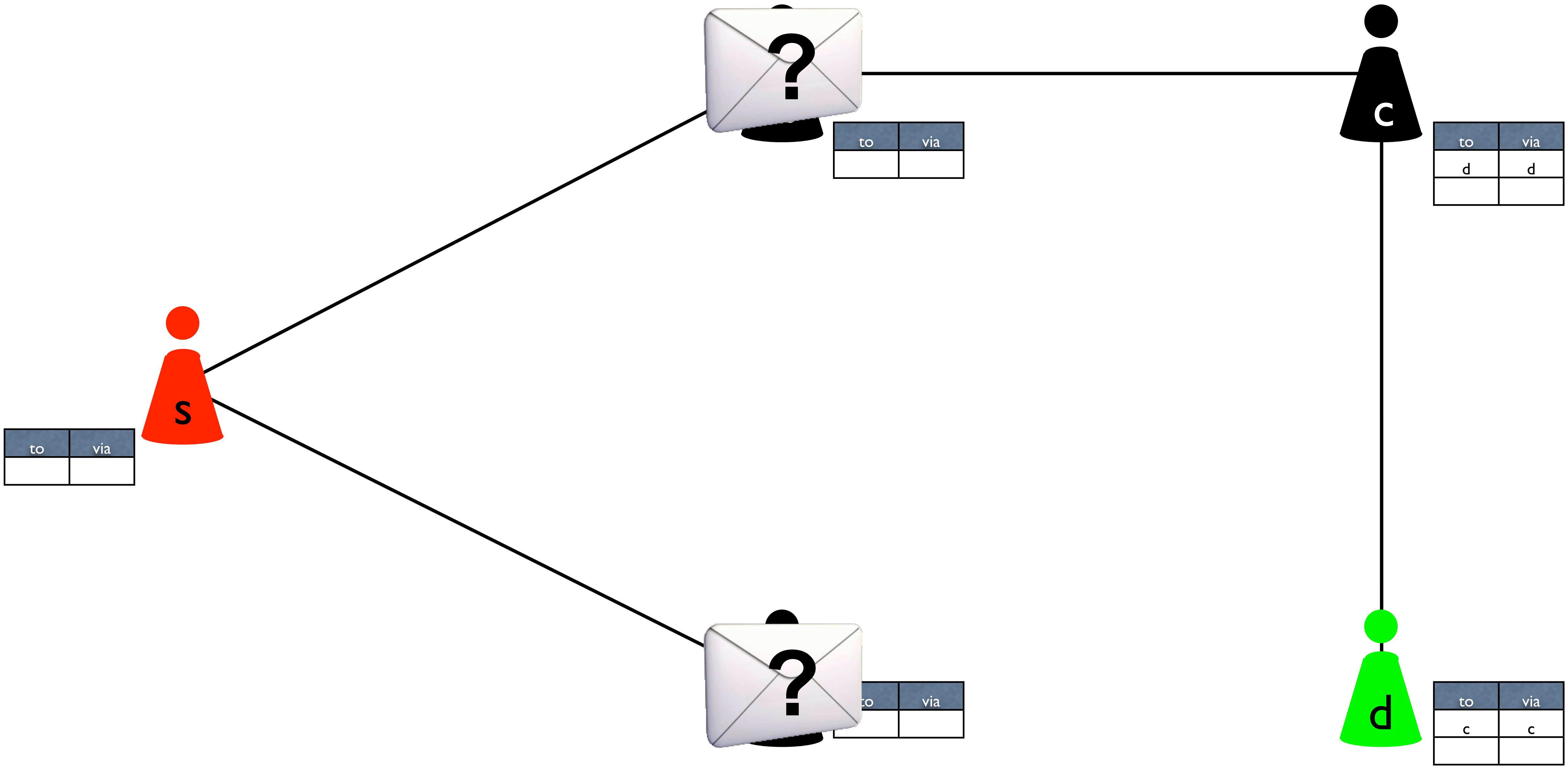
AODV – An Example



AODV – An Example

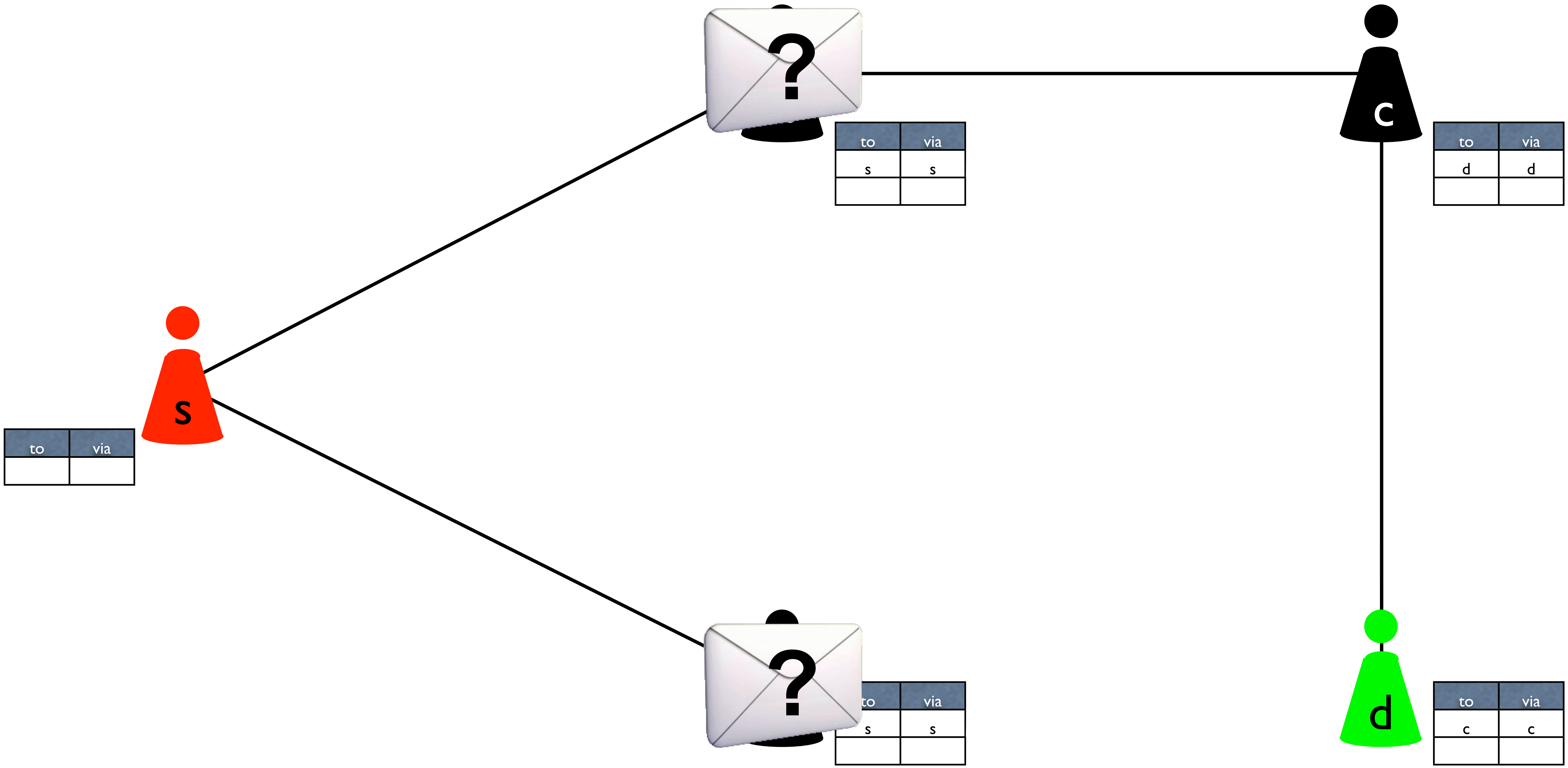


AODV – An Example



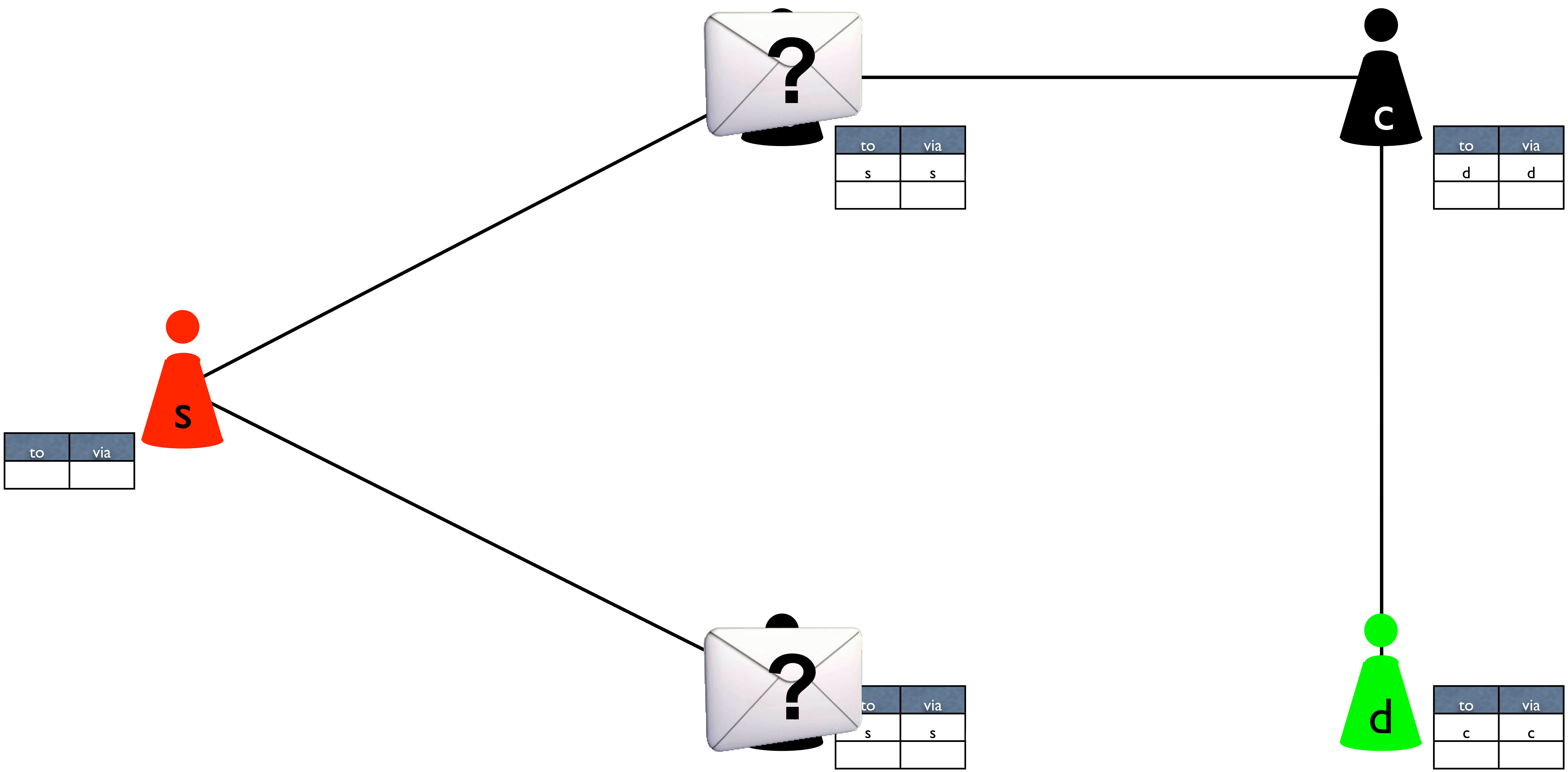
s broadcasts a route request

AODV – An Example

