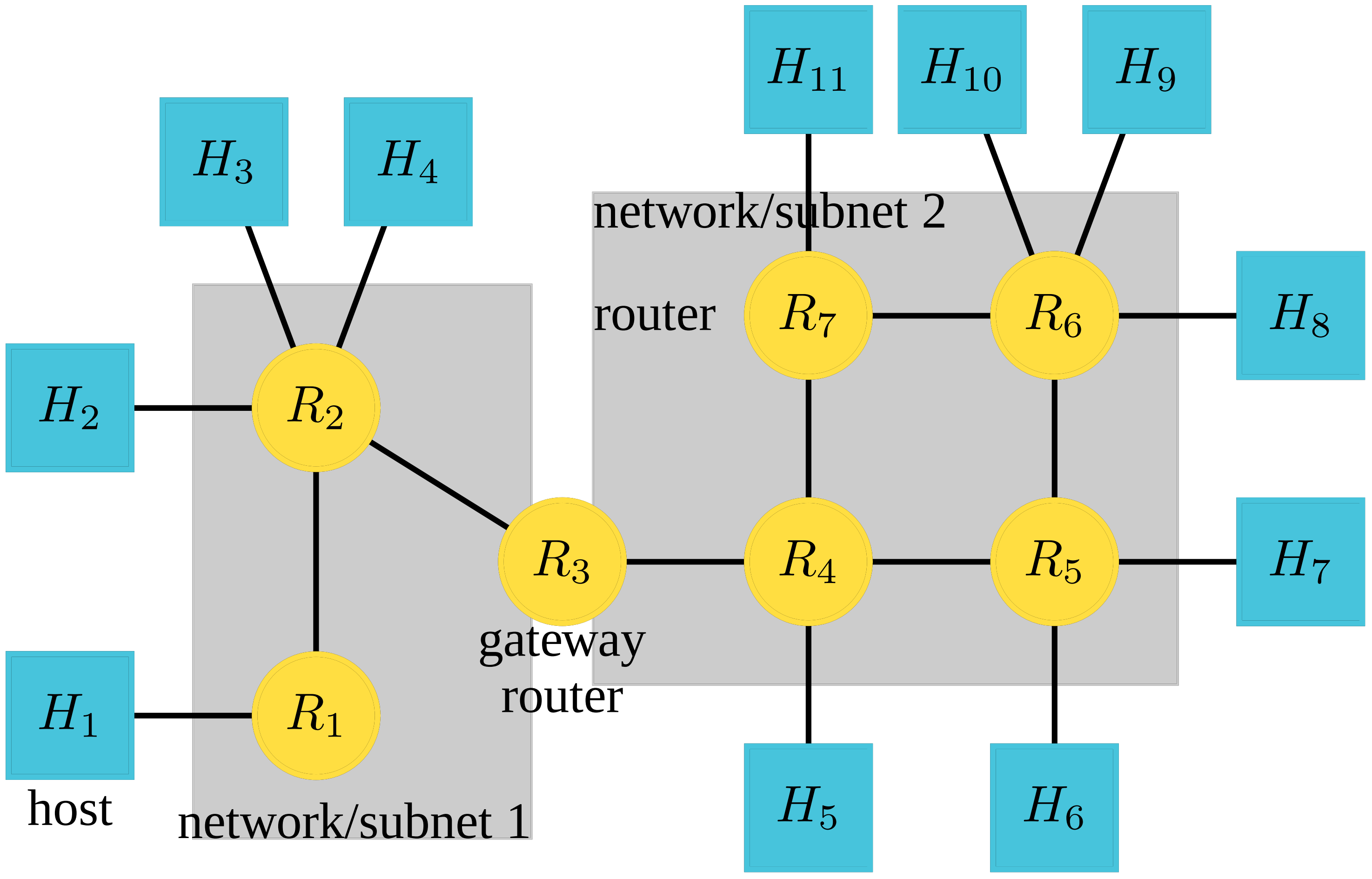
Routing and Internetworking

A Network



- **gateway** links networks together to form **internet**
- **routing** plans path through network

Internets

An internet can be

- **homogeneous internets**
  - all same type of network
  - no need to convert frames
- **heterogeneous internets**
  - free mix of network technologies
  - need to convert between frame formats
  - convert between CO and CL?

Switching the Data

- **circuit switched networks**
  - physical communication channel to be set up between two hosts
  - wasteful if bursty data

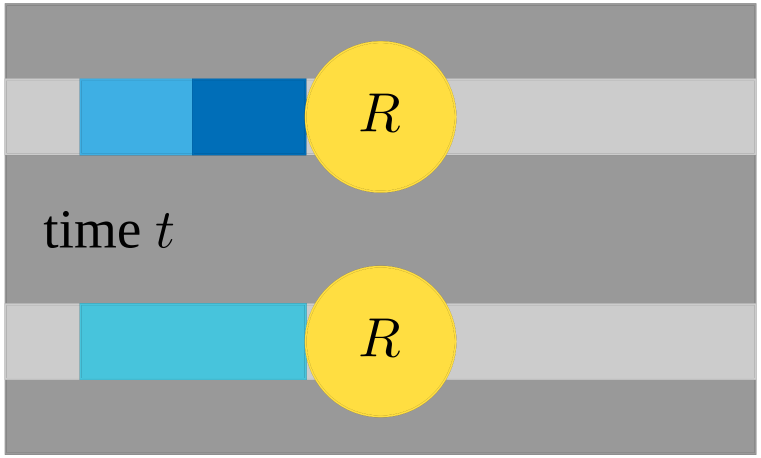
  e.g. PSTN

- **message switched networks**
  - sends data in **messages**, which can share a channel
  - **store-and-forward**

