NICTA

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

Australian Government

Department of Broadband, Communications and the Digital Economy

Australian Research Council

NICTA Funding and Supporting Members and Partners

Australian National University

UNSW THE UNIVERSITY OF NEW SOUTH WALES

NSW GOVERNMENT Trade & Investment

State Government Victoria

THE UNIVERSITY OF MELBOURNE

THE UNIVERSITY OF SYDNEY

Queensland Government

Griffith UNIVERSITY

QUT Queensland University of Technology

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Ad hoc Routing in Mesh Networks using Algebra

Peter Höfner

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

NICTA Funding and Supporting Members and Partners

NICTA Partners

Some Background (Recap)

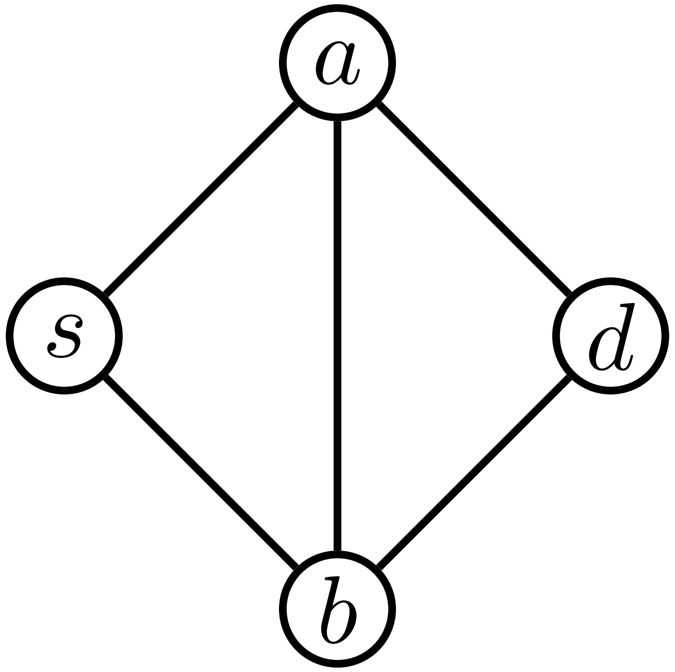
- Routing protocols
  - find a route (in a dynamic topology)
  - properties
    - route correctness (if a route is found, the route is actually present)
    - route discovery (if a route exist, the route is found)
    - loop freedom (packets do not circulate)
    - packets are delivered (eventually)

- Routing tables
  - collect (known) data
    - IP address, local connections, next hops ...

© NICTA 2013

Some Background (Recap)

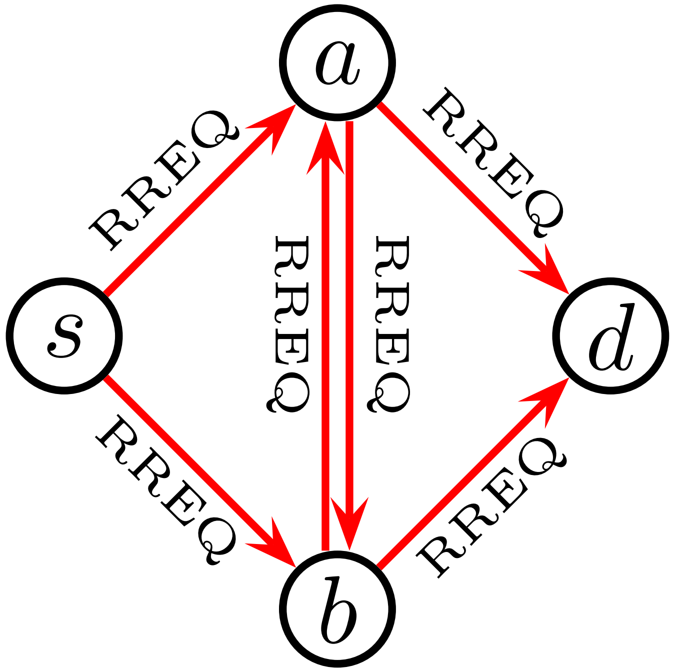
- Goal: Study routing algorithms algebraically
  – inspired by the standard, popular routing protocol AODV
- Ad hoc on-demand distance vector protocol (AODV)
  – main Mechanism
    - if route is needed
      BROADCAST RREQ
    - if node has information about a destination
      UNICAST RREP
    - if unicast fails or link break is detected
      GROUPCAST RERR
  – routing table
    - destination address
    - next hop (not the entire path)
    - length of the route
    - parameter about freshness
      (sequence numbers)



© NICTA 2013

Some Background (Recap)

