MessageWay

High Performance Intra- and Inter-Computer Communication System

IETF, Stockholm, Jul-18-1995

Danny Cohen

Myricom, Inc.

325 N. Santa Anita Ave, Arcadia, CA 91006

818-821-5555 <Cohen@Myri.com> Fx:818-821-5316

http://www.myri.com

MessageWay

Mailing-List: <MsgWay@myri.com>
and <MsgWay-request@myri.com>

POC: Danny Cohen
Myricom, Inc.
325 N. Santa Anita Ave,
Arcadia, CA 91006

<Cohen@Myri.com>
Tel: 818-821-5555
Fax: 818-821-5316

Admin Info

(1) <u>WG Name</u>: **MessageWay (MsgWay)**

(2) <u>WG-Chair</u>: **Danny Cohen <Cohen@myri.com>**

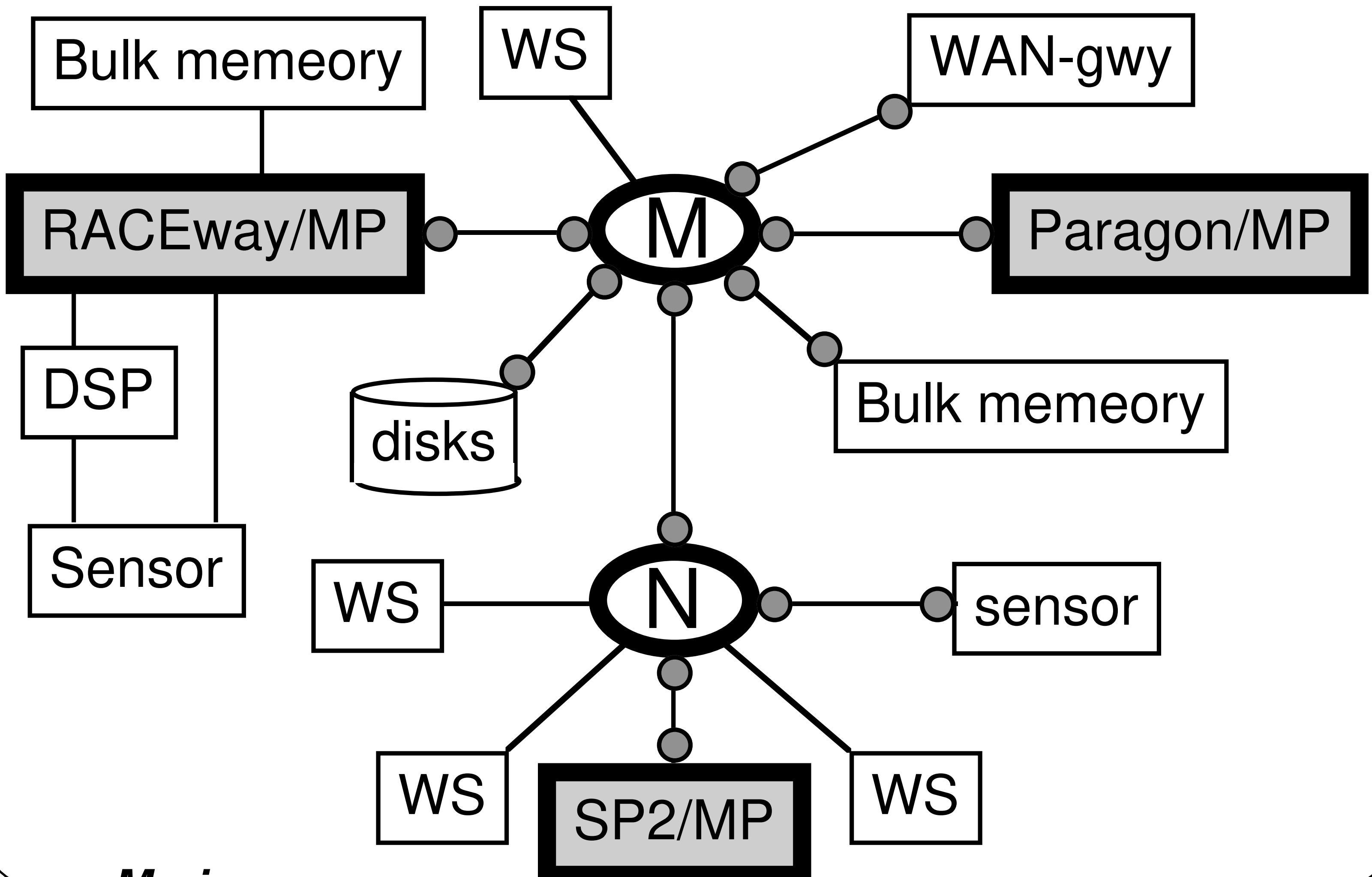
(3) <u>INTERNET Area Director</u>:
Frank Kastenholz <Kasten@ftp.com>