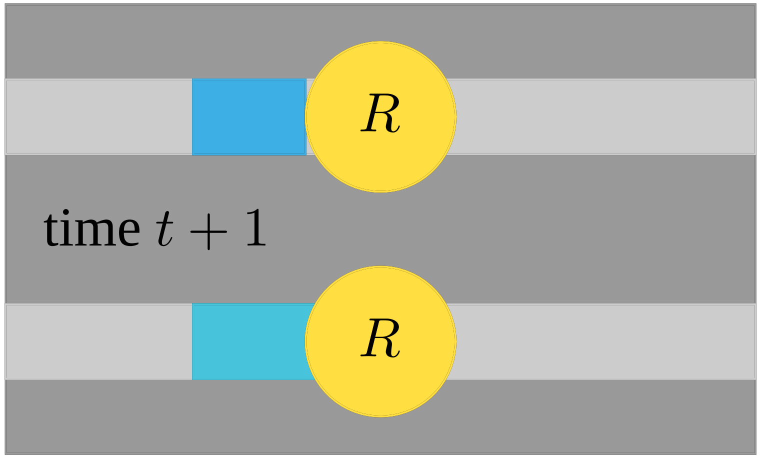
  e.g. Post Office

- **packet switched networks**
  - similar to message switching
  - limit on size $\longrightarrow$ data sent in one or more **packets**

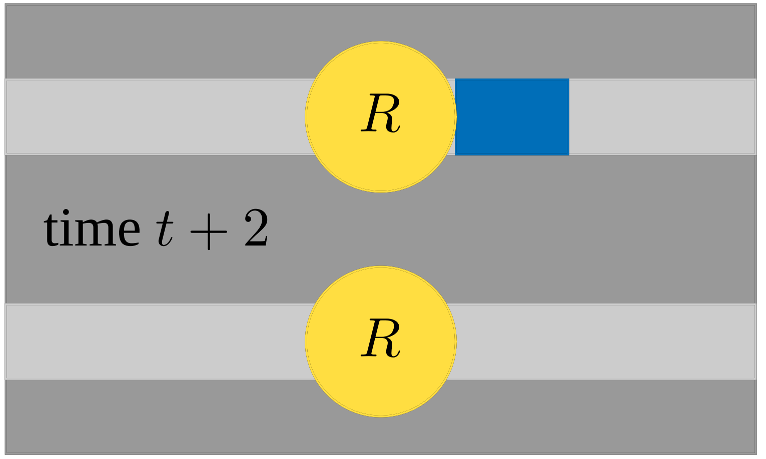
Store-and-forward (1)



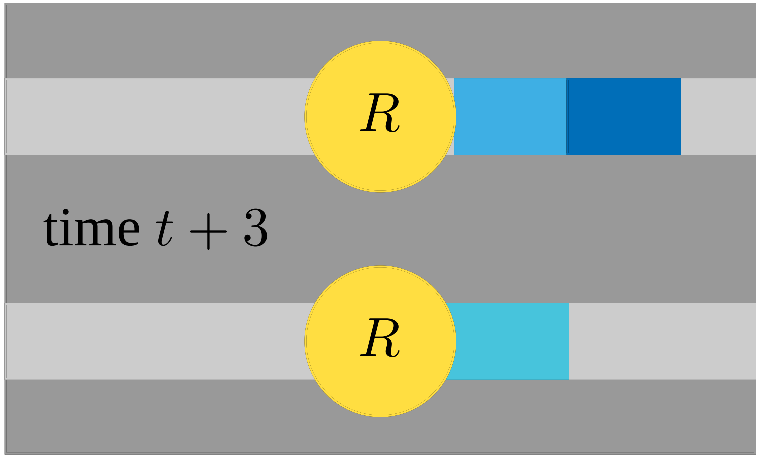
Store-and-forward (2)



Store-and-forward (3)



Store-and-forward (4)



- delay $\propto$ packet size

Example: Reducing Store-and-forward Delays

A router operating on a PSN using store-and-forward has links into it operating at $1Mbs^{-1}$, and the router is found to delay packets by $0.1s$. How many pieces must you fragment the packet into to reduce this delay to $0.02s$, and how large are those packets?

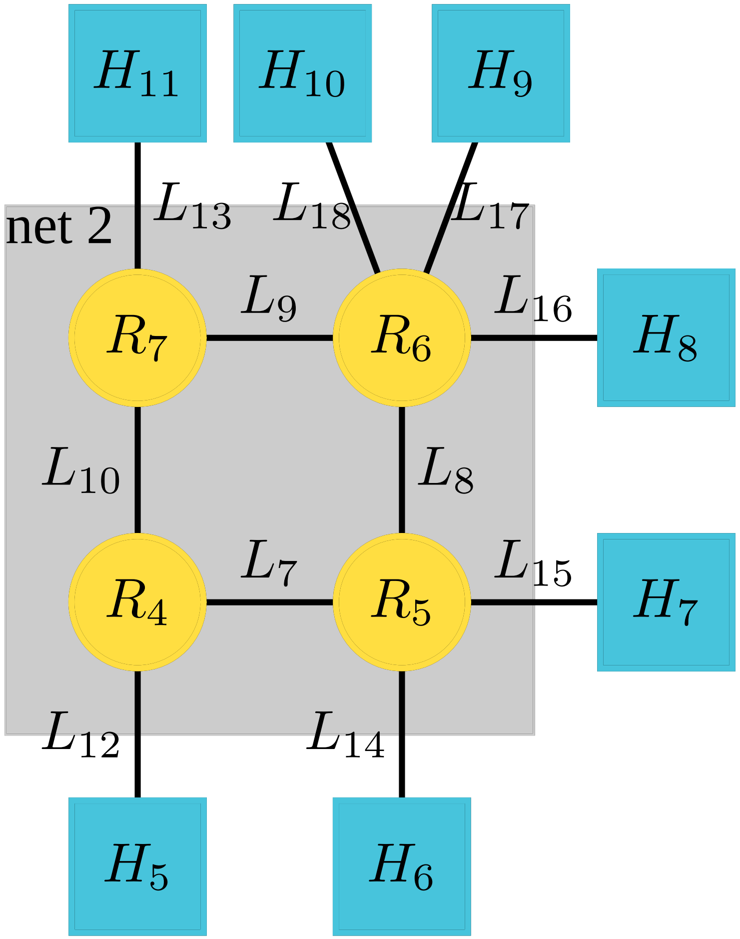
delay $\propto$ *packet size*

$$\frac{\textit{delay}}{\textit{packet size}} = \textit{constant}$$

$0.1/1 = 0.02/s \rightarrow s = 0.2$, *hence 5 packets are required for each of the old packets*