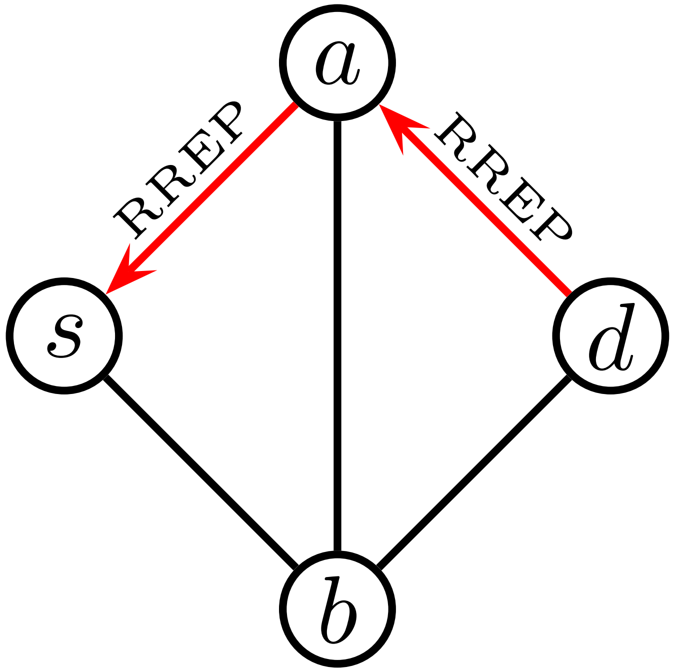
- Goal: Study routing algorithms algebraically
  - inspired by the standard, popular routing protocol AODV
- Ad hoc on-demand distance vector protocol (AODV)
  - main Mechanism
    - if route is needed
      BROADCAST RREQ
    - if node has information about a destination
      UNICAST RREP
    - if unicast fails or link break is detected
      GROUPCAST RERR
  - routing table
    - destination address
    - next hop (not the entire path)
    - length of the route
    - parameter about freshness
      (sequence numbers)



© NICTA 2013

Some Background (Recap)

- Goal: Study routing algorithms algebraically
  - inspired by the standard, popular routing protocol AODV
- Ad hoc on-demand distance vector protocol (AODV)
  - 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
      GROUPCAST RERR
  - routing table
    - destination address
    - next hop (not the entire path)
    - length of the route
    - parameter about freshness (sequence numbers)



© NICTA 2013

Towards an Algebra

- Algebra
  - offer operations for main primitives (broadcast, unicast, ...)
  - model properties such as loop freedom algebraically
- Operators
  - choice
    - if a node has the choice between two routes, it has to choose one
  - composition
    - if two routes are known they can be combined

© NICTA 2013

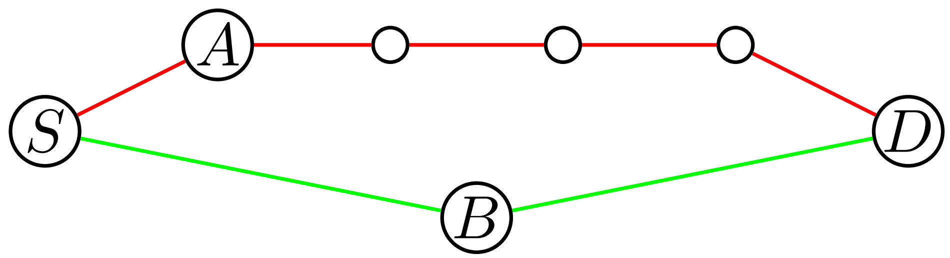
Algebraic Operations (minimal requirements)

- Routing table entries (no sequence number so far)

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

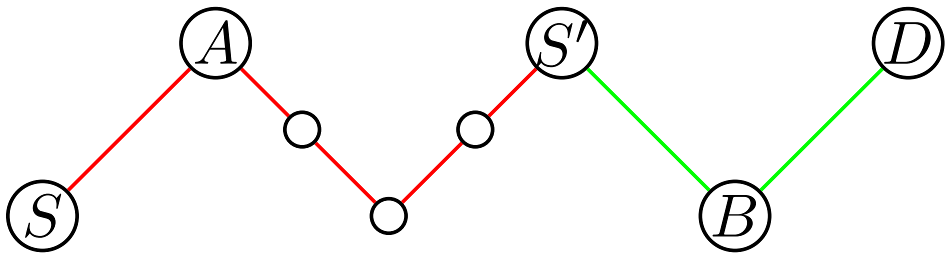
- Choice (lexicographical order):

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



- Multiplication (destination and source must coincide)

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



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

© NICTA 2013

Underlying Structure

- Both (+) and (·) structures form monoids
- Multiplication distributes over addition
- Lifts to matrices
- Use semirings and Kleene algebras to study routing protocols?
- inspired by Backhouse, Carré, Griffin, Sobrinho

© NICTA 2013

Routing Algebra – Elements, Operators

- Matrices over routing table entries

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

routing table of A

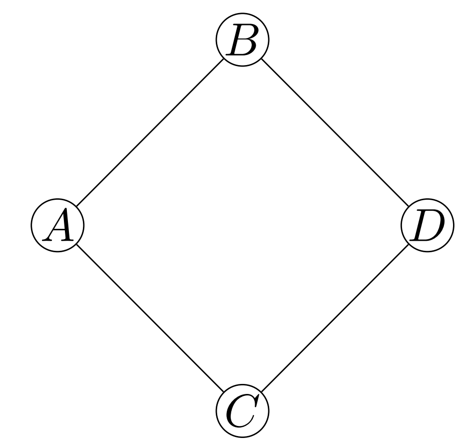
“routes” to B

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

© NICTA 2013

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 2013

Sending of Messages

- Sending messages

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

with

- a known knowledge (snapshot)
- p, q sender and receiver
- b topology
- $p \cdot b \cdot q$ restricted topology
- $1 + c$ possible updates/information sent