(4) <u>Mailing List</u>:
General Discussion: **<MsgWay@myri.com>**
To Subscribe: **<MsgWay-request@myri.com>**
Archive via:
URL = ftp://ftp.isi.edu/msgway/msgway.mail

MsgWay

1. The Problem
2. The Approach
3. $\boxed{\text{Draft EEP}}$

Example MsgWay



MsgWay Protocols

* E/E protocol ("core")
* R/R-protocol (mapping,routing, etc.)
* Address assignment
* Resource-mngmnt protocol

Only the E/E protocol will be initially defined (+demonstrated). The others may start manually, and be automated later.

GOALS AND MILESTONES

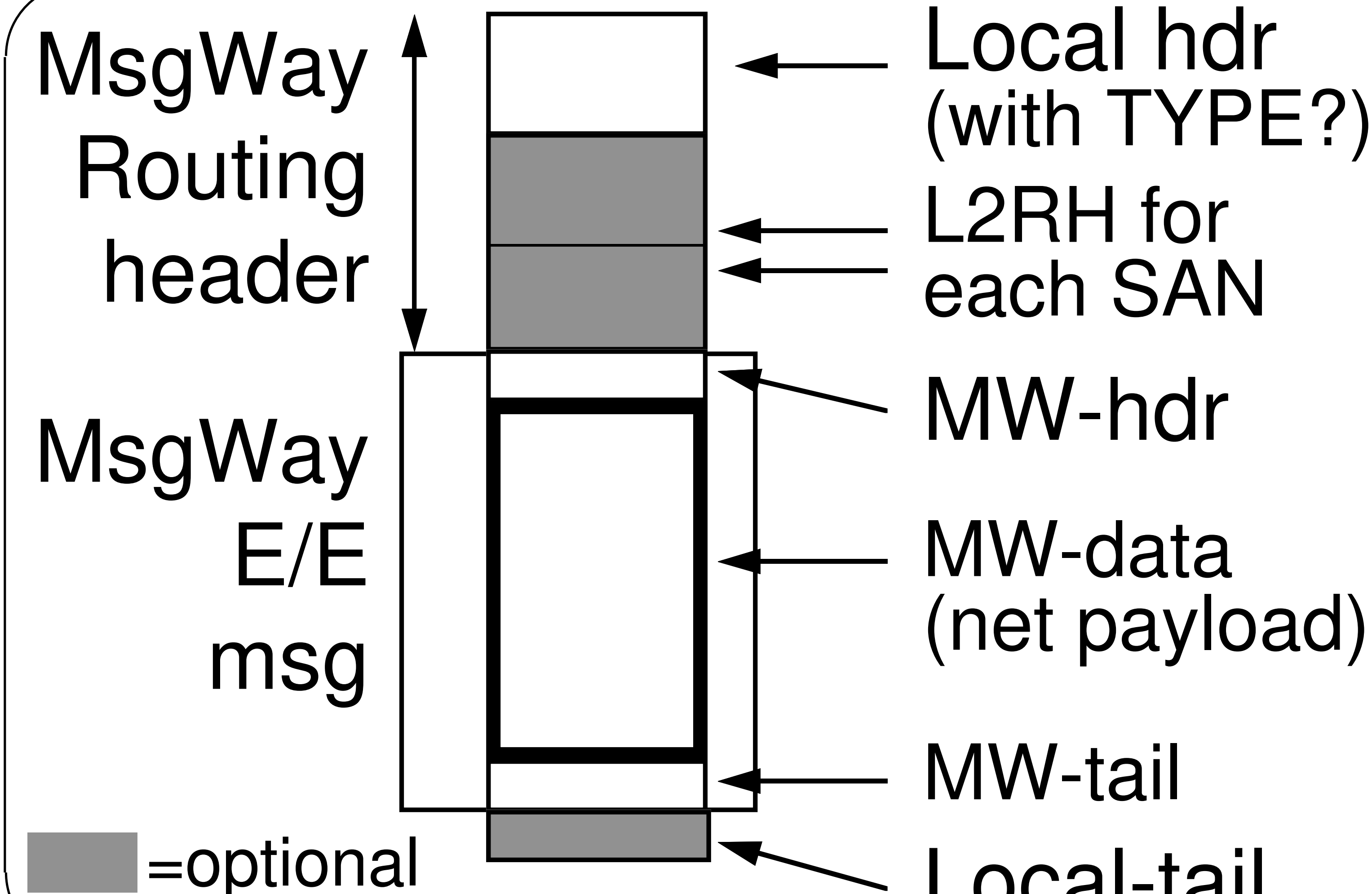
Jul'95: Initial draft specification for the
(IETF33) MsgWay-**EEP** (End-to-End protocol)