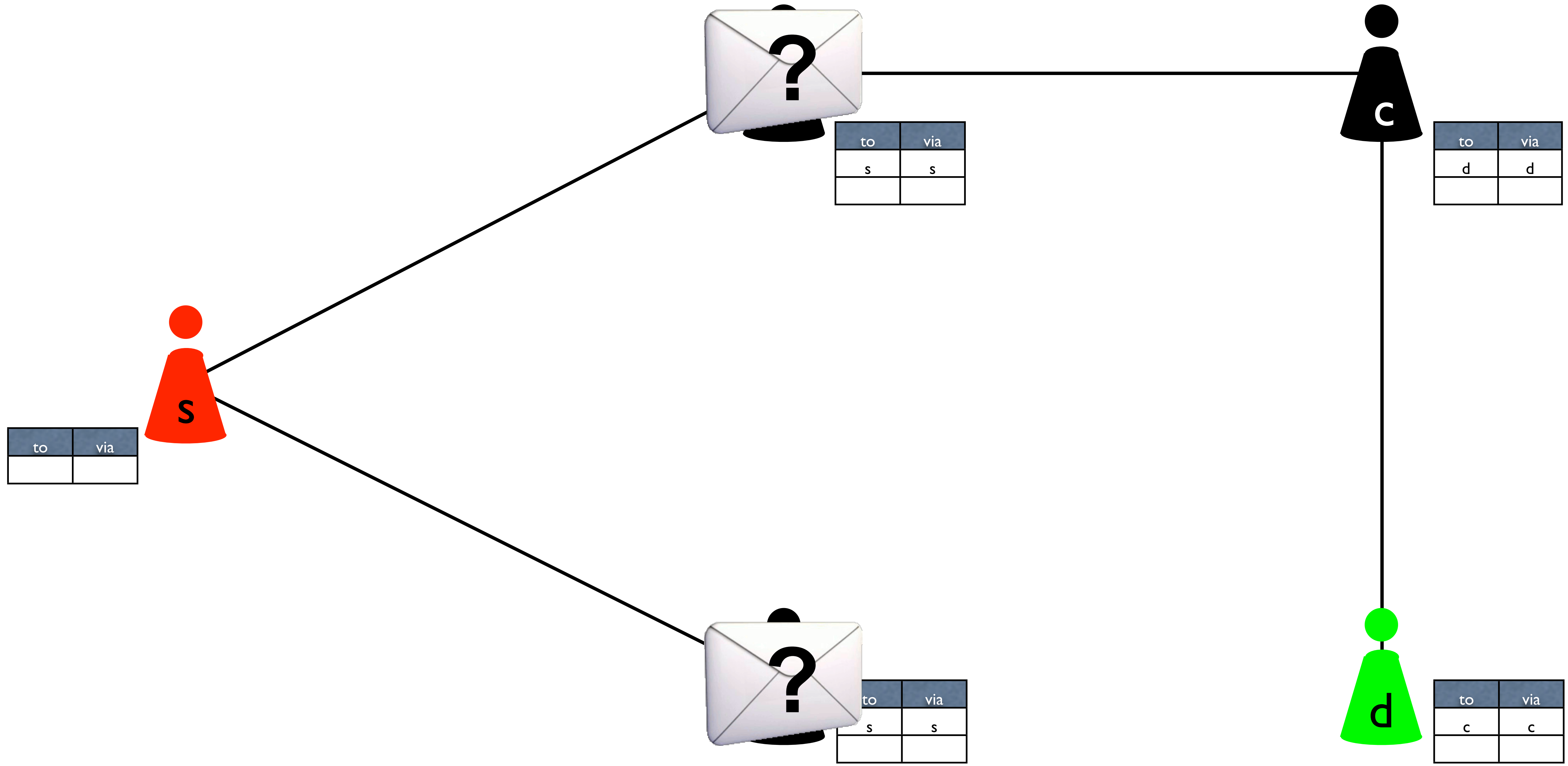


s broadcasts a route request

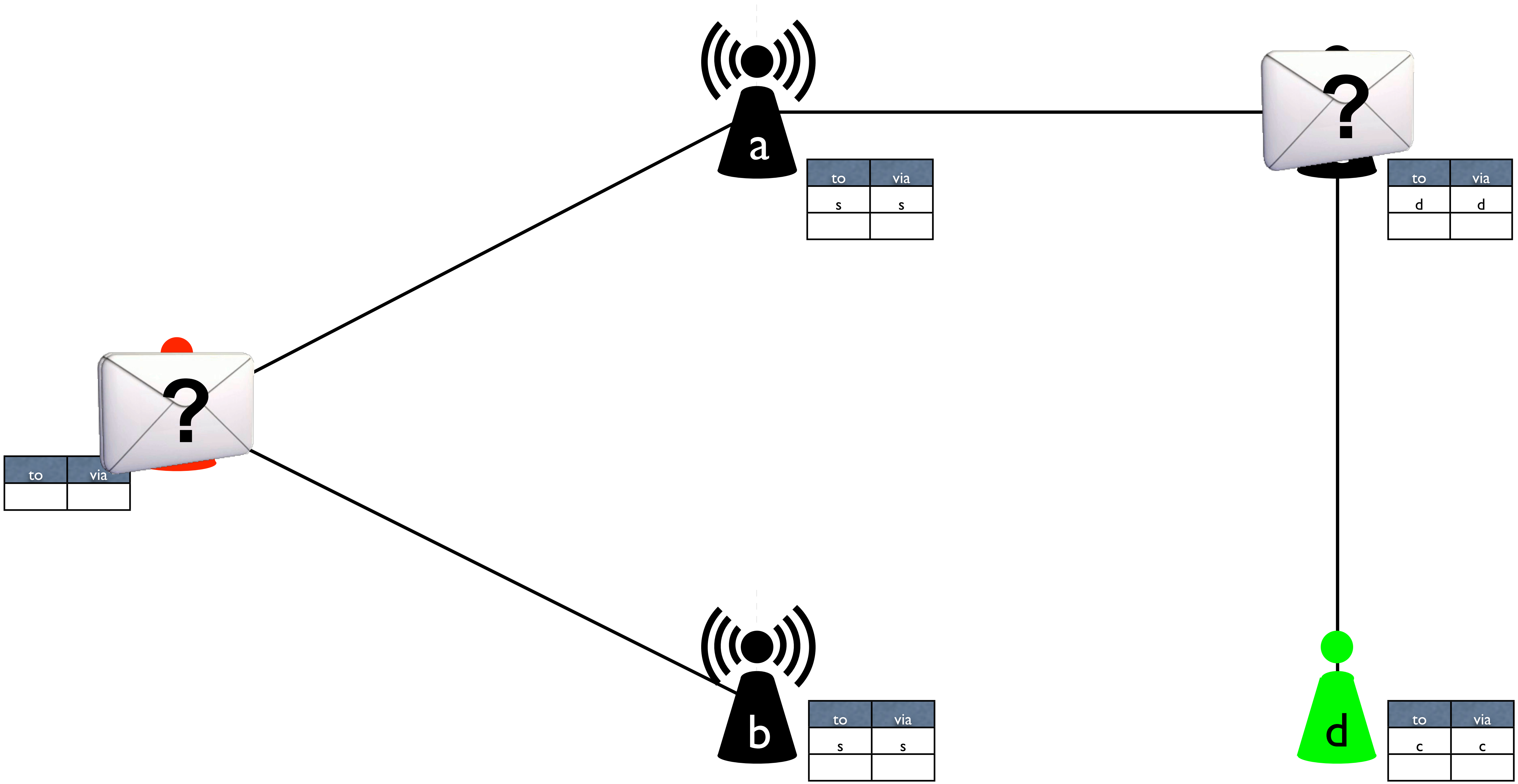
AODV – An Example



AODV – An Example

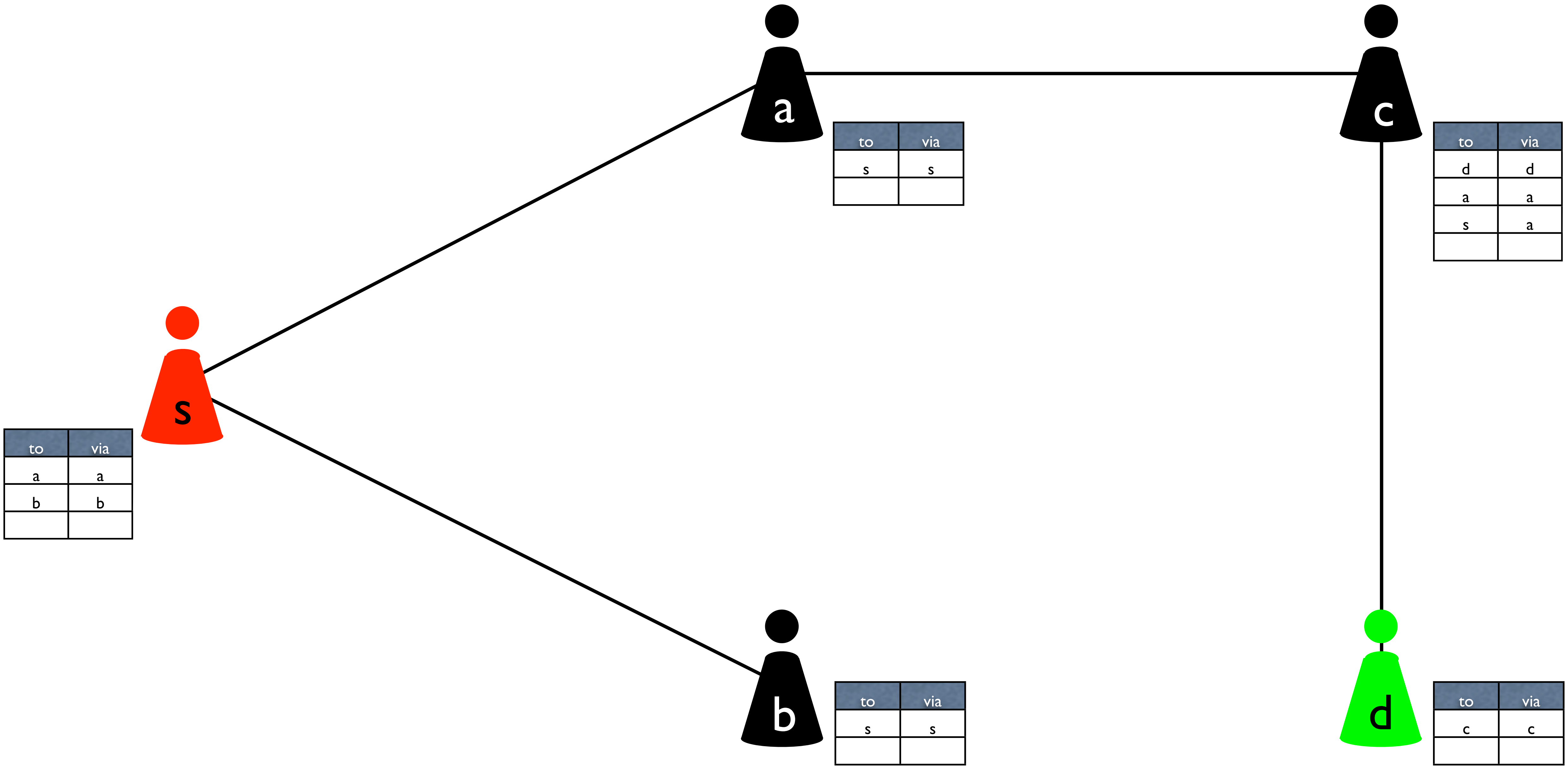


AODV – An Example