packet size= delay = $\frac{0.02 \times 1 \times 10^6}{8}$ $=$ *2500 bytes*

Virtual Circuits



R_7	
to/from	from/to

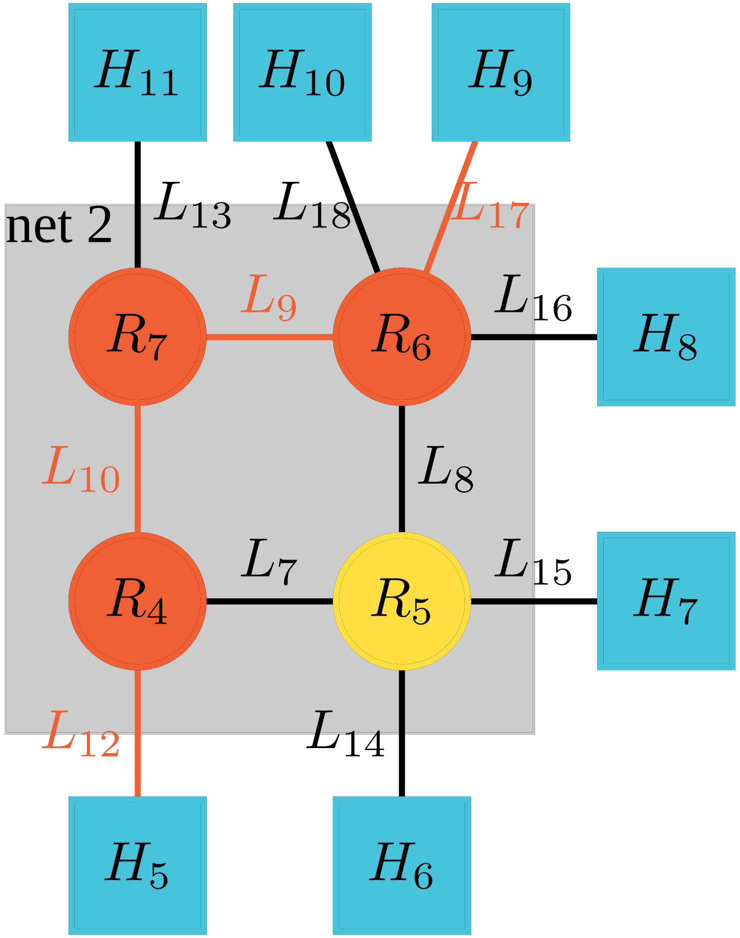
R_6	
to/from	from/to

R_4	
to/from	from/to

R_5	
to/from	from/to

- *Virtual Circuits* allow some features of CO to be present in PSN
- Divide links by **virtual circuit identifiers (VCI)** $L_x(1), L_x(2), L_x(3), \ldots$
- Connect links up VCIs

Virtual Circuits



R_7	
to/from	from/to
$L_{10}(1)$	$L_9(1)$

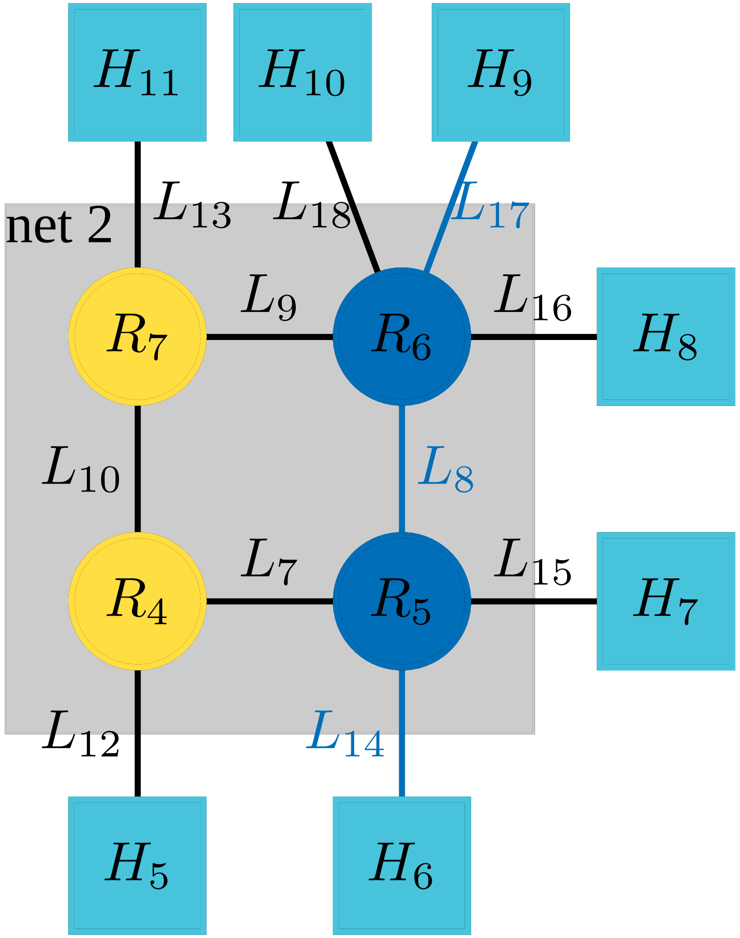
R_6	
to/from	from/to
$L_9(1)$	$L_{17}(1)$