Dec'95: Final specification for the
(IETF34) MsgWay-**EEP** (End-to-End protocol)

Mar'96: Final specification of the
(IETF35) MsgWay-**RRP** (Router-to-Router protocol)

.Jun'96: Final specification of the MsgWay-**REDAP**
(IETF36) (Resource Discovery and Allocation Prot'l)

Anatomy of a MW-Msg



MW E/E Header

Field	Description
Destination-addr	Dynamic addr>63
Priority / Version	This is version 5
E/E msg type with extension	Extended msg-type (e.g., with address)
Pad-L / Msg Length	In bytes. The ML includes the PL.
Source-addr	Dynamic addr
Free	For arbitrary use

L2 Routing Headers

Length (bytes) of this block	8N bytes,<64
The native routing header of this SAN	
Extension of the native routing header of this SAN, if needed.	

E/E-Message-Types

- L2-fwd
- Public data (e.g., IP,ATM,...)
- Private data (e.g., net / DMA)
- RRP

Msg-Type-extensions may be used for earliest demuxing (“flows”), and for alternative formats (e.g., addresses).

R/R-Message-Types

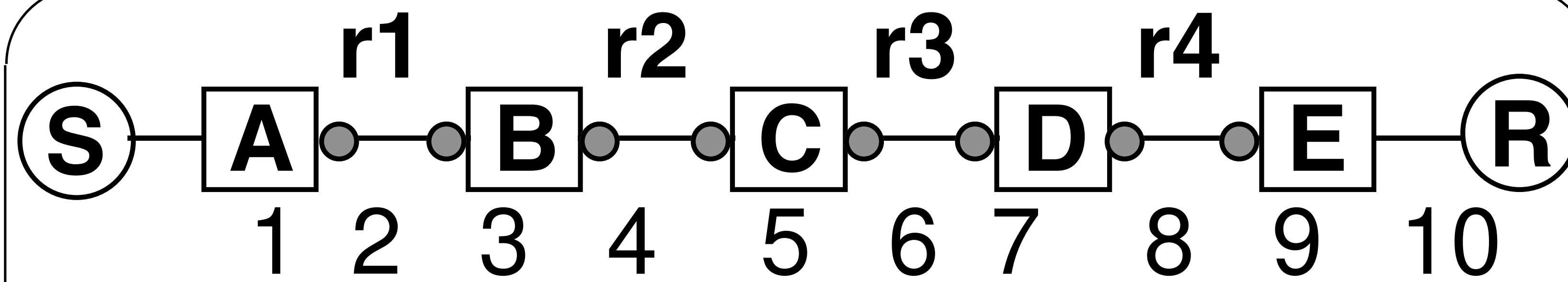
- What-are-you?
- I-am(type,name,ID,chunk,MTU,F/C)
- Give-tables(my-chunk)
- Take-tables(L,chunks,name...)