© NICTA 2013

Calculating with Messages

Definition: messages can be defined as

$$\mathtt{msg}(a, b, c) = a + b \cdot (1 + c) \qquad (1 \leq b)$$

Properties:

- If the c and c' is fixed (does not change when sending a message), the order of sending does not matter, i.e.,

$$\mathtt{msg}(\mathtt{msg}(a, b, c), b', c') = \mathtt{msg}(\mathtt{msg}(a, b', c'), b, c) \ .$$

- If different messages are sent via a shared topology b, the messages can be sent in parallel, i.e.,

$$\mathtt{msg}(\mathtt{msg}(a, b, c), b, c') = \mathtt{msg}(a, b, c + c') \ .$$

- If the same message is sent via different connections, connections can be joined, i.e.,

$$\mathtt{msg}(\mathtt{msg}(a, b, c), b', c) = \mathtt{msg}(a, b + b', c) \ .$$

These properties as well as others can be automatically proven (e.g. by Prover9)

© NICTA 2013

Distributing a message (Flooding the Network)

- Forwarding message

$$\begin{aligned} & \mathtt{msg}(\mathtt{msg}(a,b,c),b',b\cdot c) \\ = \quad & a+b+b\cdot c+b'+b'\cdot b\cdot c \\ \leq \quad & a+b'+b'\cdot b+b'\cdot b\cdot c \\ = \quad & a+b'(1+b+b\cdot c) \\ = \quad & \mathtt{msg}(a,b',b+b\cdot c) \end{aligned}$$

- knowledge after forwarding a message once can be approximated by sending a single message via b′ with knowledge of the first topology b and the learnt component $b \cdot c$
- in case the topology does not change

$$\mathtt{msg}(\mathtt{msg}(a,b,c),b,b\cdot c) = \mathtt{msg}(a,b,b\cdot c)\ .$$

© NICTA 2013

Distributing a message (Flooding the Network)

- Broadcasting a message

$$\begin{aligned}\mathtt{msg}(a, b, b^* \cdot c) &= a + b \cdot (1 + b^* \cdot c) \\ &= a + b + b \cdot c + b \cdot b \cdot c + b \cdot b \cdot b \cdot c + \ldots\ .\end{aligned}$$

where * is Kleene star

- Single source

$$\mathtt{msg}(a, b \cdot |b^*\rangle p, b^* \cdot p) = a + b \cdot |b^*\rangle p + b^* \cdot p$$

with sender p $(p \leq 1, \mathrm{test})$
and receivers $|b^i\rangle p$

$$|a\rangle p \leq q \Leftrightarrow \neg q \cdot a \cdot p \leq 0 \ \text{ and } \ |a \cdot b\rangle p = |a\rangle(|b\rangle p)$$