R_4	
to/from	from/to
$L_{12}(1)$	$L_{10}(1)$

R_5	
to/from	from/to

- Connect $H_5 \leftrightarrow H_9$

Routing Tables for VC



$H_5 \leftrightarrow H_9$

R_7	
to/from	from/to
$L_{10}(1)$	$L_9(1)$

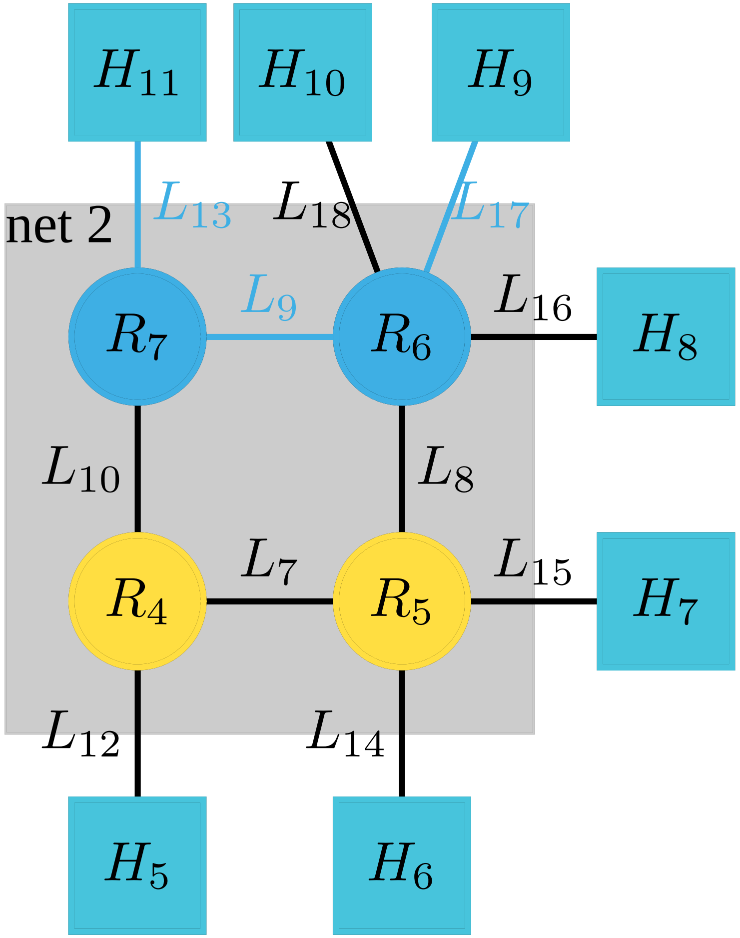
R_6	
to/from	from/to
$L_9(1)$	$L_{17}(1)$
$L_8(1)$	$L_{17}(2)$