RRP msgs are encapsulated inside msgs with E/E headers

Message Forwarding (L2 and L3)

(Odd numbered page)

Message Flow

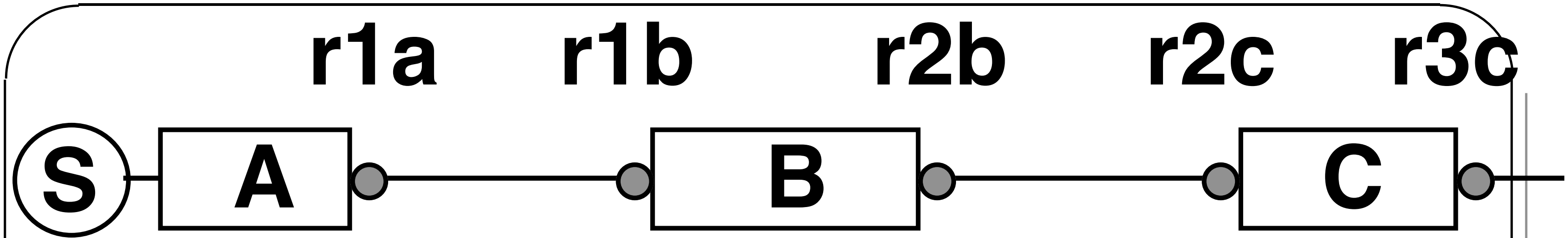


1 From Sender, thru SAN-A, to r1a
2 From r1a, over RR-link, to r1b
3 From r1b, thru SAN-2, to r2b
4 From r2b, over RR-link, to r2c

....

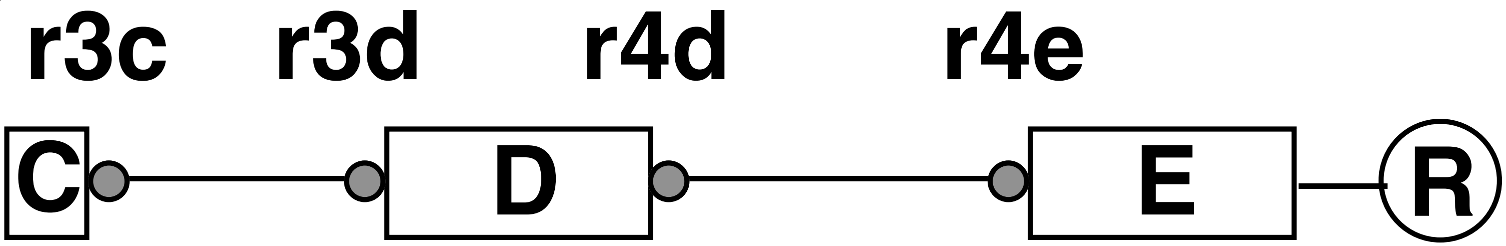
9 From r4e, thru SAN-E, to Receiver
10 As arriving at the final Receiver

L3 Forwarding / 1



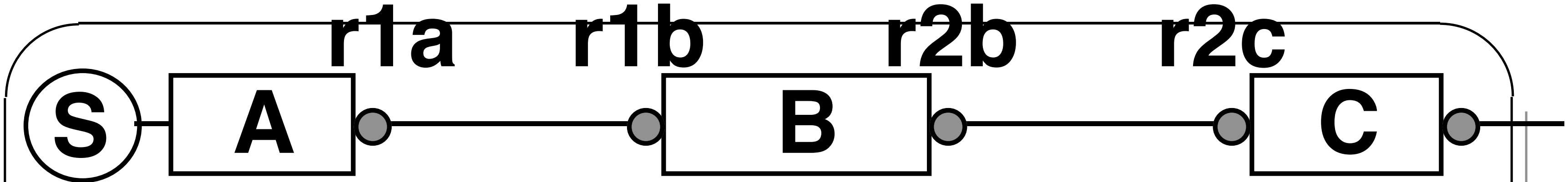
S:	r1a:	r1b:	r2b:	r2c:
A:D=r1a		B:D=r2b		C:D=r3c
A:T=L3F		B:T=L3F		C:T=L3F
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail
1	2	3	4	5