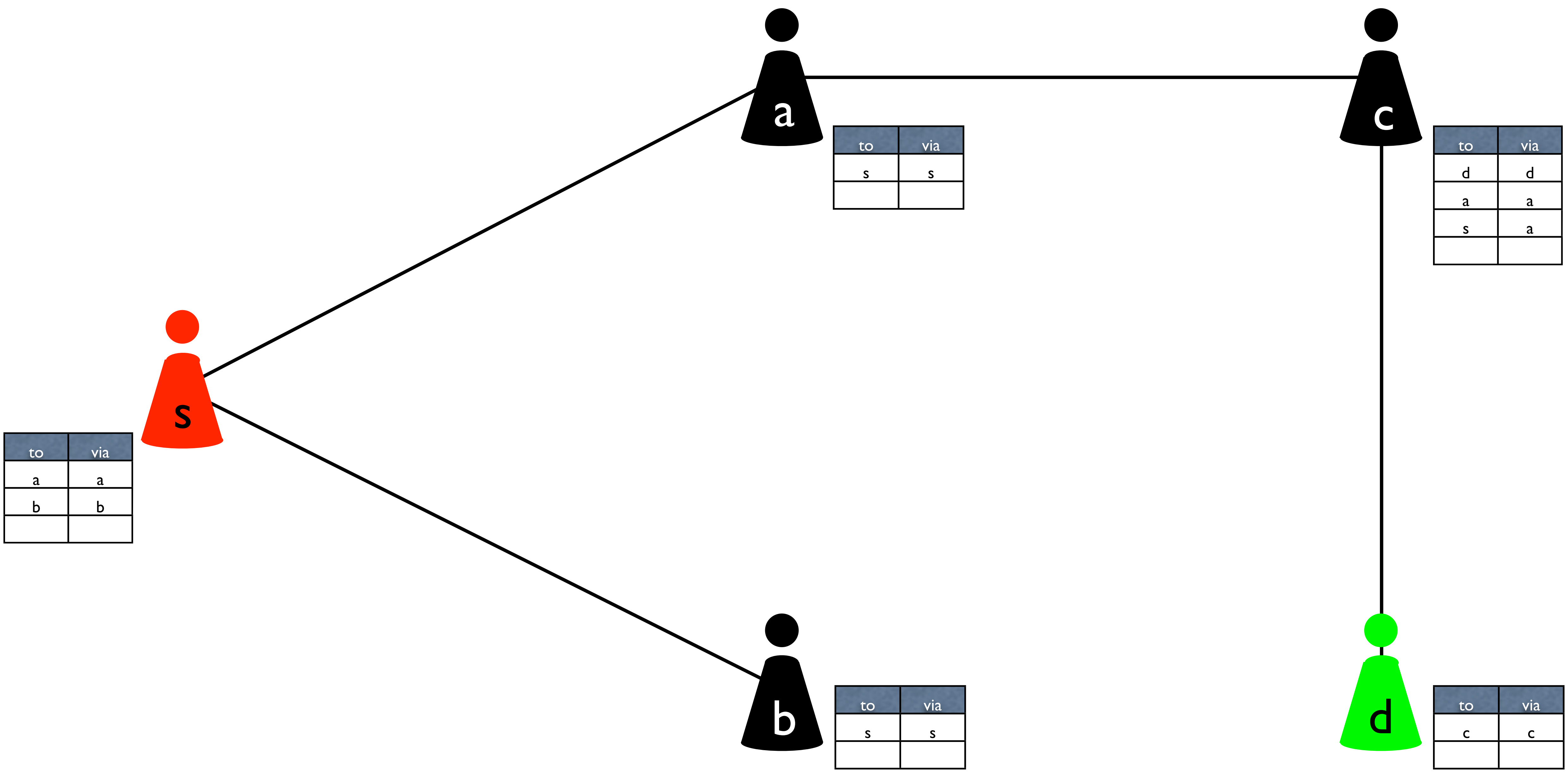
a,b forward the route request

AODV – An Example

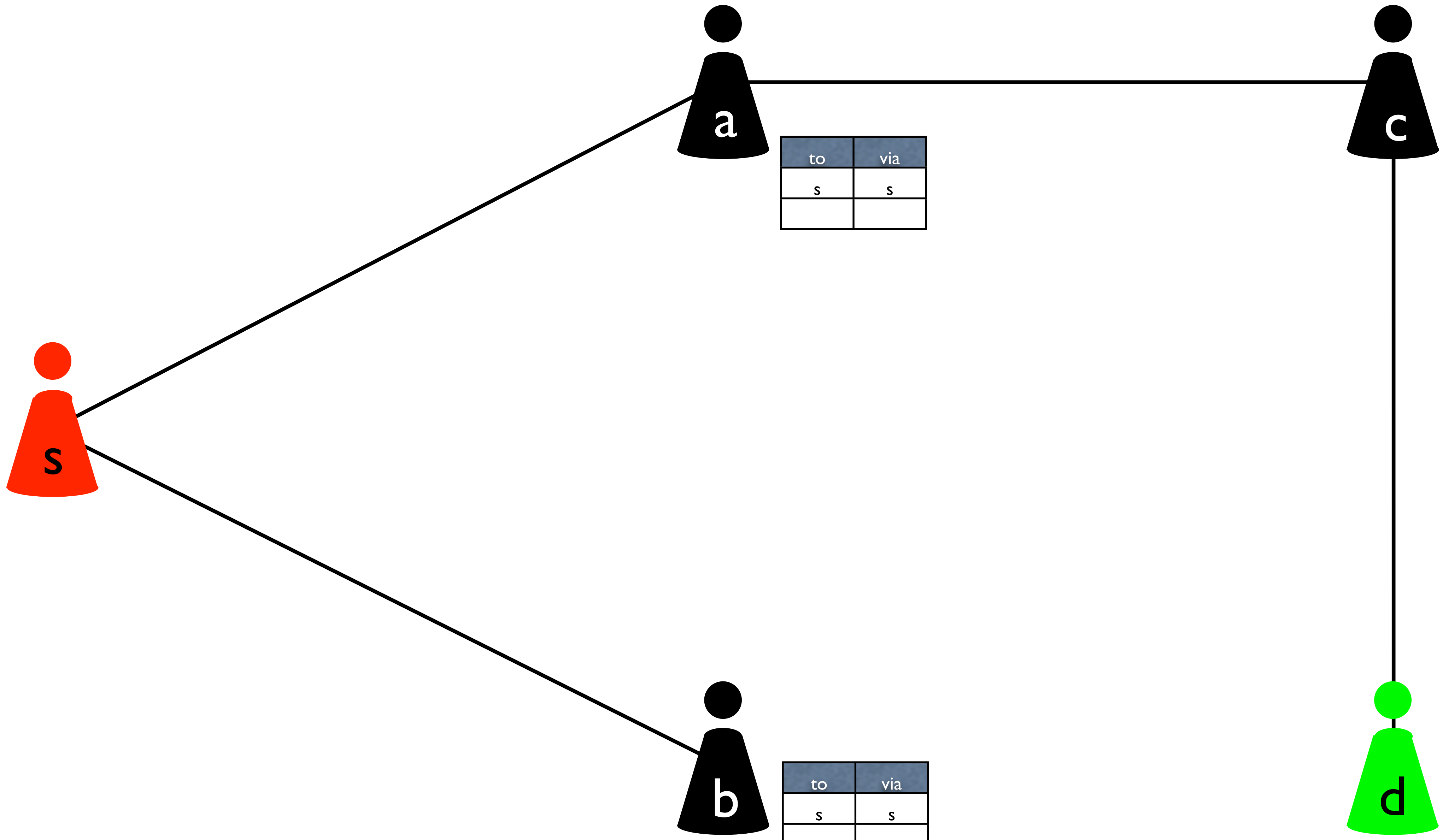


a,b forward the route request

AODV – An Example



AODV – An Example



to	via
s	s

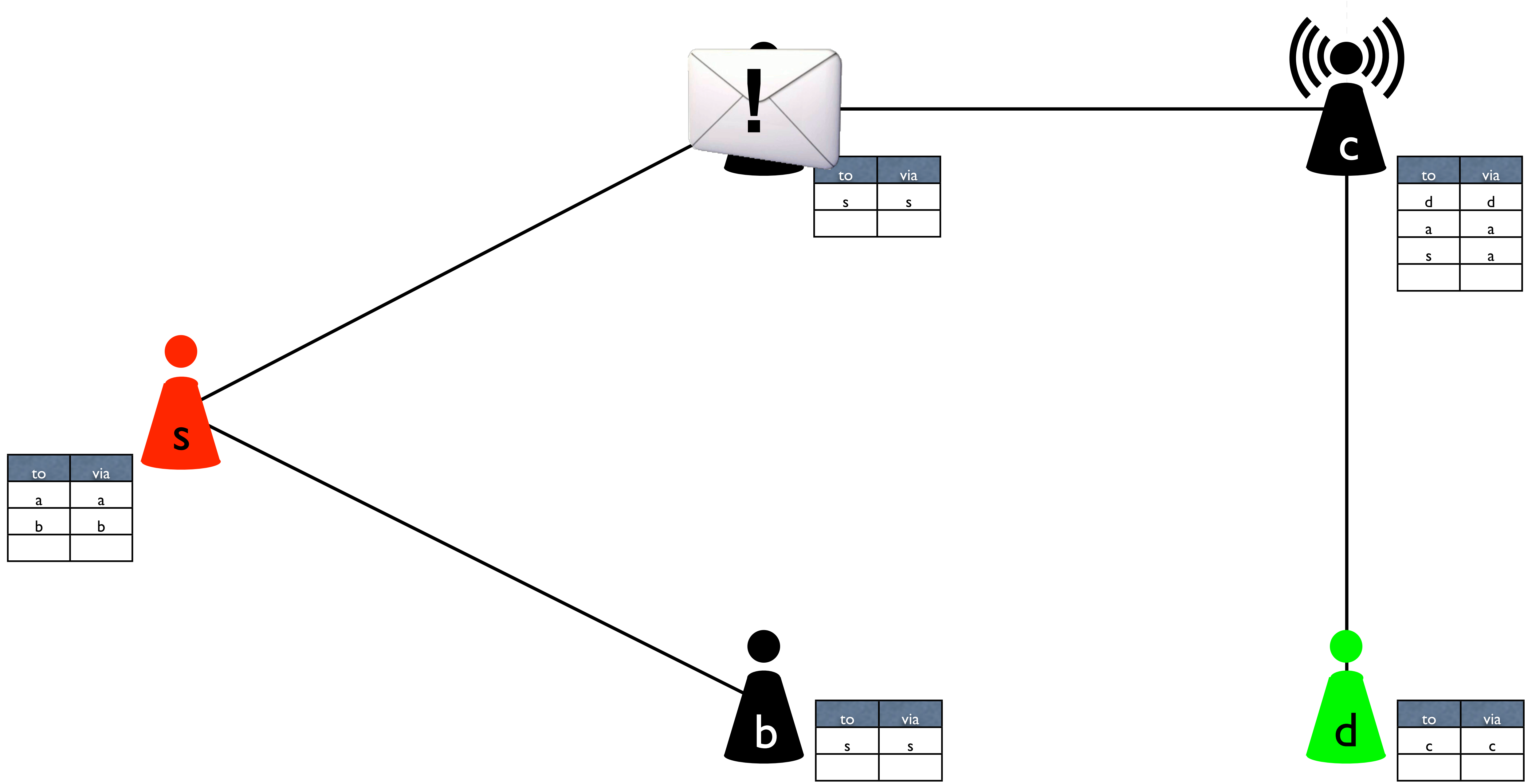