© NICTA 2013

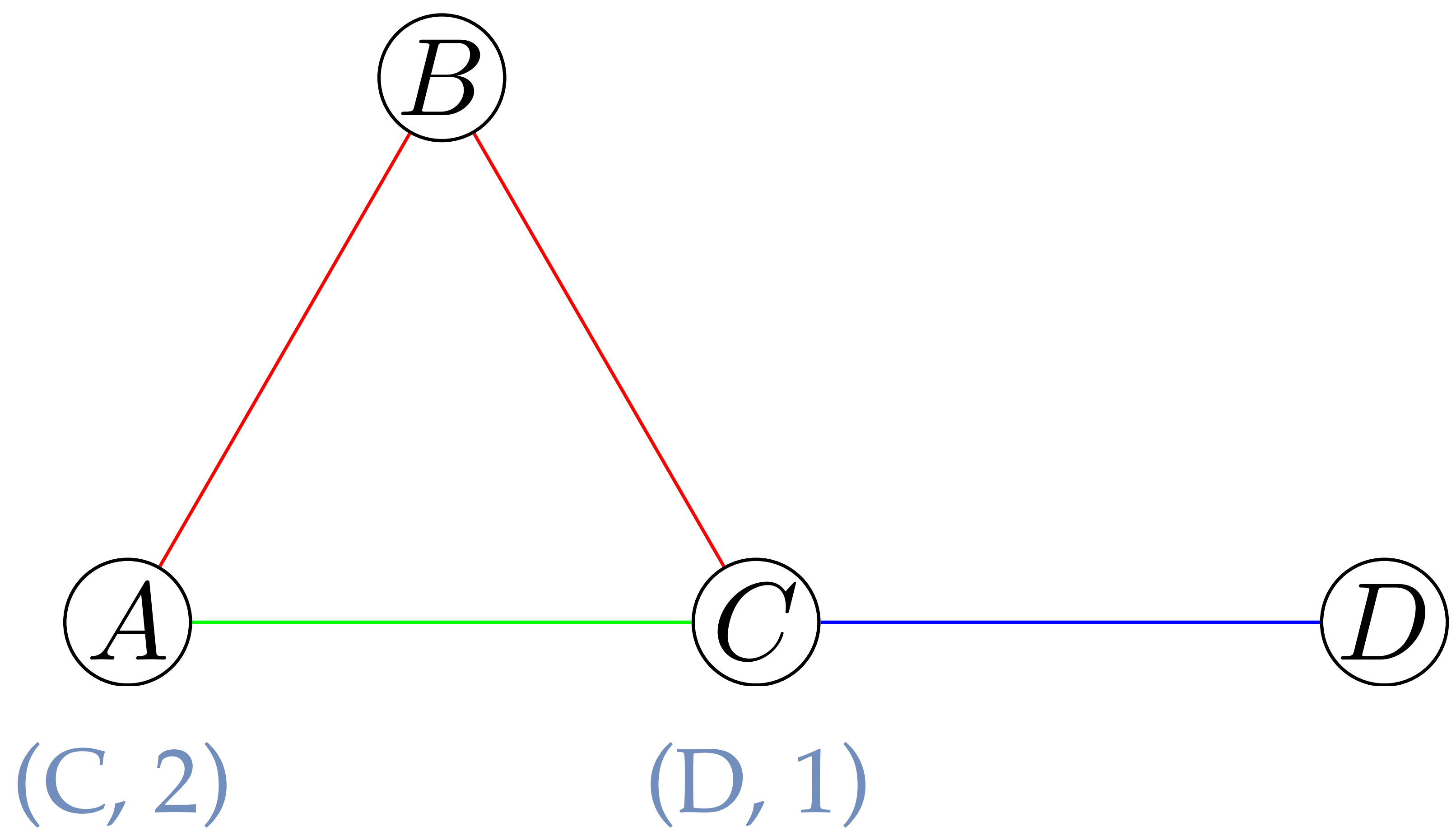
Unicast and Broadcast

- By varying the topology one can model broadcast, multicast and unicast.
- Modal operators can be used to characterise stopping criteria (of AODV)
  (use $b \cdot |a]\neg q$ as topology, where $|a]p = \neg|a\rangle\neg p$)

© NICTA 2013

Loop Freedom and Wrong Routing Protocols

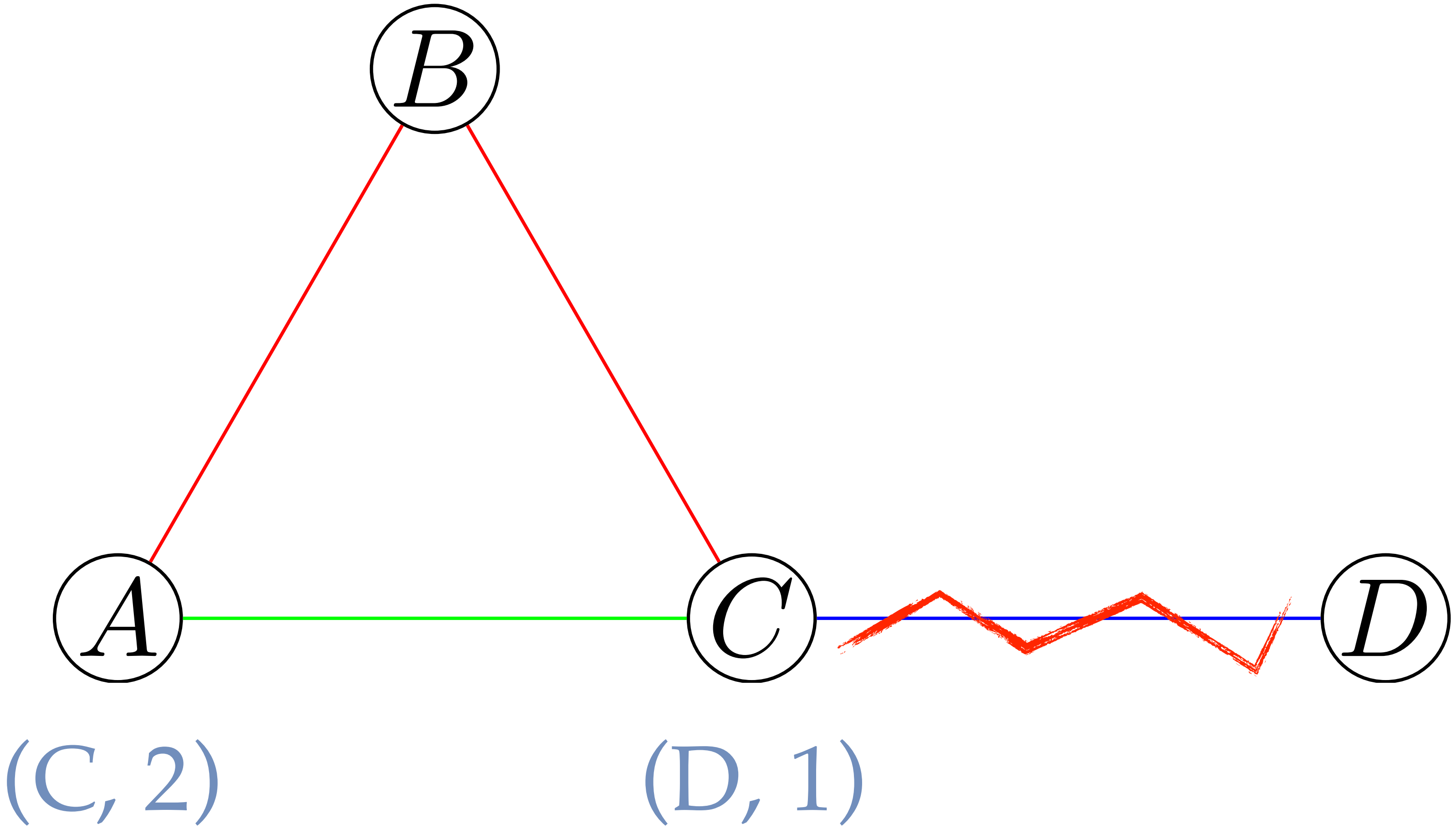
- Routing protocols (on top of dynamic topologies) must avoid routing loops
  - C and A have established routes to D