L3 Forwarding / 2



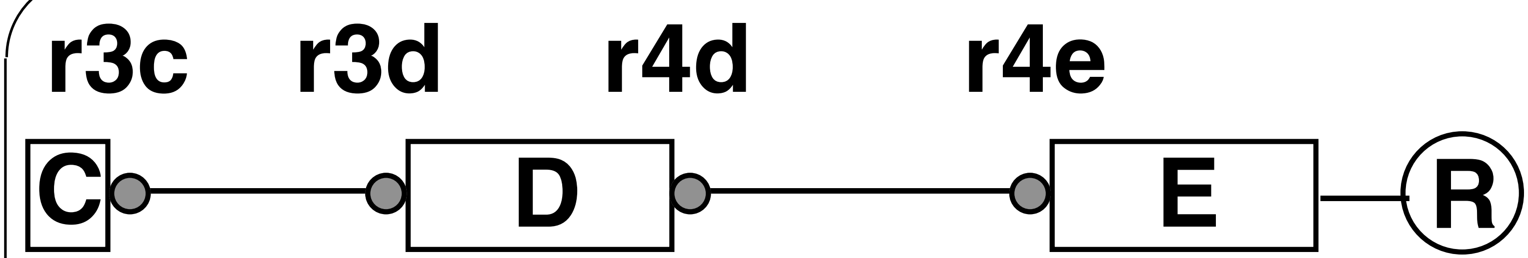
r3c:	r3d:	r4d:	r4e:	R:
	D:D=r4d		E:D=R	
	D:T=L3F		E:T=data	E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail
6	7	8	9	10

L2 Forwarding / 1



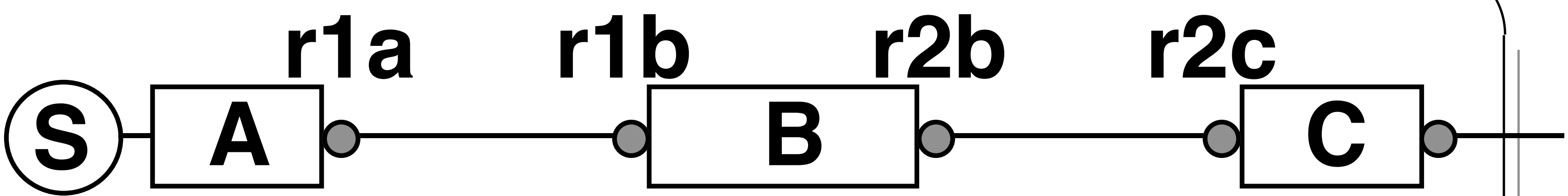
S:	r1a:	r1b:	r2b:	r2c:
A:D=r1a				
A:T=L2F				
B:D=r2b	B:D=r2b	B:D=r2b		
B:T=L2F	B:T=L2F	B:T=L2F		
C:D=r3c	C:D=r3c	C:D=r3c	C:D=r3c	C:D=r3c
C:T=L2F	C:T=L2F	C:T=L2F	C:T=L2F	C:T=L2F
D:D=r4d	D:D=r4d	D:D=r4d	D:D=r4d	D:D=r4d
D:T=L2F	D:T=L2F	D:T=L2F	D:T=L2F	D:T=L2F
E:D=R	E:D=R	E:D=R	E:D=R	E:D=R
E:T=data	E:T=data	E:T=data	E:T=data	E:T=data
MW:R=D	MW:R=D	MW:R=D	MW:R=D	MW:R=D
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