R_4	
to/from	from/to
$L_{12}(1)$	$L_{10}(1)$

R_5	
to/from	from/to
$L_{14}(1)$	$L_8(1)$

Connect $H_6 \leftrightarrow H_9$

Routing Tables for VC



$H_5 \leftrightarrow H_9$ $H_6 \leftrightarrow H_9$

R_7

to/from	from/to
$L_{10}(1)$	$L_9(1)$
$L_{13}(1)$	$L_9(2)$

R_6

to/from	from/to
$L_9(1)$	$L_{17}(1)$
$L_8(1)$	$L_{17}(2)$
$L_9(2)$	$L_{17}(3)$

R_4

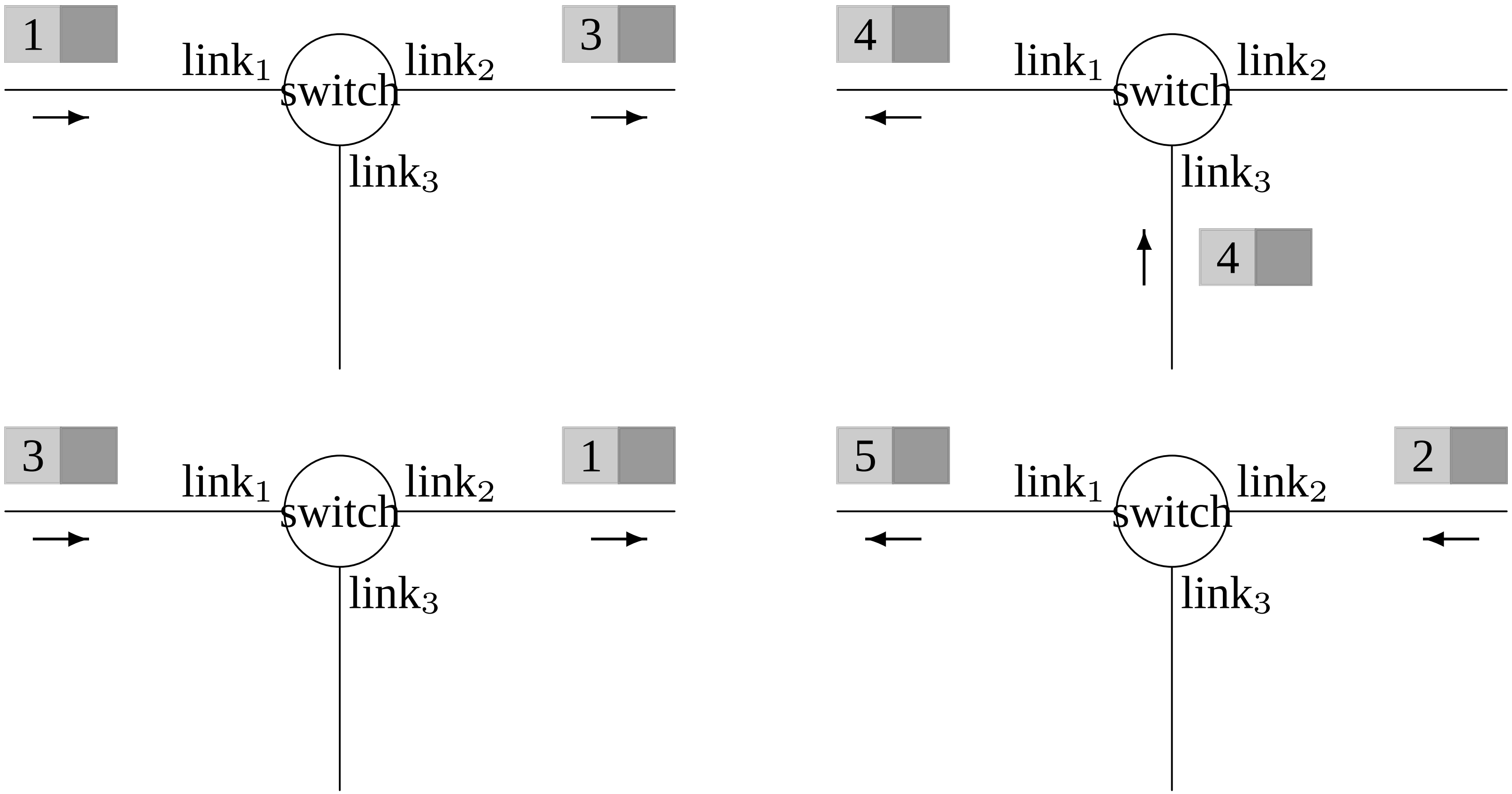
to/from	from/to
$L_{12}(1)$	$L_{10}(1)$

R_5