© NICTA 2013

Loop Freedom and Wrong Routing Protocols

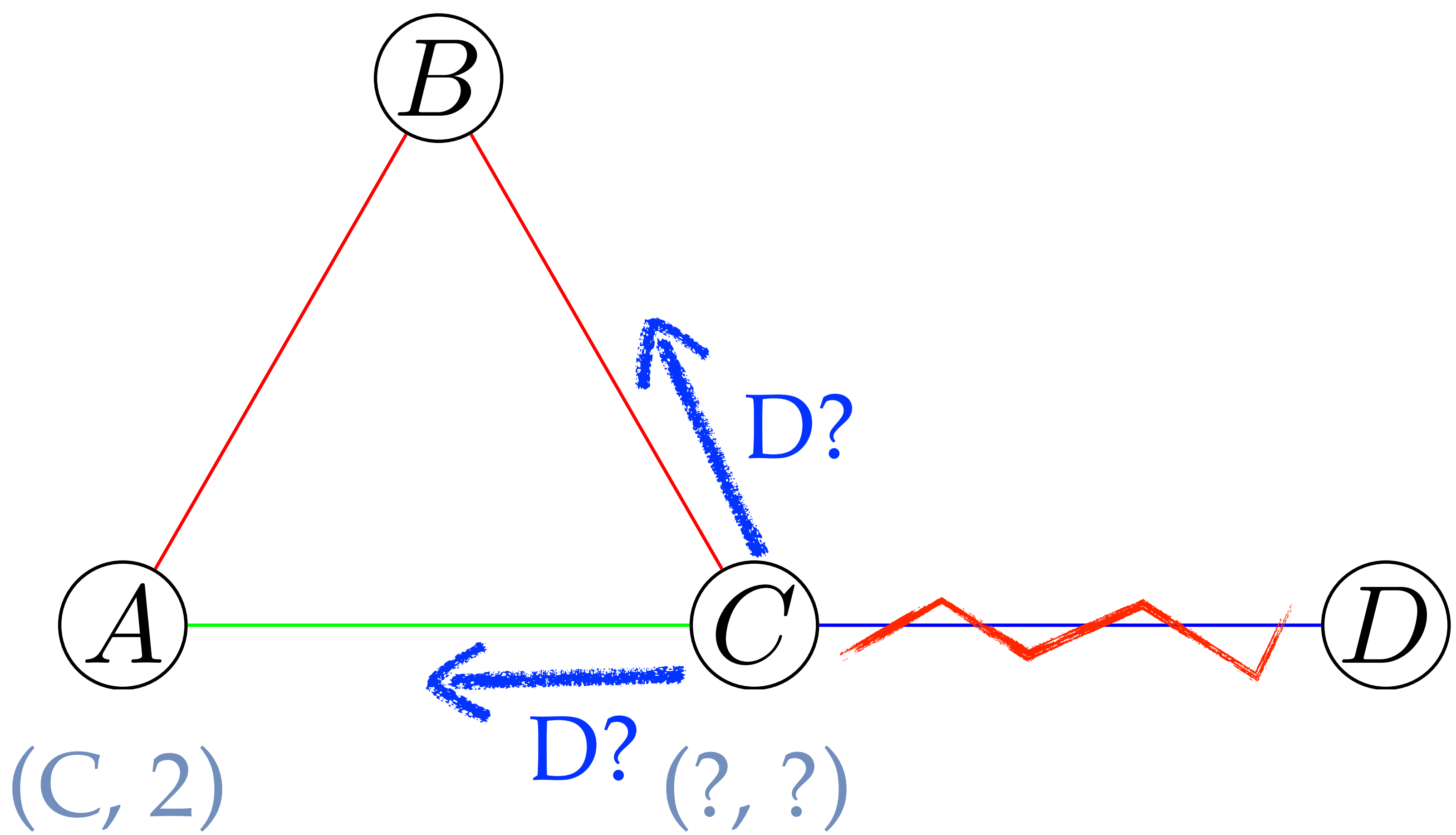
- Routing protocols (on top of dynamic topologies) must avoid routing loops
  - C and A have established routes to D