to	via
d	d
a	a
s	a

to	via
a	a
b	b

to	via
s	s

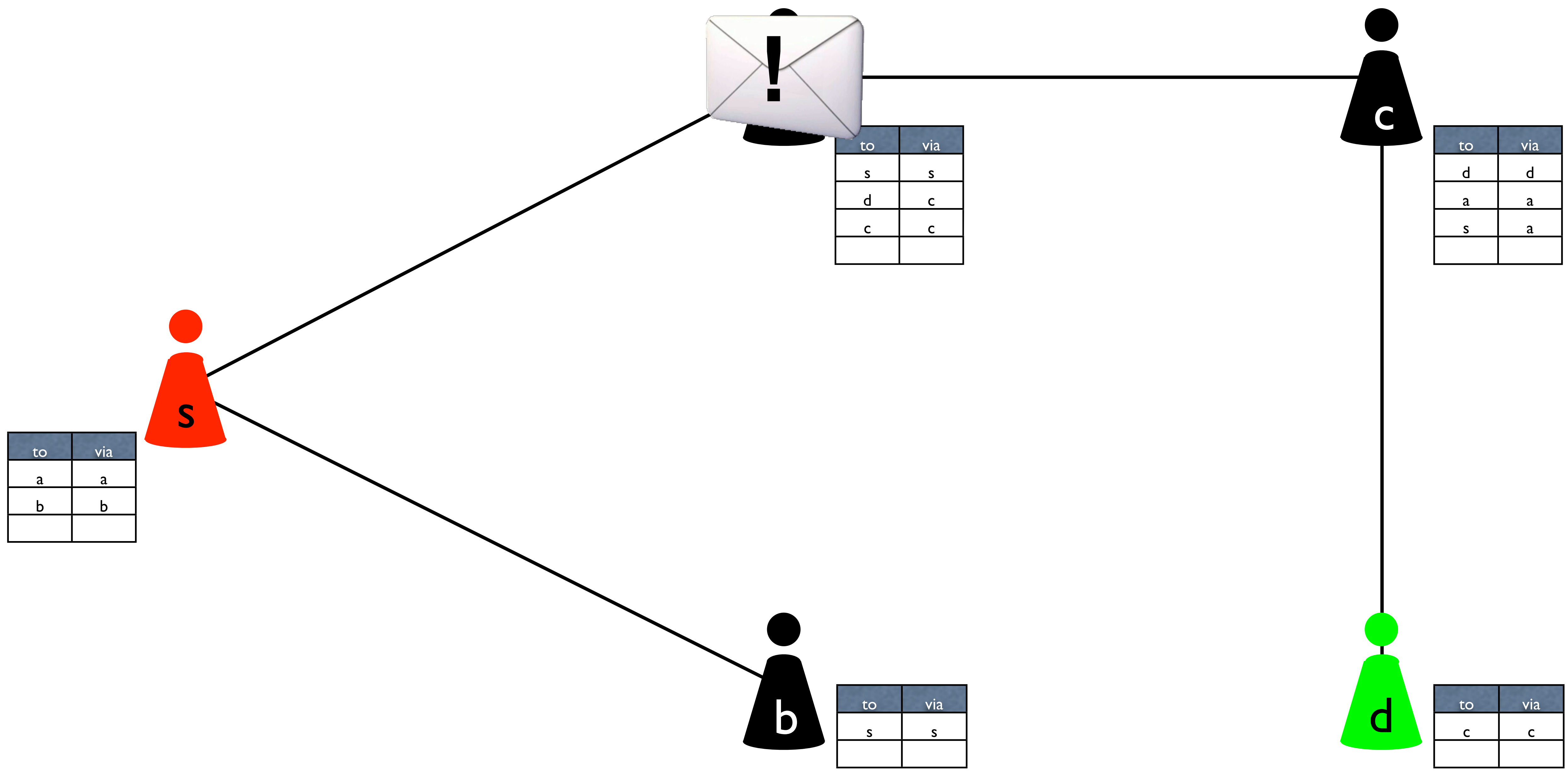
to	via
c	c

AODV – An Example



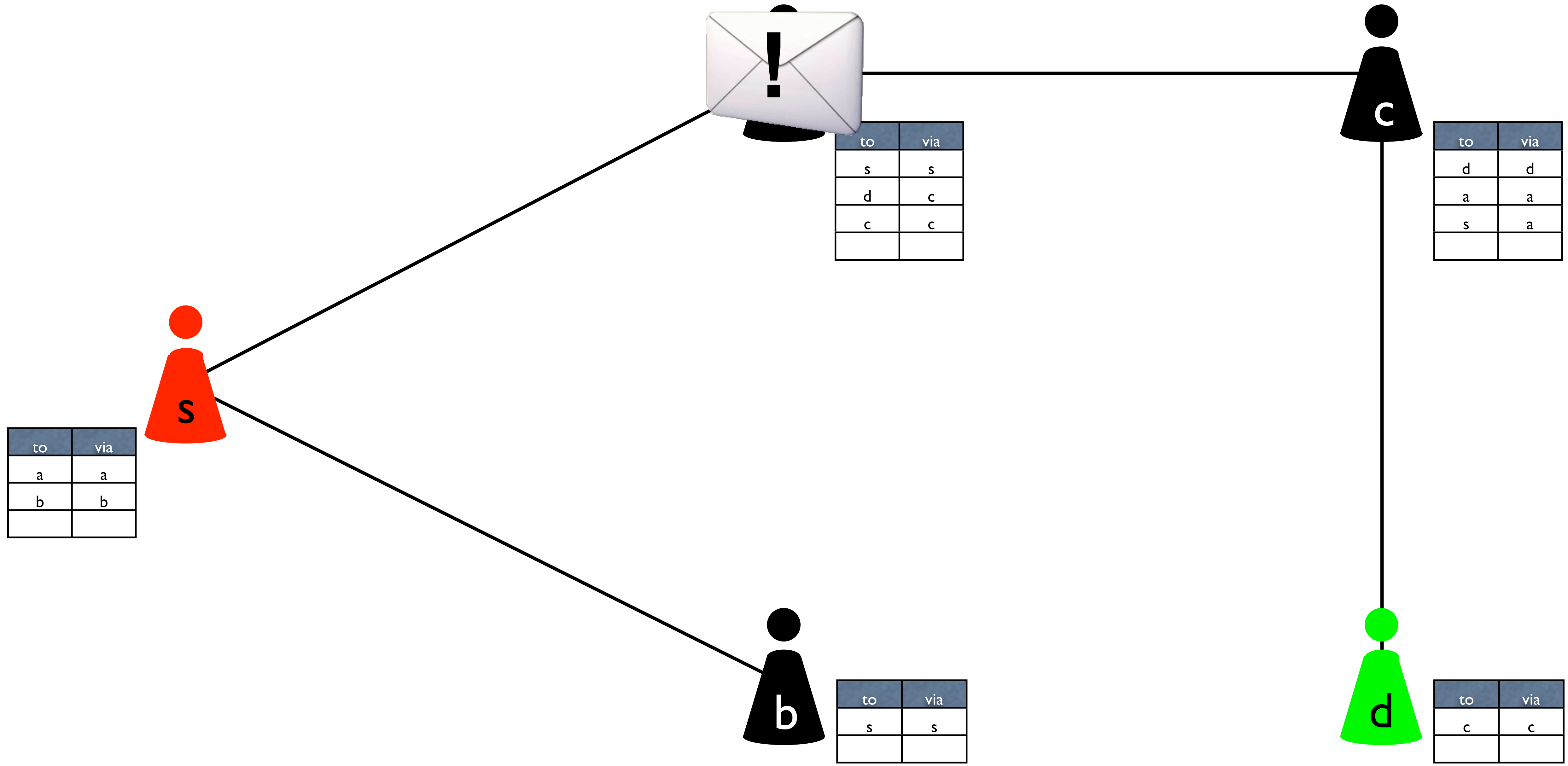
c has information about d
c answers route request and sends reply