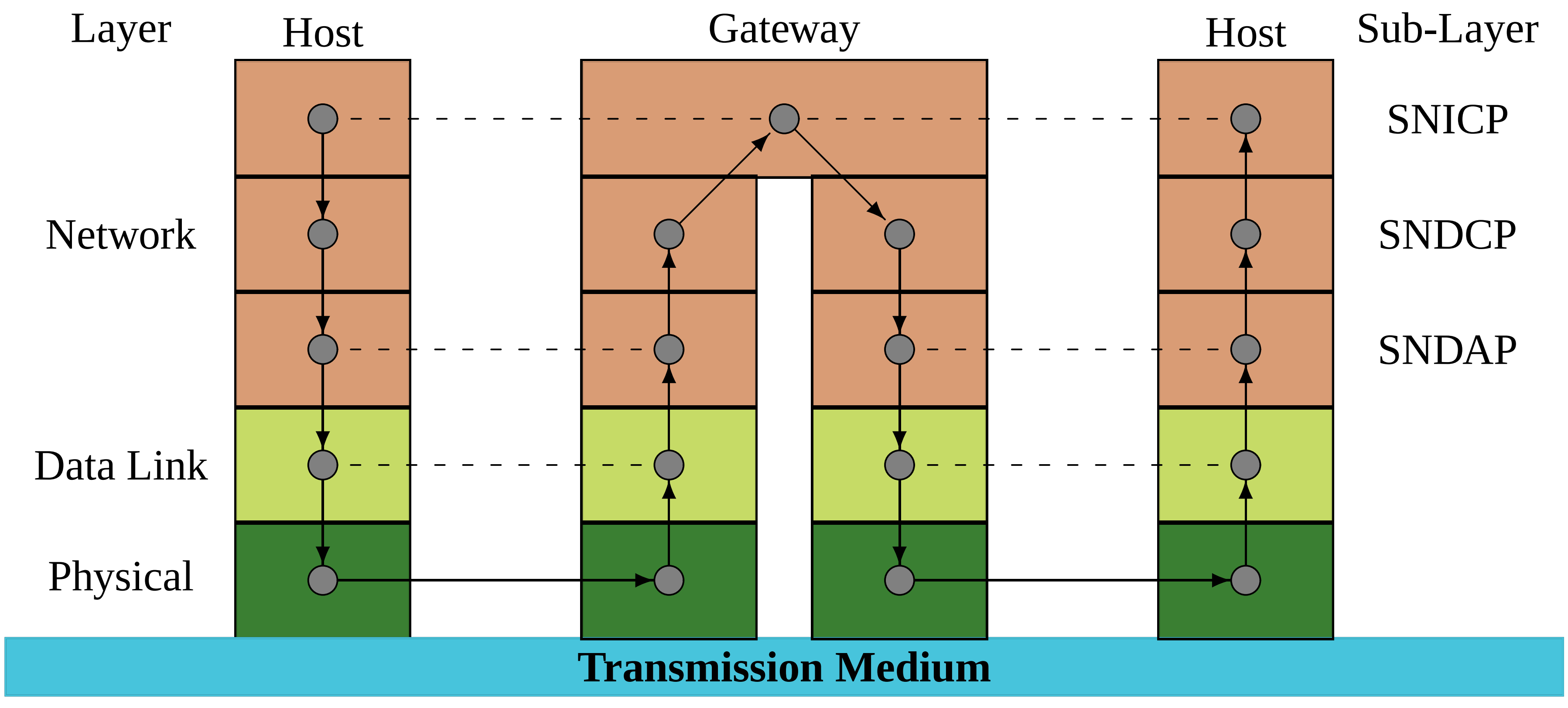
to/from	from/to
$L_{14}(1)$	$L_8(1)$

Connect $H_{11} \leftrightarrow H_9$

Worksheet: Virtual Circuit Identifiers

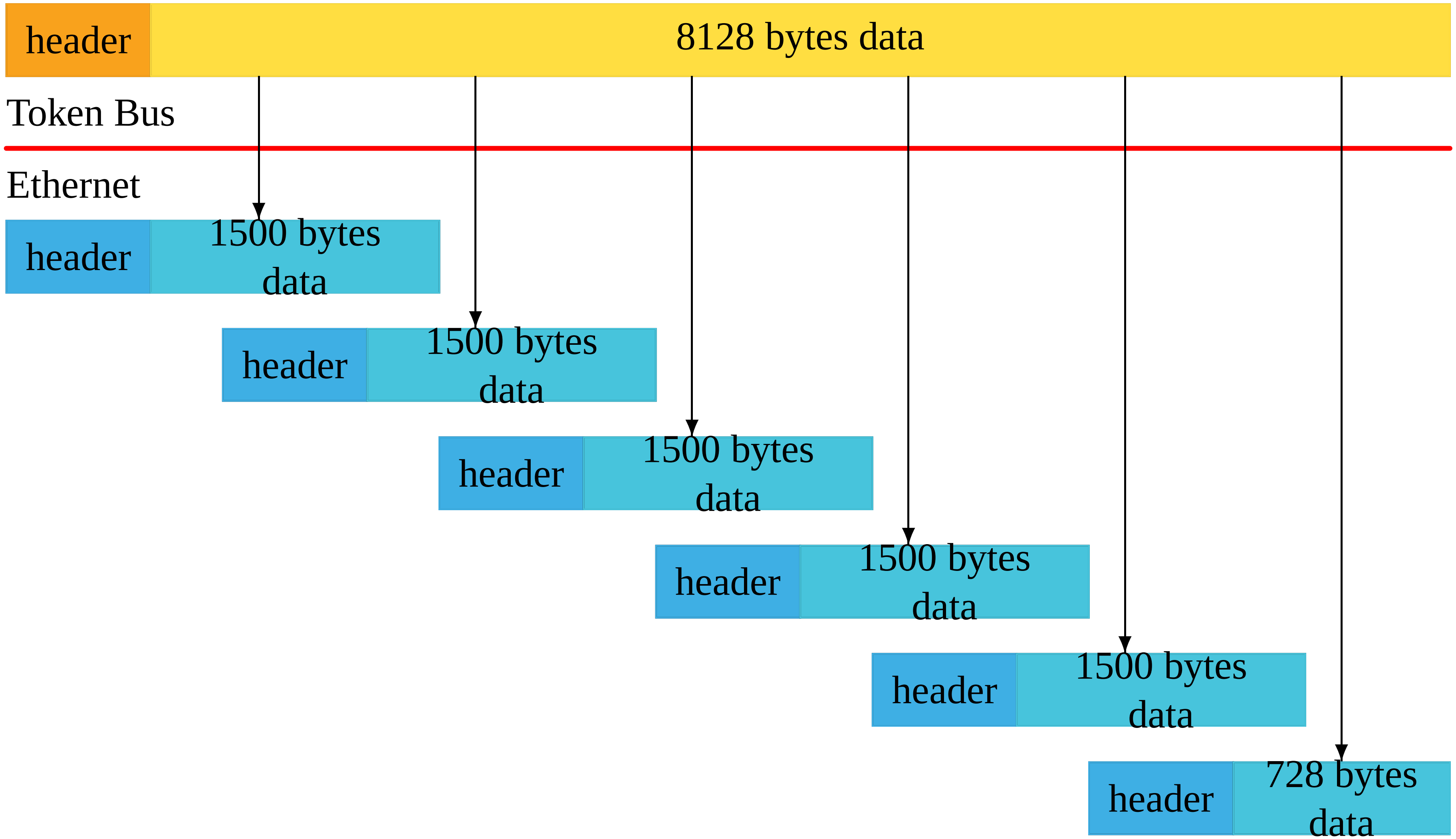


The OSI Internetworking Model

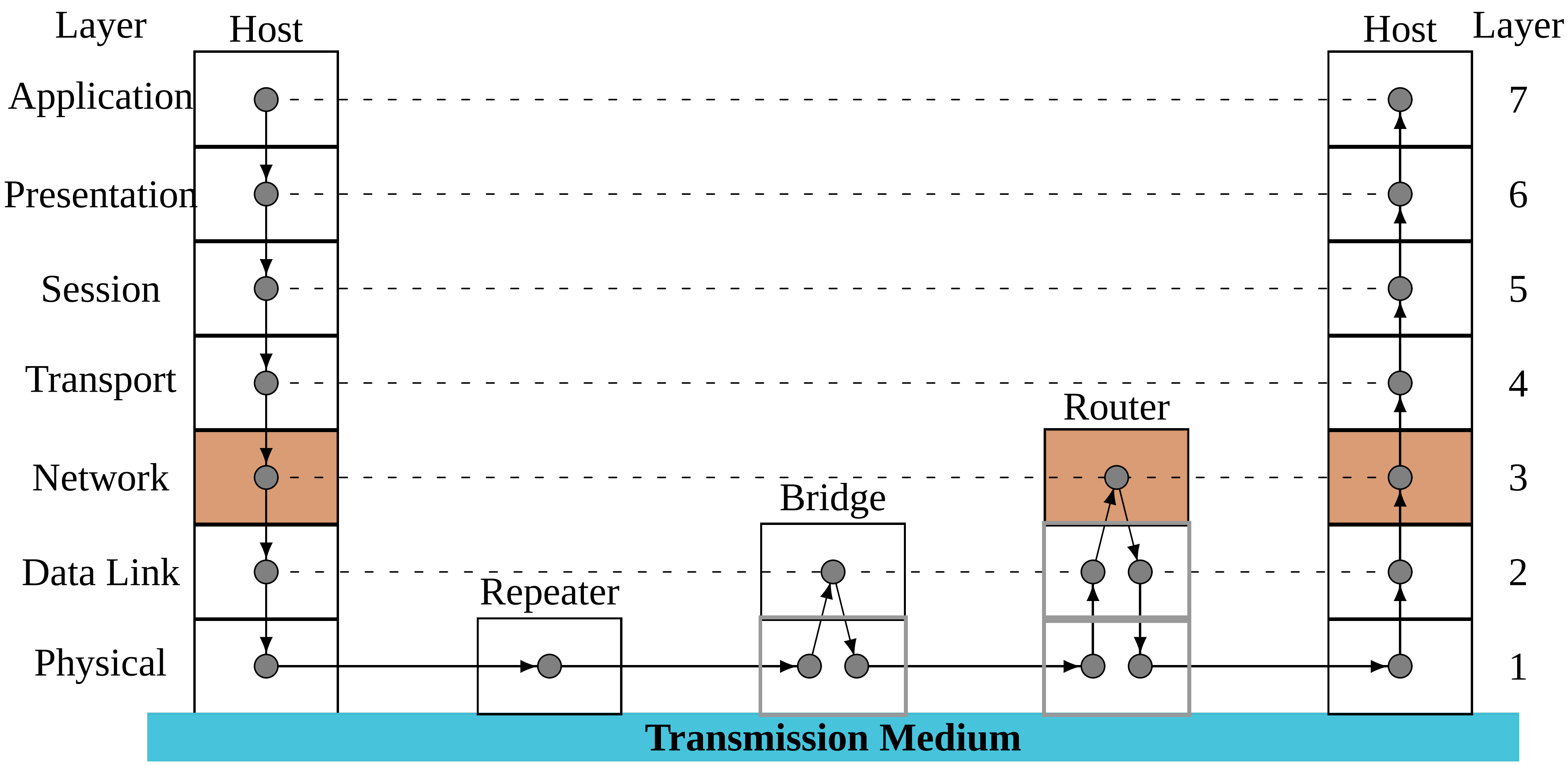


- **Subnetwork Independent Convergence Protocol (SNICP)**
- **Subnetwork Dependent Convergence Protocol (SNDCP)**
- **Subnetwork Dependent Access Protocol (SNDAP)**