L2 Forwarding / 2



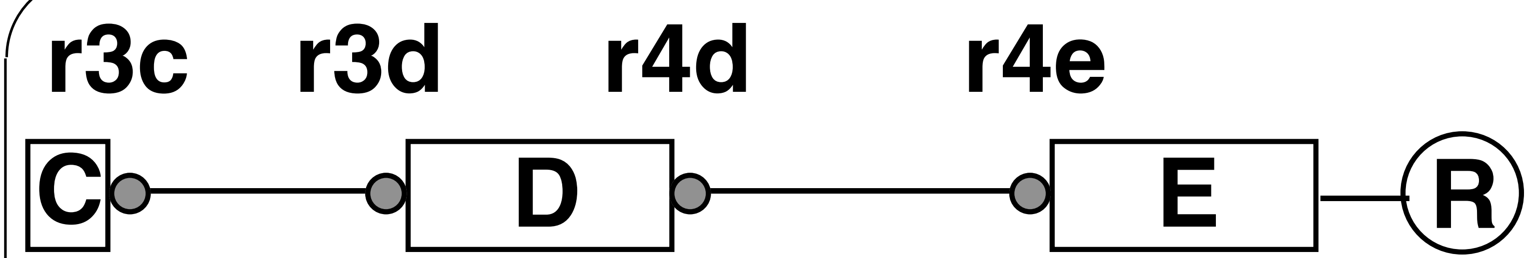
r3c:	r3d:	r4d:	r4e:	R:
D:D=r4d	D:D=r4d			
D:T=L2F	D:T=L2F			
E:D=R	E:D=R	E:D=R	E:D=R	
E:T=data	E:T=data	E:T=data	E:T=data	E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

L2+L3 Forwarding / 1



S:	r1a:	r1b:	r2b:	r2c:
A:D=r1a				
A:T=L2F				
B:D=r2b	B:D=r2b	B:D=r2b		
B:T=L2F	B:T=L2F	B:T=L2F		
C:D=r3c	C:D=r3c	C:D=r3c	C:D=r3c	C:D=r3c
C:T=L3F	C:T=L3F	C:T=L3F	C:T=L3F	C:T=L3F
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

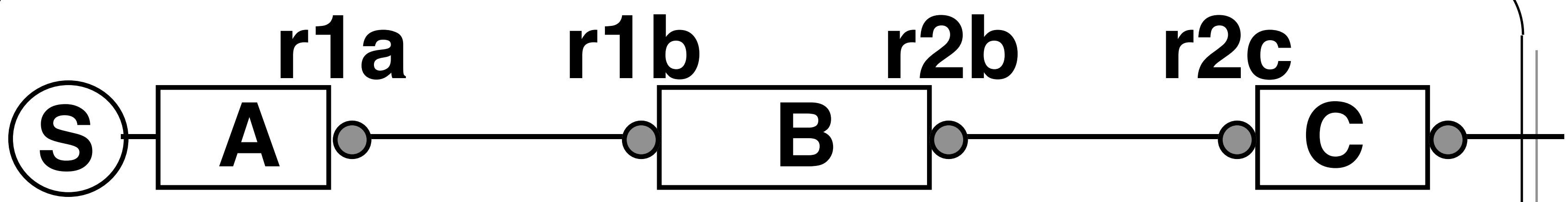
L2+L3 Forwarding / 2



r3c:	r3d:	r4d:	r4e:	R:
	D:D=r4d		E:D=R	
	D:T=L3F		E:T=data	E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