AODV – An Example



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

AODV – An Example



to	via
s	s
d	c
c	c

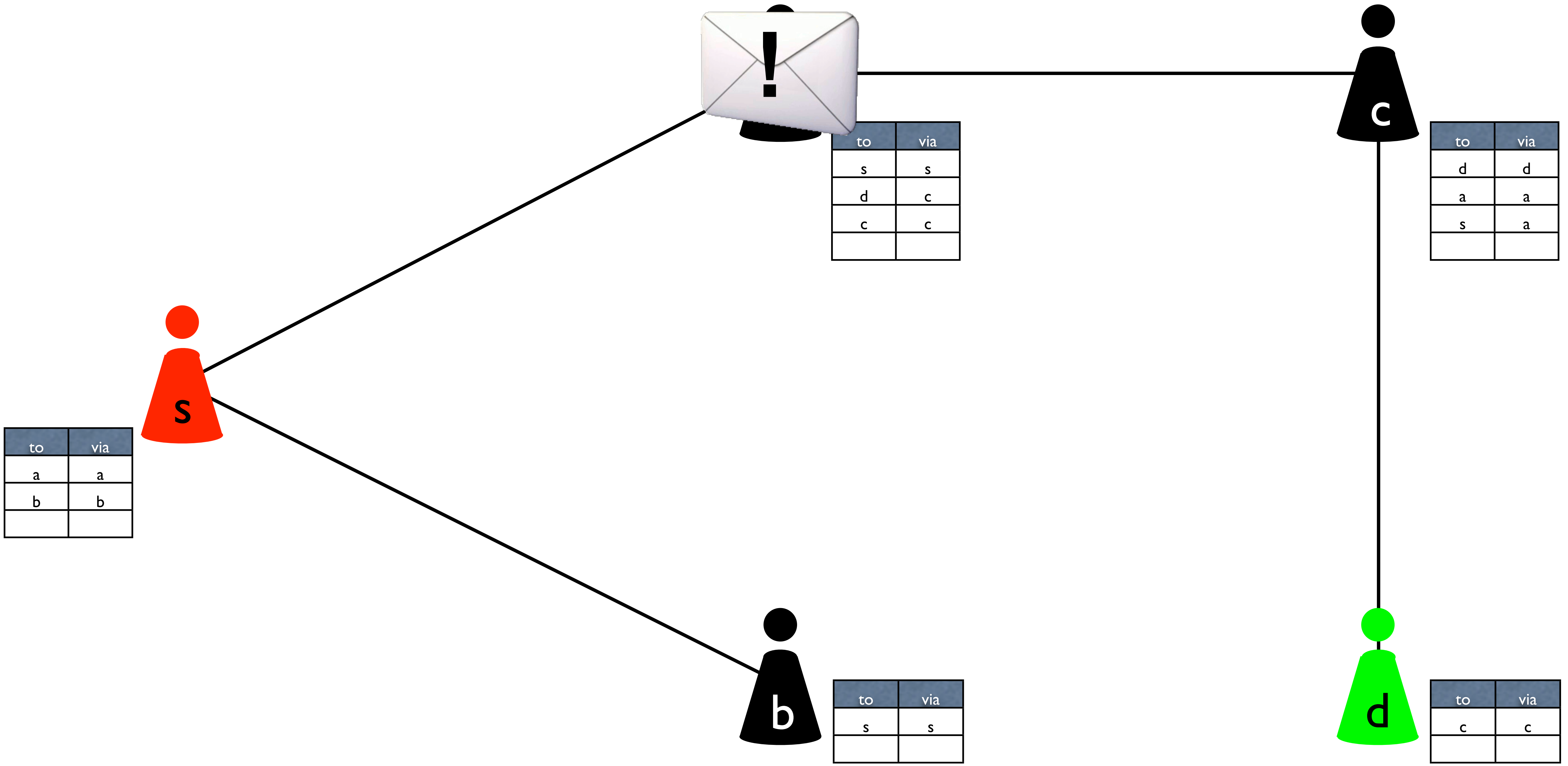
to	via
d	d
a	a
s	a

to	via
a	a
b	b

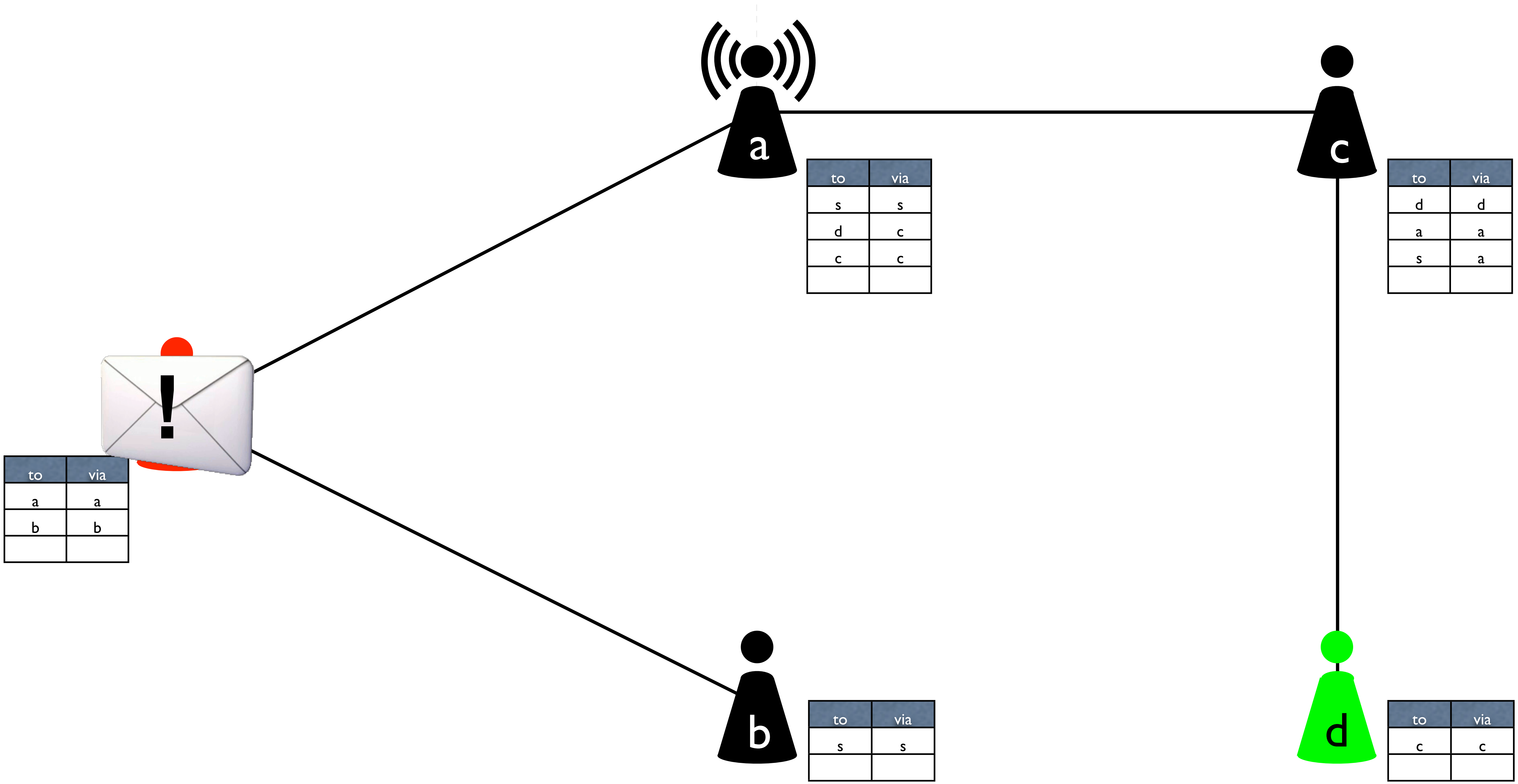
to	via
s	s

to	via
c	c

AODV - An Example