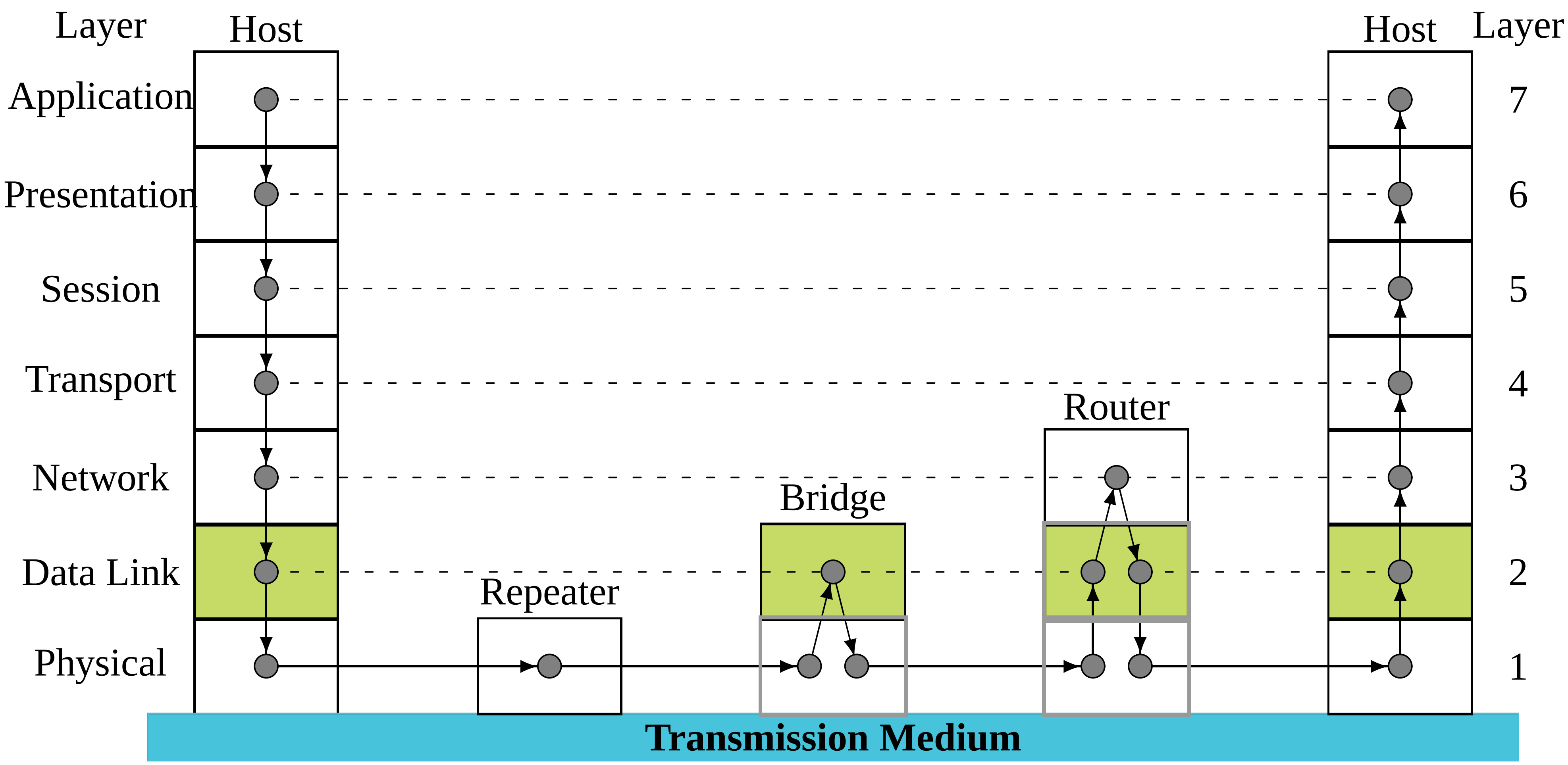
Fragmentation



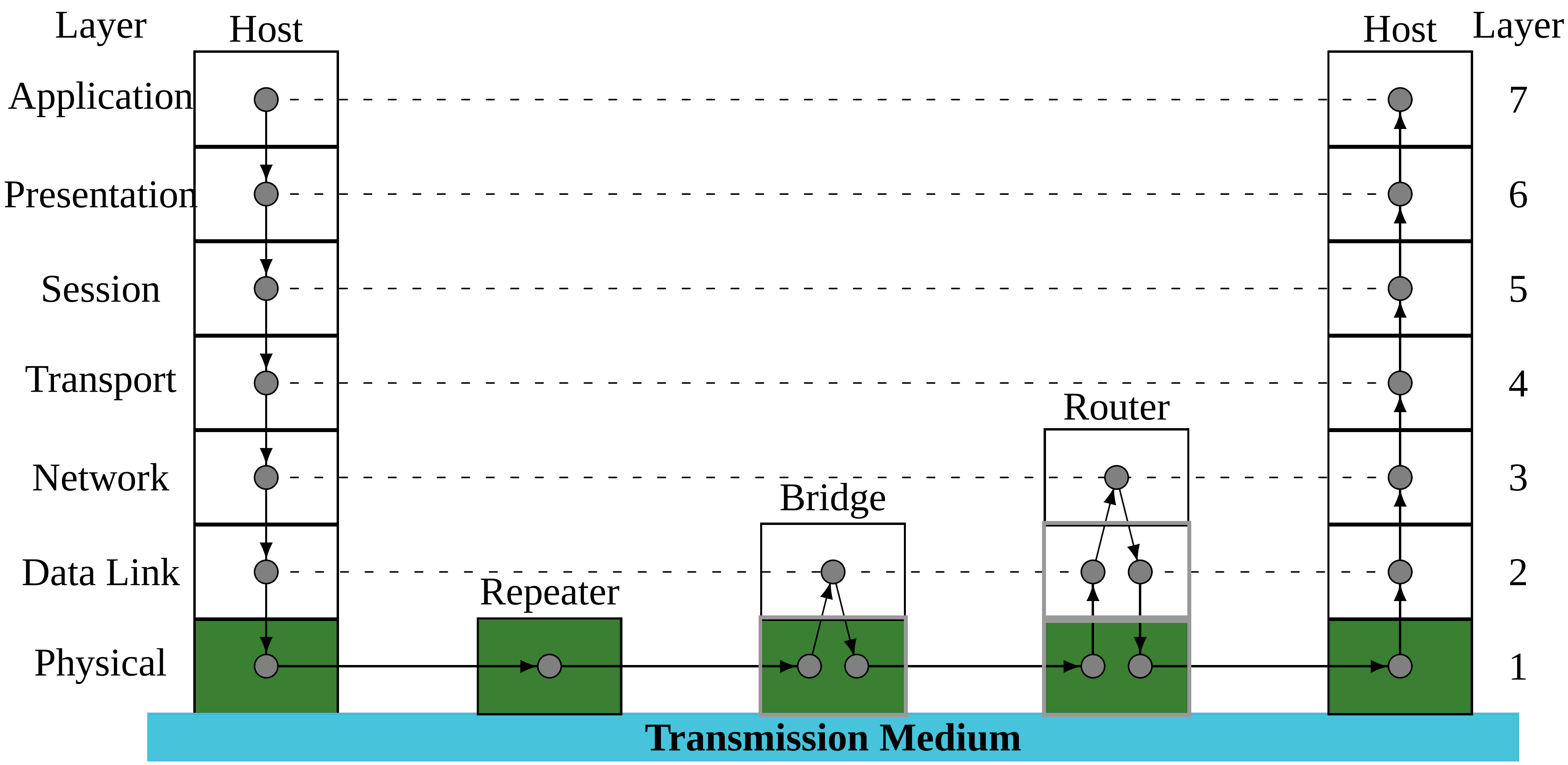
Homogeneous Internetworking: Network Layer