L3+L2 Forwarding / 1

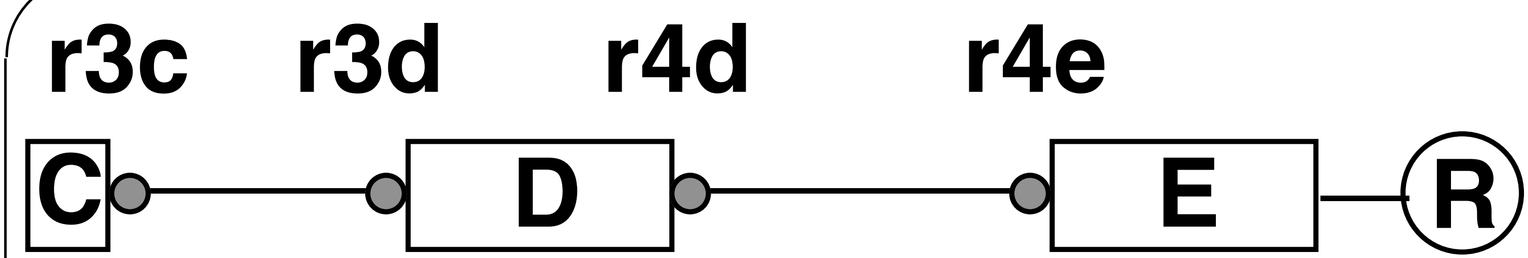
r2c knows how to find R



S:	r1a:	r1b:	r2b:	r2c:
				C:D=r3c
				C:T=L2F
				D:D=r4d
				D:T=L2F
A:D=r1a		B:D=r2b		E:D=R
A:T=L3F		B:T=L3F		E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

L3+L2 Forwarding / 2

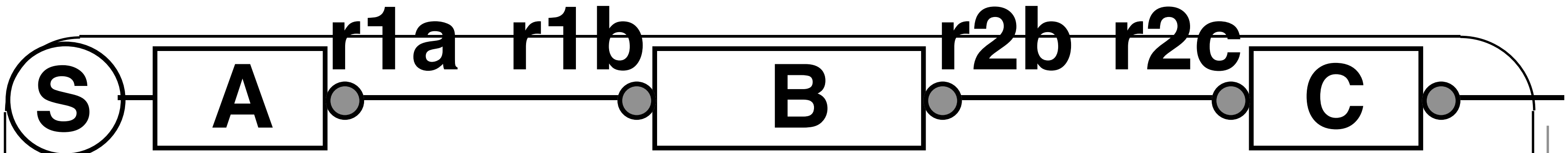
r2c knows how to find R



r3c:	r3d:	r4d:	r4e:	R:
D:D=r4d	D:D=r4d			
D:T=L2F	D:T=L2F			
E:D=R	E:D=R	E:D=R	E:D=R	
E:T=data	E:T=data	E:T=data	E:T=data	E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

L3+L2 Forwarding / 1

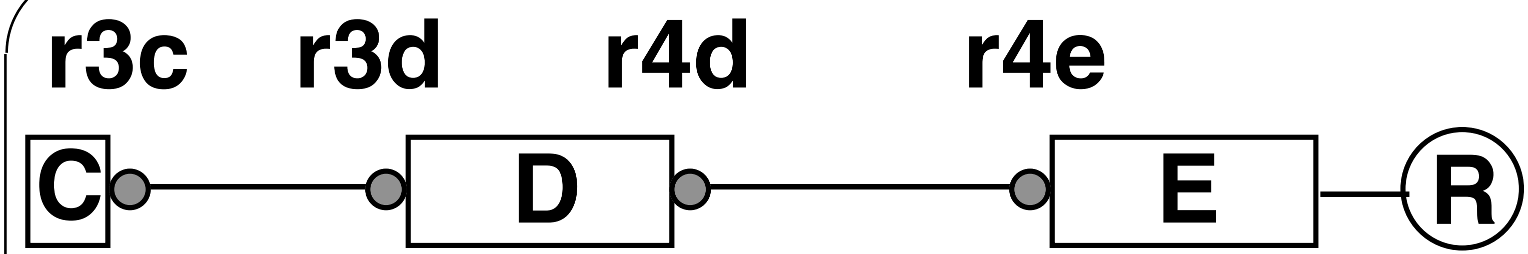
S knows how to find R (from r2)