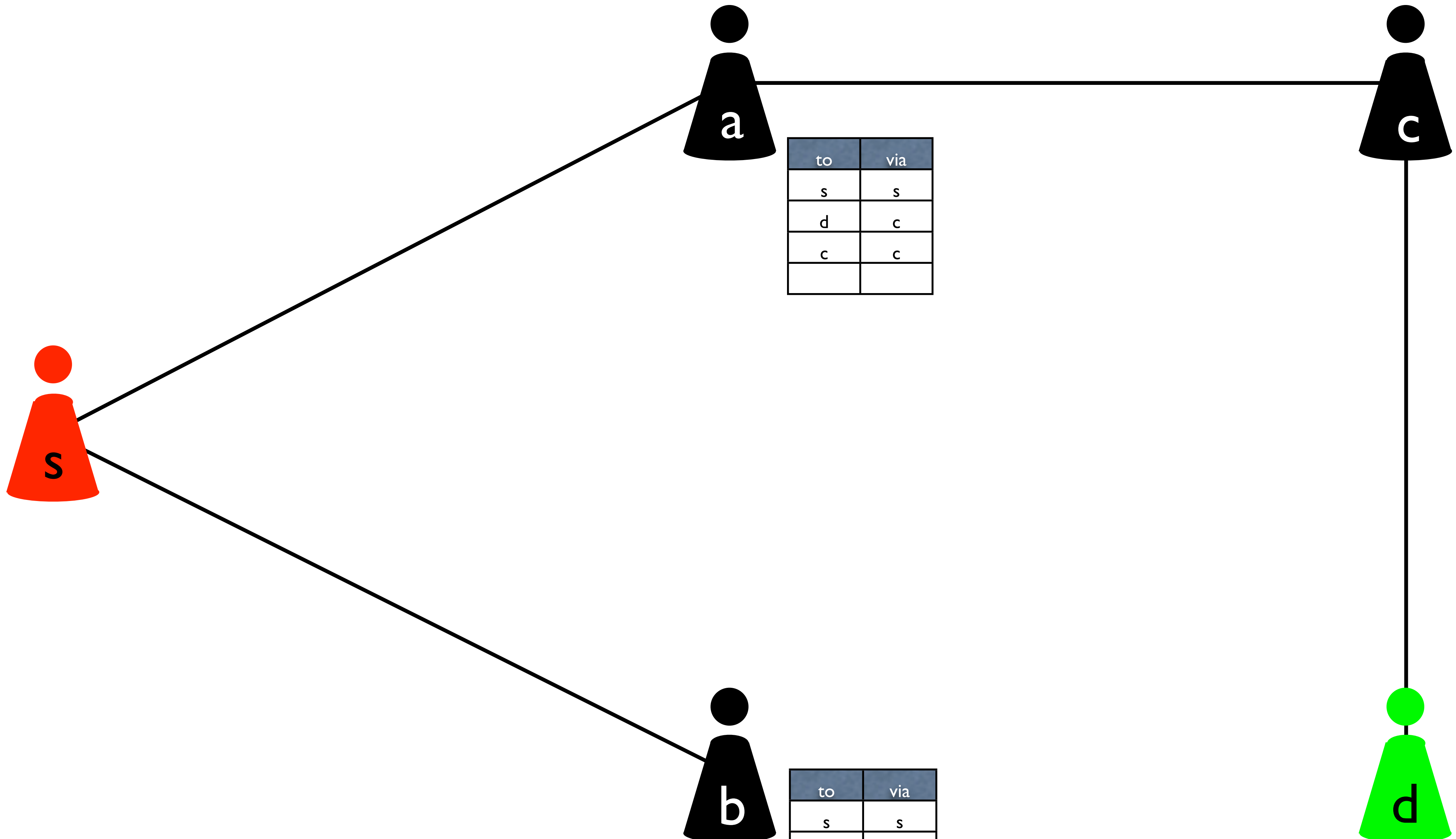


AODV – An Example



a forwards route reply

AODV – An Example



to	via
s	s
d	c
c	c

to	via
d	d
a	a
s	a

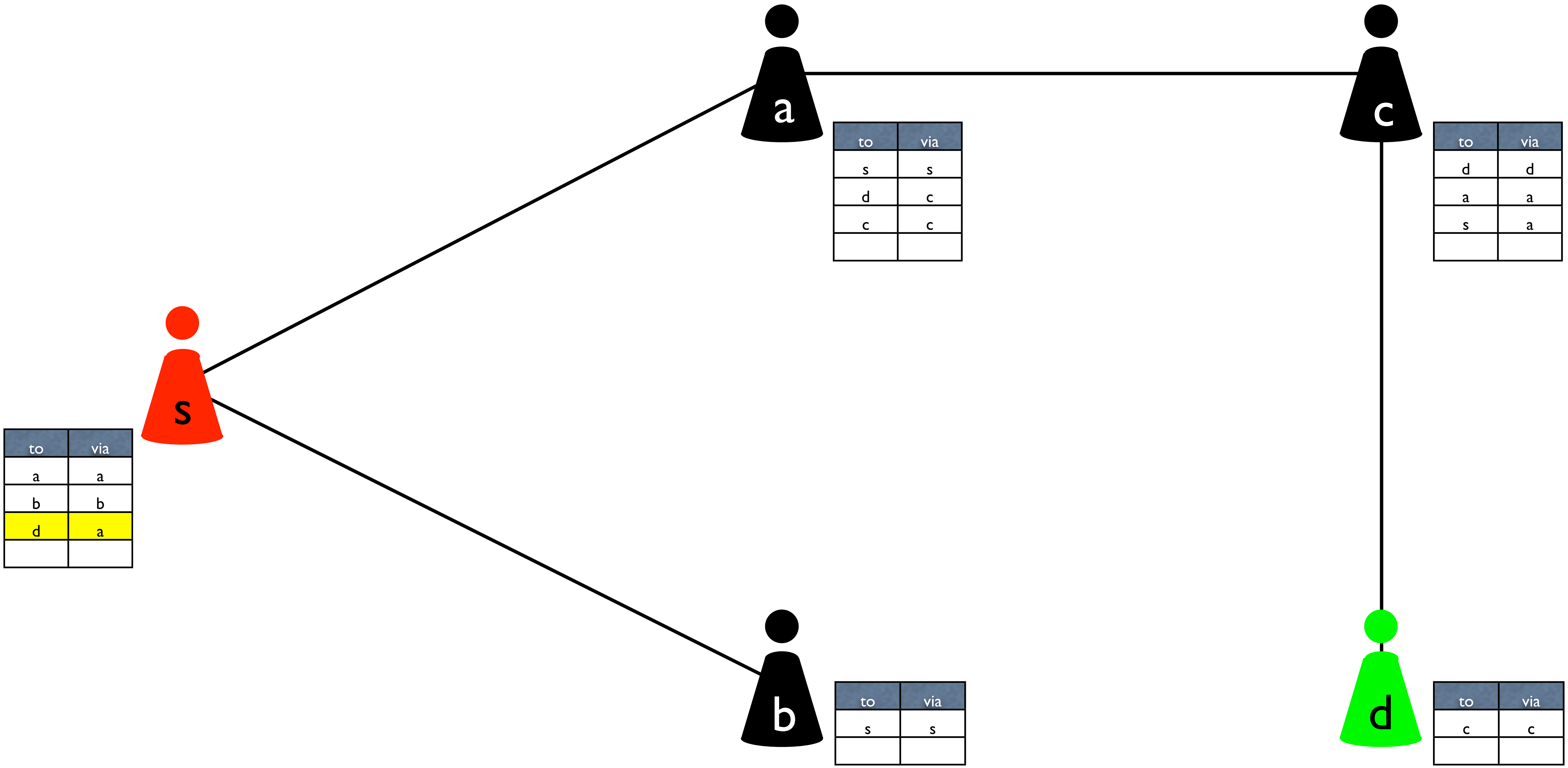
to	via
a	a
b	b
d	a

to	via
s	s

to	via
c	c

a forwards route reply

AODV – An Example