© NICTA 2013

Loop Freedom and Wrong Routing Protocols

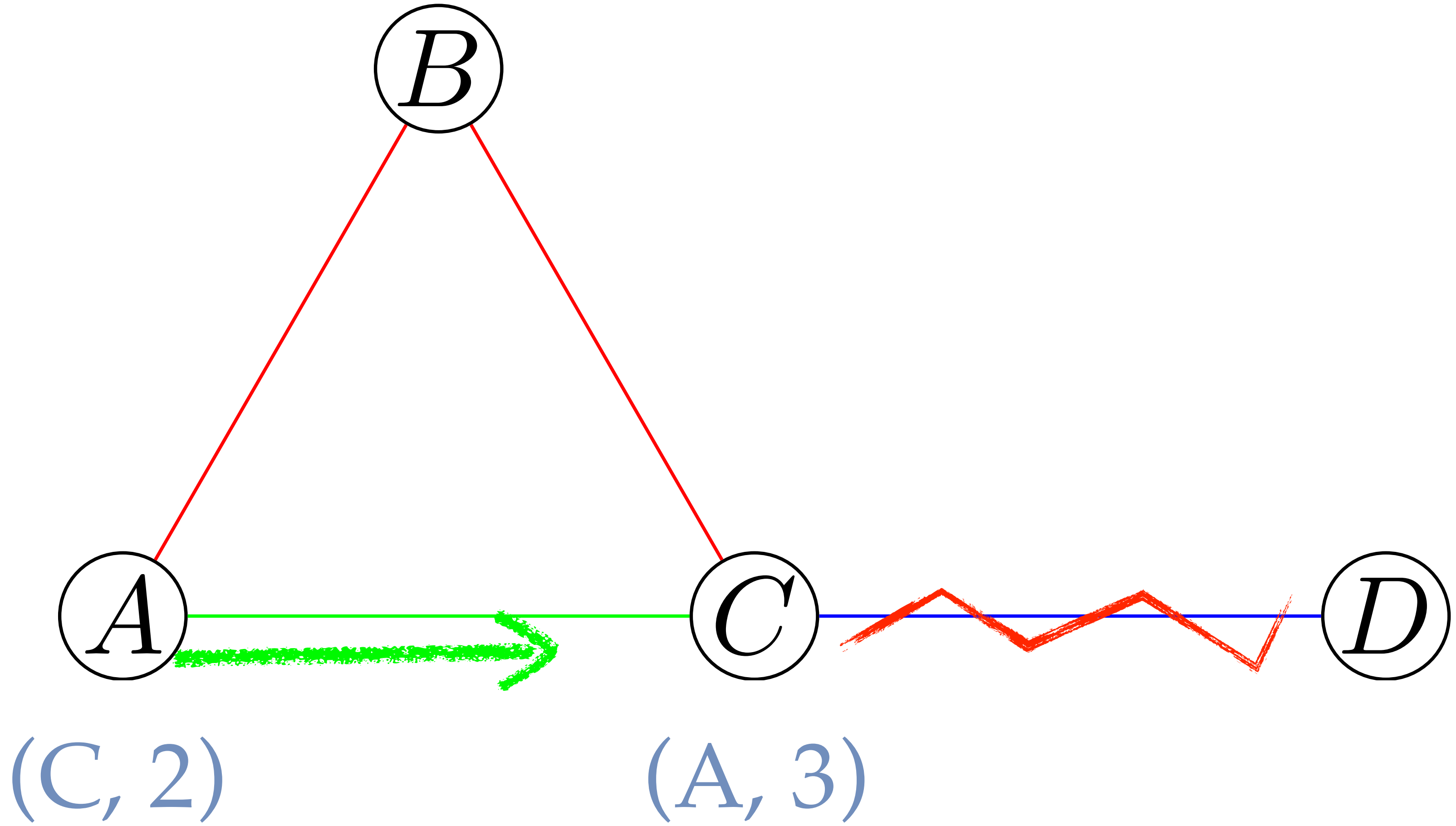
- Routing protocols (on top of dynamic topologies) must avoid routing loops
  - C send request to find route to D