S:	r1:	B:	r2:	C:
A:D=r1a		B:D=r2b		
A:T=L3F		B:T=L3F		
MW:T=L2F	MW:T=L2F	MW:T=L2F	MW:T=L2F	
MW:D=r2c	MW:D=r2c	MW:D=r2c	MW:D=r2c	
C:D=r3c	C:D=r3c	C:D=r3c	C:D=r3c	C:D=r3c
C:T=L2F	C:T=L2F	C:T=L2F	C:T=L2F	C:T=L2F
D:D=r4d	D:D=r4d	D:D=r4d	D:D=r4d	D:D=r4d
D:T=L2F	D:T=L2F	D:T=L2F	D:T=L2F	D:T=L2F
E:D=R	E:D=R	E:D=R	E:D=R	E:D=R
E:T=data	E:T=data	E:T=data	E:T=data	E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail
MW:tail	MW:tail	MW:tail	MW:tail	

L3+L2 Forwarding / 2

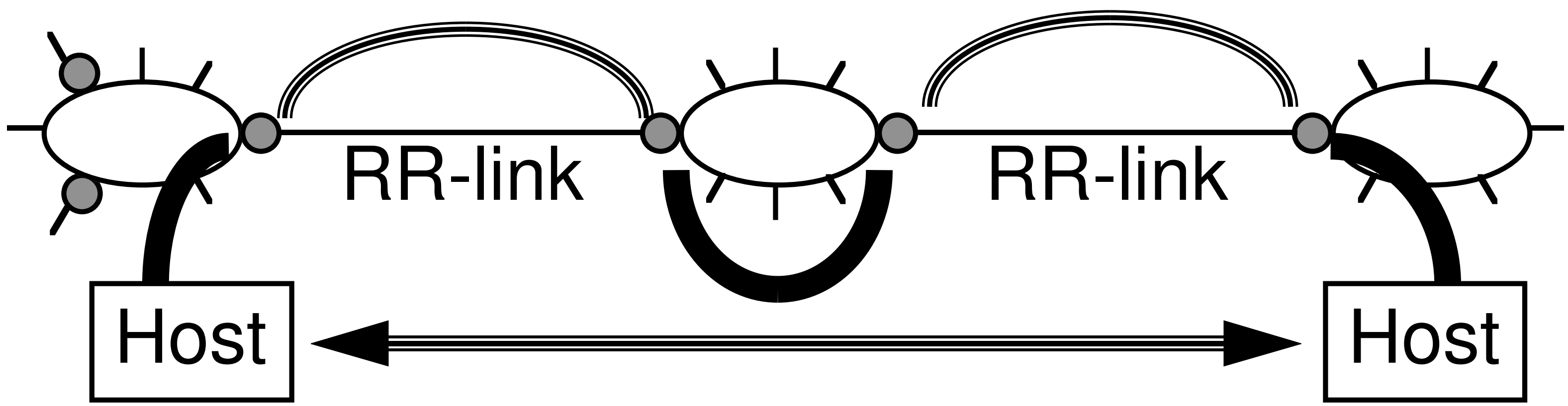
S knows how to find R (from r2)



r3c:	r3d:	r4d:	r4e:	R:
D:D=r4d	D:D=r4d			
D:T=L2F	D:T=L2F			
E:D=R	E:D=R	E:D=R	E:D=R	
E:T=data	E:T=data	E:T=data	E:T=data	E:T=data
MW:D=R	MW:D=R	MW:D=R	MW:D=R	MW:D=R
MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data	MW:T=Data
Payload	Payload	Payload	Payload	Payload
MW:tail	MW:tail	MW:tail	MW:tail	MW:tail

Summary

E/E + R/R



MsgWay defines: E/E format, inter-SAN R/R, error-indication, but ***NOT***: intra-SAN msgs and intra-SAN R/R, physical links, flow control,error-check

MsgWay Features

Source Route and/or Addresses
Fine grain addressing
DMA orientation
Padding / Alignment
Cut-Thru or Stop-&-Forward
Self Mapping (for Plug-&-Play)
Support of IP+
Deferred binding (by receivers)
Dynamic short addresses

The End