AODV – An Example

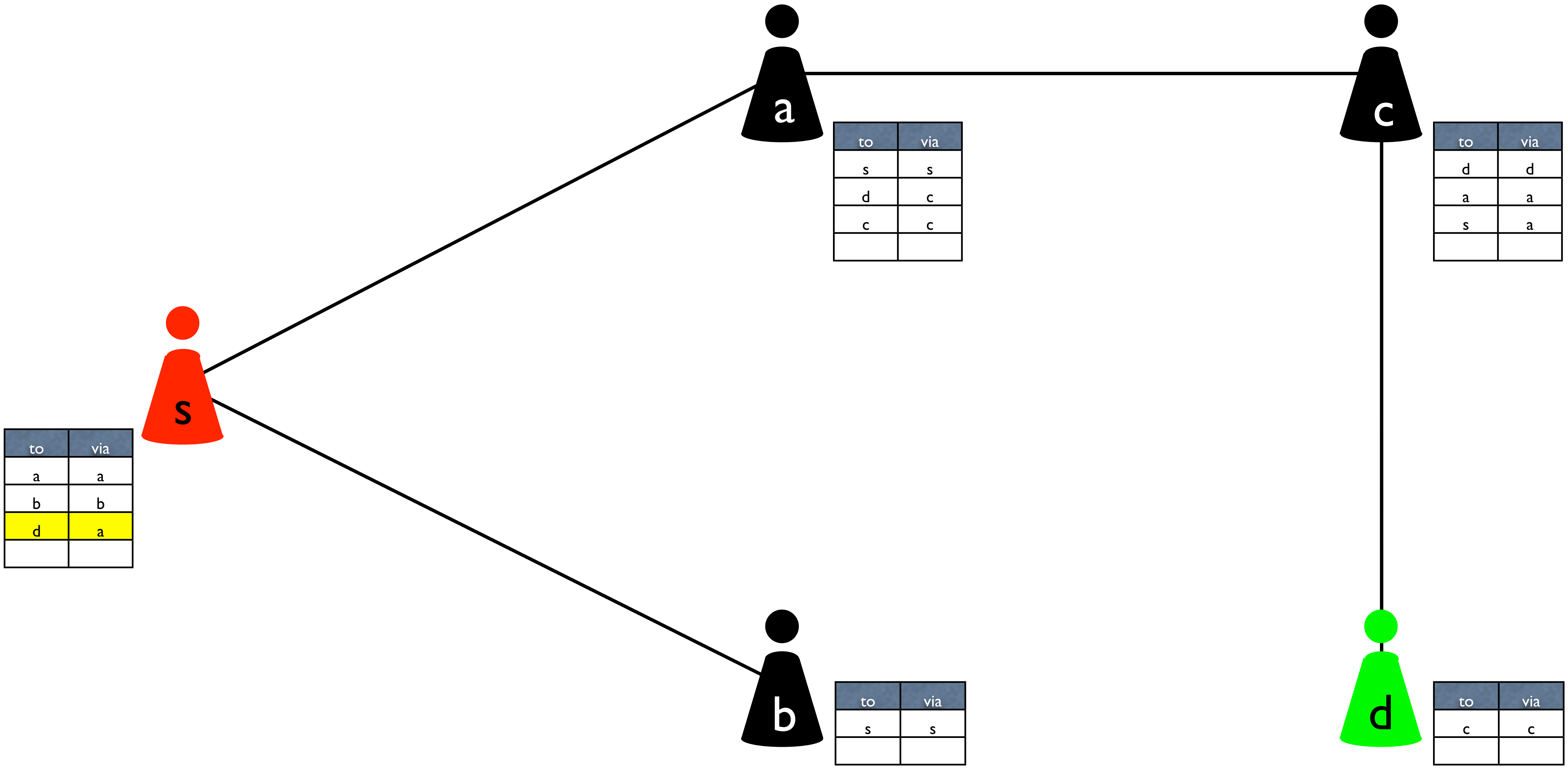


Table at node s:

to	via
a	a
b	b
d	a

Table at node a:

to	via
s	s
d	c
c	c

Table at node c:

to	via
d	d
a	a
s	a

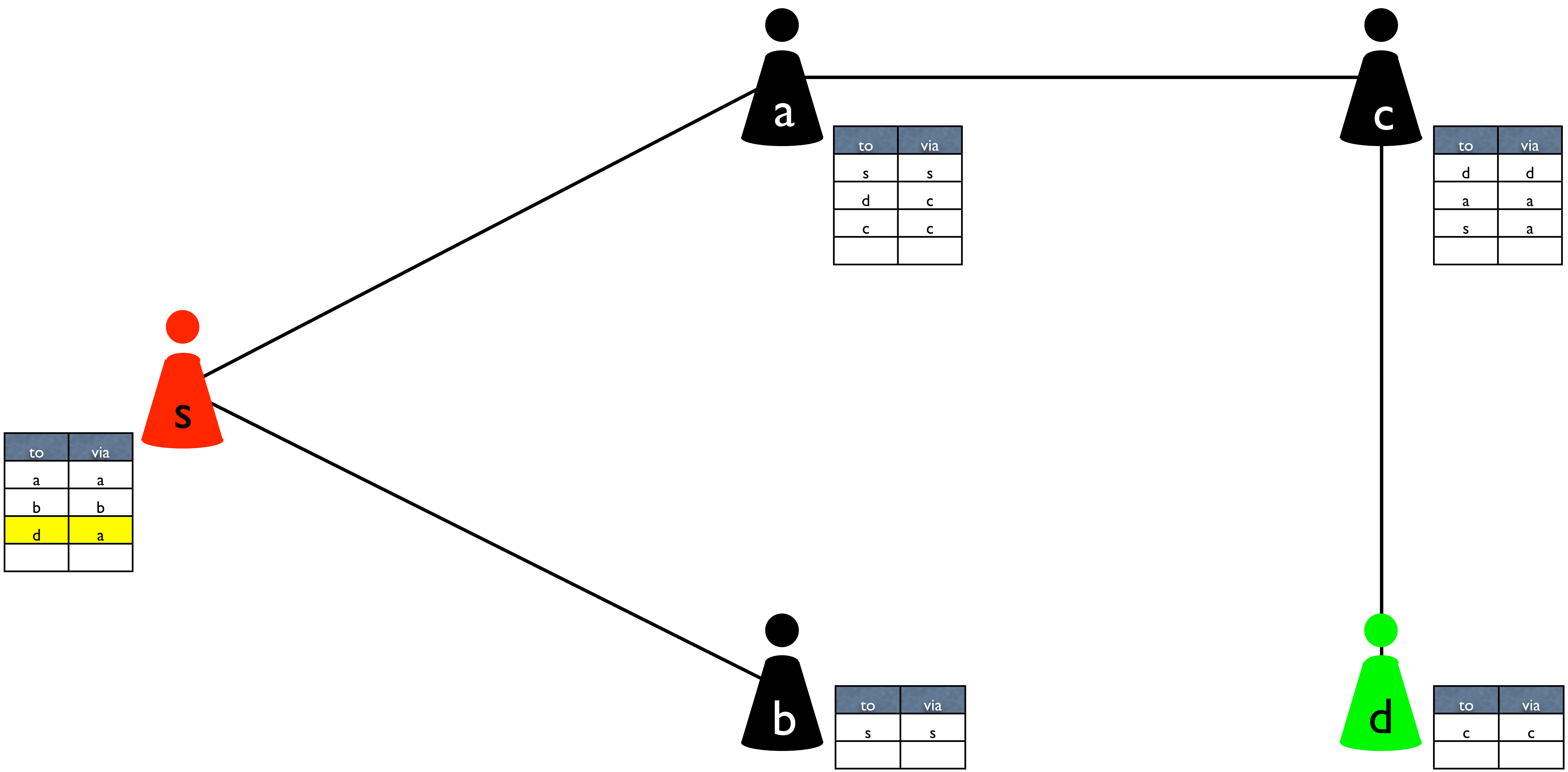
Table at node b:

to	via
s	s

Table at node d:

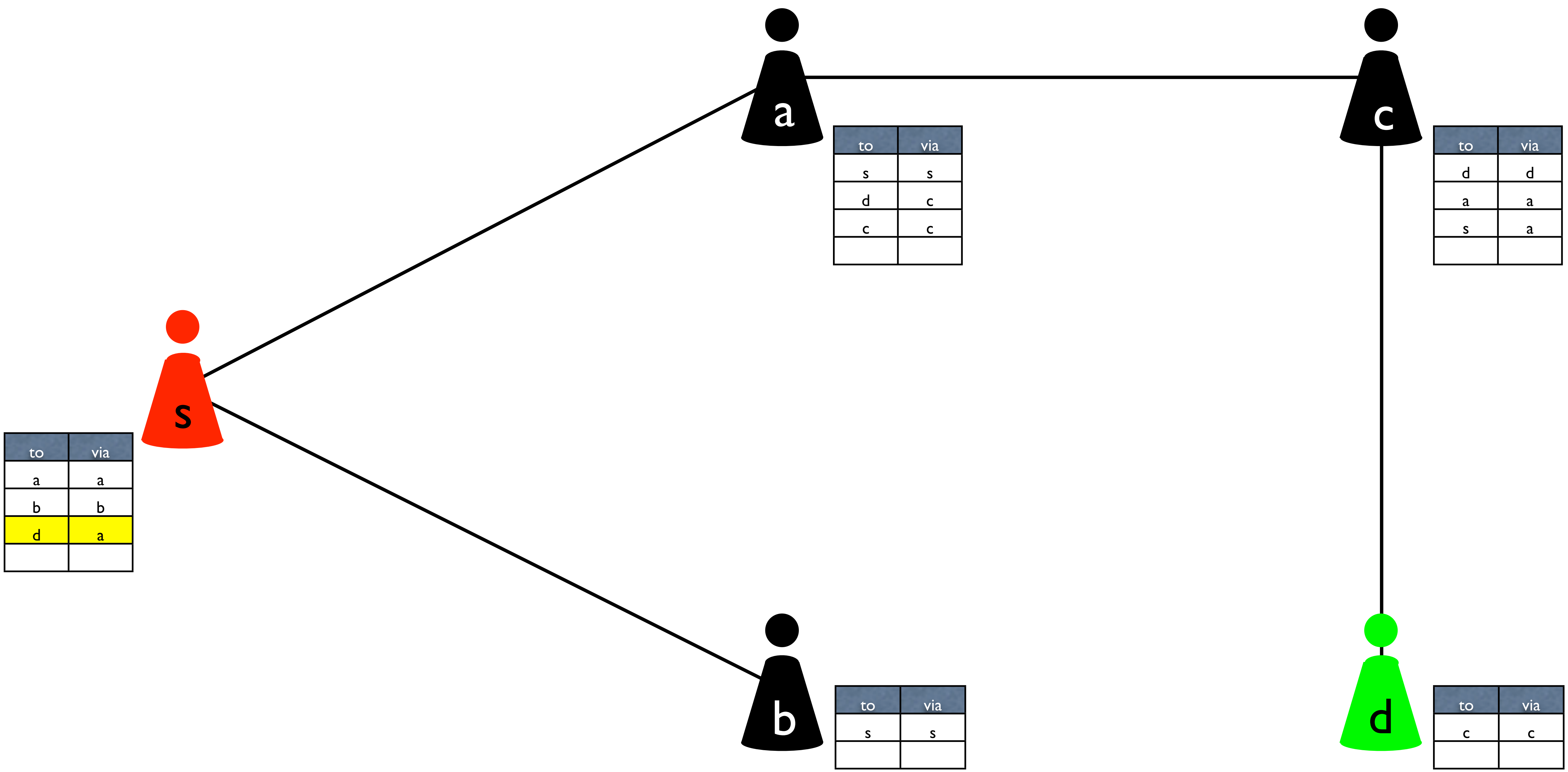
to	via
c	c

AODV – An Example



s has found a route to d

AODV – An Example



s has found a route to d

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

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

Routing Algebra – Elements, Operators

- Routing table entries (no sequence number so far)

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

- Special symbols: $(\_, 0)$, $(\_, \infty)$

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

- both structures form monoid
- composition distributes over addition

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

Routing Algebra – Elements, Operators

- Matrices over routing table entries

$$
\begin{array}{c c}
 & \begin{array}{ccccc} A & B & C & D & \cdots \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ \vdots \end{array} &
\left(\begin{array}{ccccc}
(\_, 0) & (B, 1) & (B, 2) & (\_, \infty) & \\
(A, 1) & (\_, 0) & (C, 1) & (\_, \infty) & \cdots \\
(\_, \infty) & (B, 1) & (\_, 0) & (\_, \infty) & \\
(\_, \infty) & (\_, \infty) & (\_, \infty) & (\_, 0) & \\
 & \vdots & & & \ddots
\end{array}\right)
\end{array}
$$

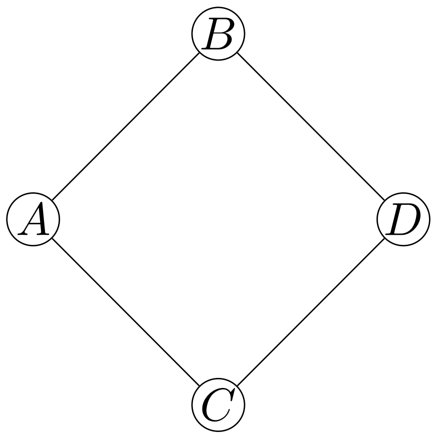
routing table of A

"routes" to B

- standard matrix operations

Example

- A route request is broadcast



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

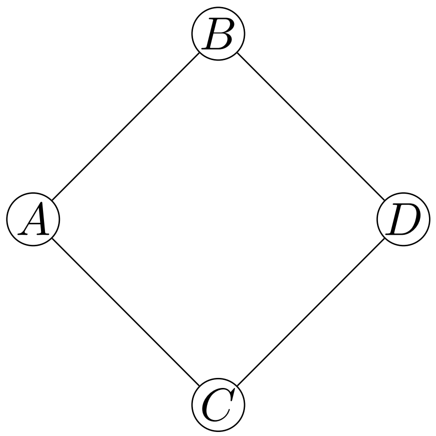
$$= \underbrace{\begin{pmatrix} (\_,0) & (B,1) & (\_,\infty) & (\_,\infty) \\ (\mathbf{A},\mathbf{1}) & (\_,0) & (\_,\infty) & (\_,\infty) \\ (A,1) & (\_,\infty) & (\_,0) & (D,1) \\ (C,2) & (\_,\infty) & (C,1) & (\_,0) \end{pmatrix}}_{\text{updated routing table}}$$

Further Abstraction

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

Example

- A route request is broadcast



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

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

Sent Messages

- sending messages

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

- by distributivity

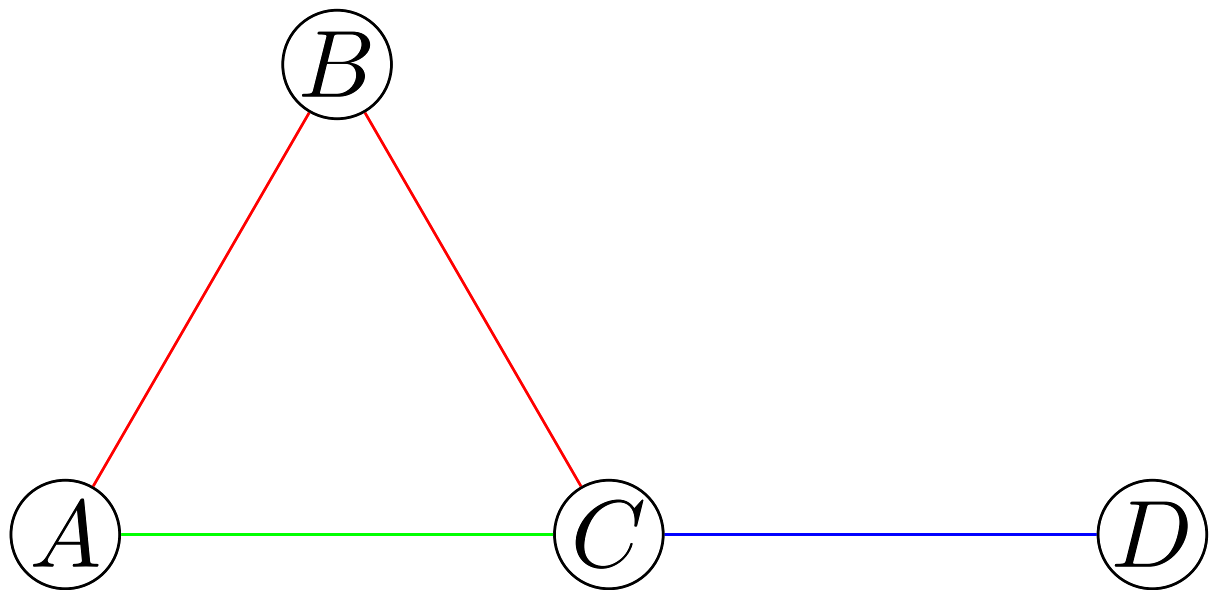
$$a \;+\; p \cdot b \cdot q \;+\; p \cdot b \cdot q \cdot c$$

  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?

Lost and Found

- Adding sequence numbers



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

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

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)

Miscellaneous / Future Work

- Ad hoc prototype in Haskell
- Theorems at algebraic level proven with Prover9

- Include sequence numbers (partial Kleene algebra)
- Can everything be lifted to the algebraic level?
- Important properties loop freedom, route correctness
- Improvement/refinement
- Probably domain-theoretic (model) knowledge needed
- Use Isabelle/HOL to switch between model and algebra
- ...

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

NICTA

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

Different Network Layers