© NICTA 2013

Loop Freedom and Wrong Routing Protocols

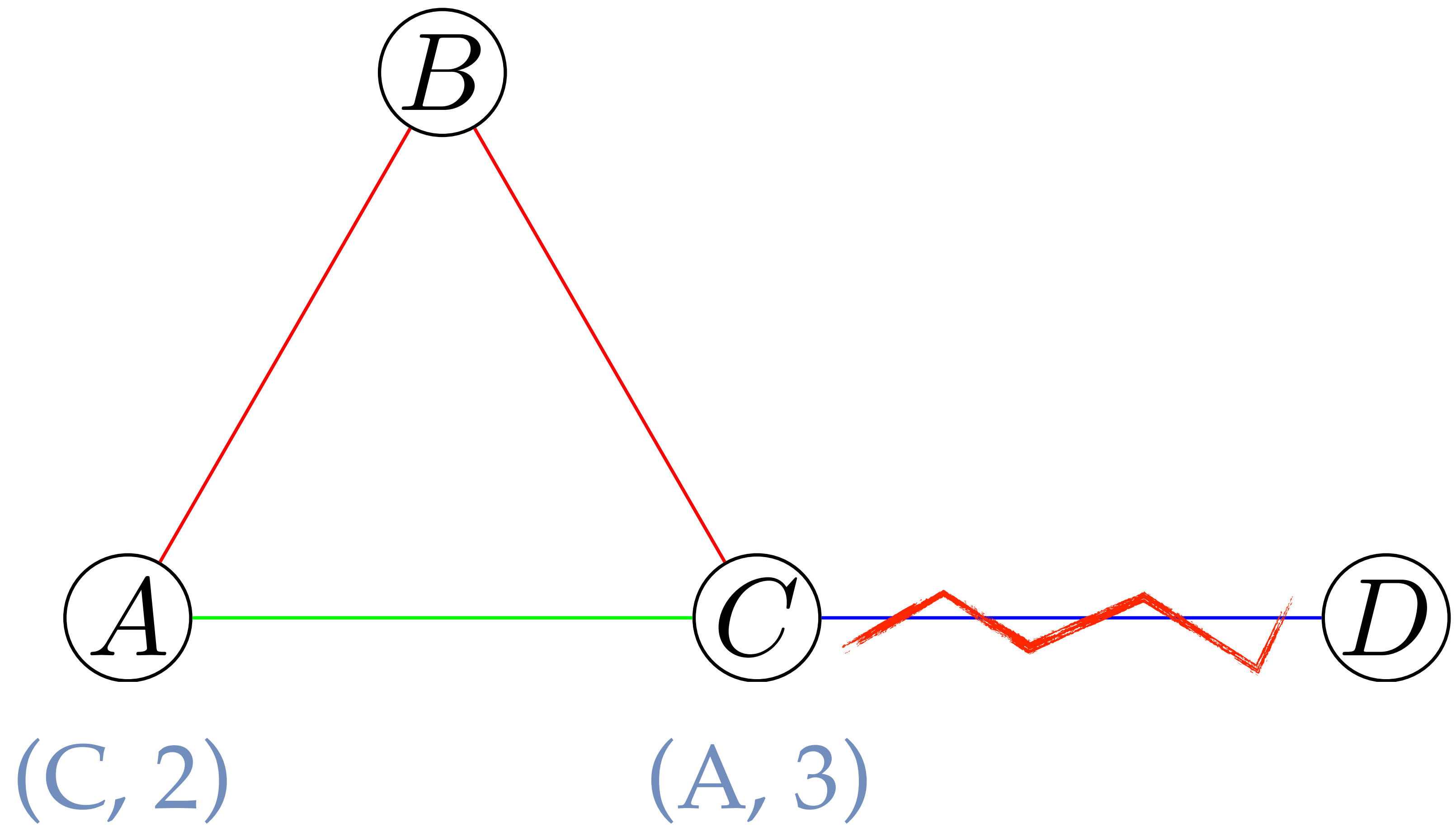
- Routing protocols (on top of dynamic topologies) must avoid routing loops
  - A answers with a route reply



© NICTA 2013

Loop Freedom and Wrong Routing Protocols

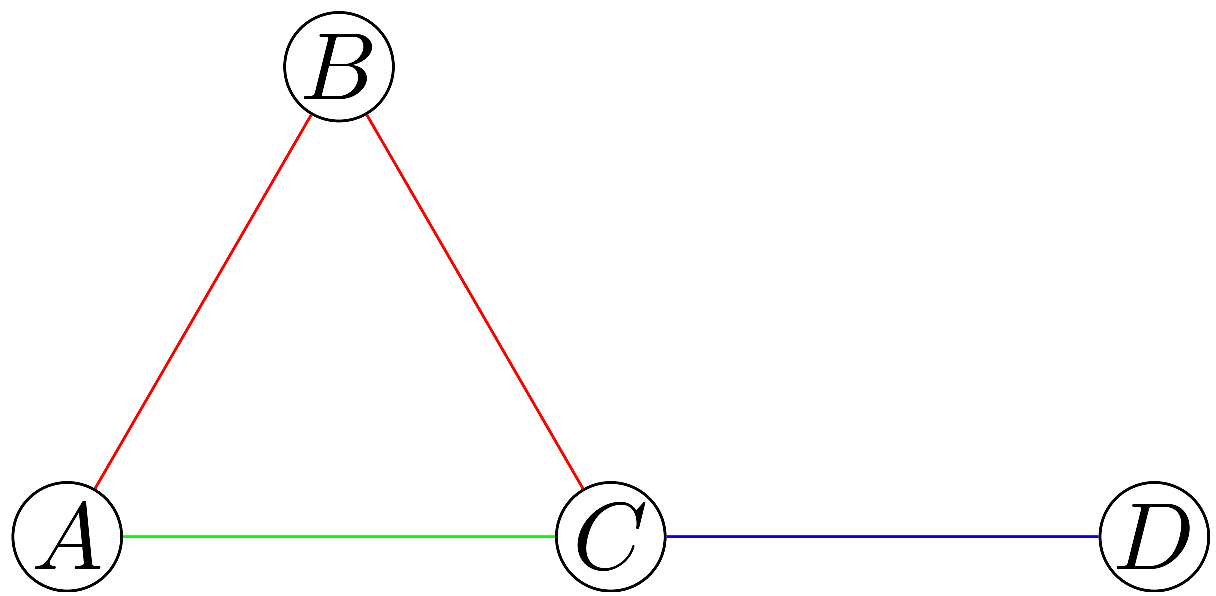
- Routing protocols (on top of dynamic topologies) must avoid routing loops
  - A routing loop has been established



© NICTA 2013

Guaranteeing Loop Freedom

- add an attribute “freshness”
  - routing information records the “destination sequence number”, i.e. the sequence number reported by messages “coming from” that destination: (nhop, hops, dsn)



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

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

© NICTA 2013

Towards a Solution

- The problem is that in our algebraic setting the topology would carry sequence numbers.
  - intuitively this does not make sense

- Idea: distinguish between routing tables and topologies
  - routing table
    - knowledge of nodes
    - information sent via the topology
  - topology
    - information about (current) connectivity

© NICTA 2013

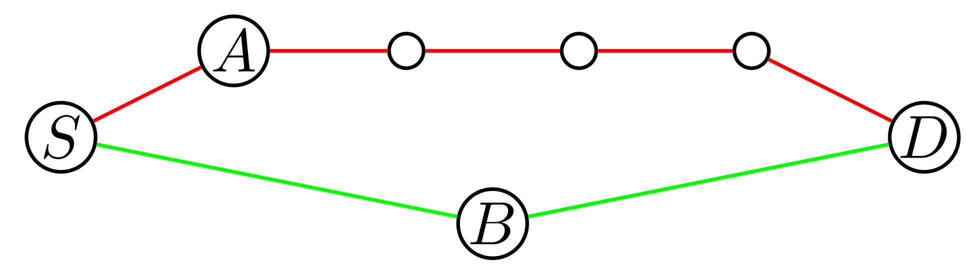
Algebraic Operations (incl. sequence numbers)

- Topologies (no sequence number)

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

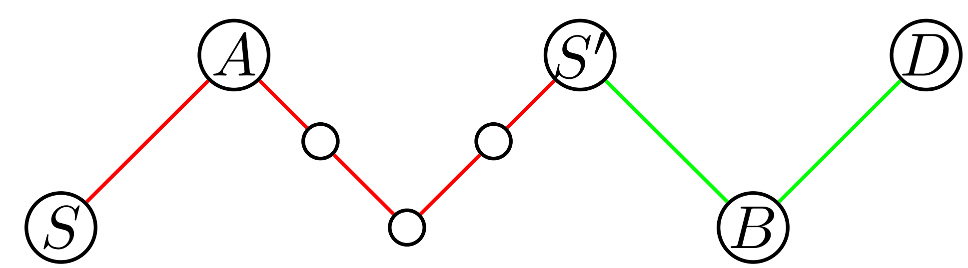
- Choice (lexicographical order):

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



- Multiplication (destination and source must coincide)

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



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

© NICTA 2013

Algebraic Operations (incl. sequence numbers)

- Routing table entries

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

- Choice (on topologies):

$$(A,5,10)\sqcup(B,2,3)=(A,5,10)$$

- Multiplication does not exist
- Special symbol: $(\_,\infty,\infty)$

© NICTA 2013

Algebraic Operations (incl. sequence numbers)

- Mapping topologies to routing tables (updating routing tables)

$$(A, 5) : (B, 2, 5) = (A, 7, 5)$$

© NICTA 2013

Underlying Structure

- (+), (·), (⊔) structures form monoids
- multiplication distributes over addition
- scalar product (:) satisfies

$$\begin{array}{lrcl}
\textit{unit} & 1 : r & = & r \ , \\
\textit{distributivity} & (t + t') : r & = & (t : r) \sqcup (t' : r) \\
\textit{distributivity} & t : (r \sqcup r') & = & (t : r) \sqcup (t : r') \\
\textit{associativity} & (t \cdot t') : r & = & t : (t' : r)
\end{array}$$

- together this structure forms a Kleene Module
  – (à la Leiß)
- lift to matrices

© NICTA 2013

From Kleene Algebra to Kleene Modules

- all theory presented can be transferred to Kleene modules
  - e.g. sending messages

$$\mathtt{msg}(a, b, c) = a \sqcup b : c$$

$$\begin{aligned}
\mathtt{msg}(a, b, b^* : c) &= a \sqcup b \cdot (1 + b^* : c) \\
\mathtt{msg}(a, b \cdot |b^*\rangle p, b^* : p') &= a \sqcup b \cdot |b^*\rangle p : \mathsf{e} \sqcup b^* : p' \\
&= a \sqcup b \cdot \ulcorner(b^*) : p' \sqcup b^* : p'
\end{aligned}$$

© NICTA 2013

On-Going Work

- Modelling AODV
  - require additional operations for
    - incrementing sequence numbers
    - invalidating routes ...
- Unicast
  - so far unicast was modelled by a given topology
  - can this topology determined automatically?
  - maybe via fixpoints

© NICTA 2013

On-Going Work

- Properties of Routing Protocol
  - route correctness (by construction)
  - route discovery

$$s \cdot P \cdot d \neq 0$$

  - route optimality (for static topology b)

$$s \cdot P \cdot d = s \cdot b^* \cdot d$$

  - loop freedom
    - details still open
    - use “inverse” of scalar product to forget sequence numbers
    - then compare with identity

© NICTA 2013

Future Work

- Formalise main aspects of AODV
  - AODV works on 4-tuples rather than triples (fits well in the theory of modules)
- Try to derive a “correct” protocol from algebraic specification
- Maybe introduce time in the model
  - (seminal work by Hoare, von Karger, Hayes)

© NICTA 2013

NICTA

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

Advertisement

- Title: **Ad hoc Routing in Mesh Networks using Algebra**
  - Author: Peter Höfner
  - Affiliation: NICTA (National ICT Australia) and UNSW
  - Research Interest: Modelling and Verification of (Software Systems) using formal methods such as algebraic structures. At the moment focus on routing and communication protocols
  - Abstract: At the meeting in Rome I gave an overview of formal modelling and analysis of routing protocols for wireless mesh networks (WMNs). Afterwards I was asked to present details about the methods used. This talk presents some more details about the algebra used to model main aspect of routing protocols.
  - time: 30-40 min

© NICTA 2013