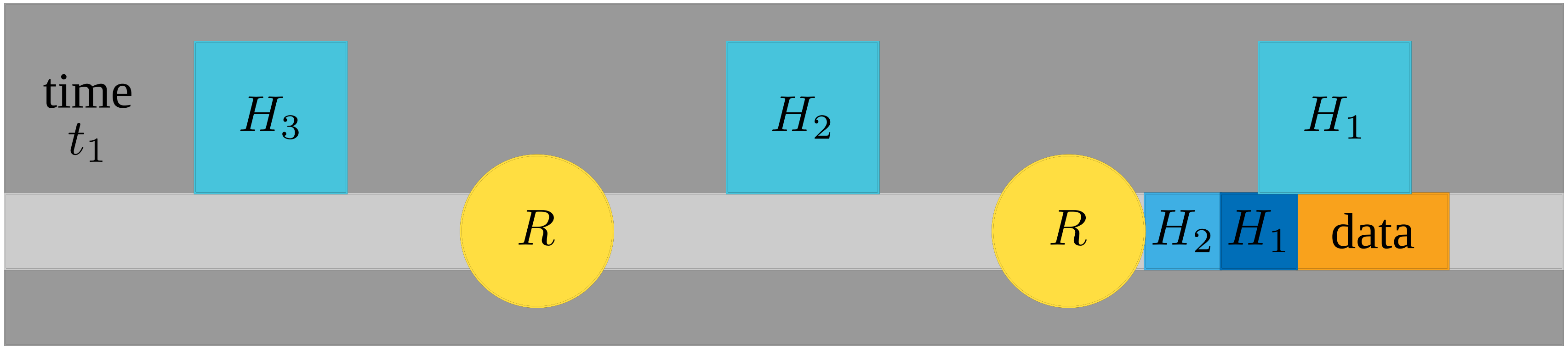
Homogeneous Internetworking: Data Link Layer



Homogeneous Internetworking: Physical Layer

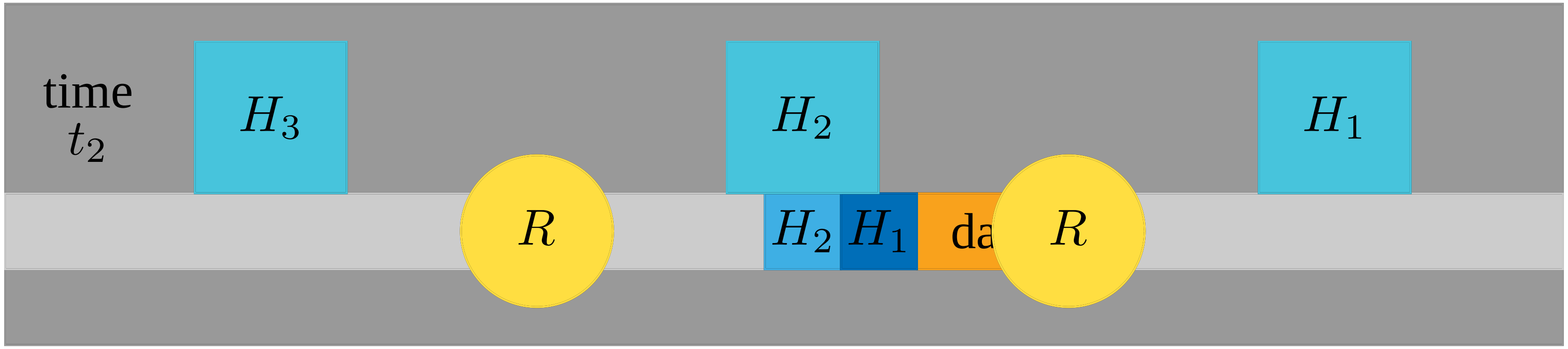


Repeater (1)

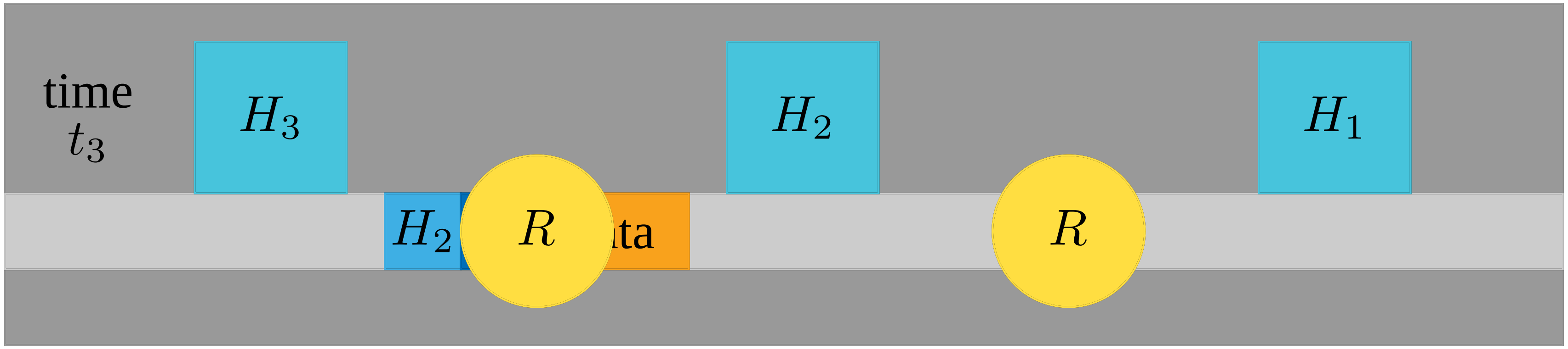


- ‘repeats’ frames bit by bits; no processing of frames or routing

Repeater (2)

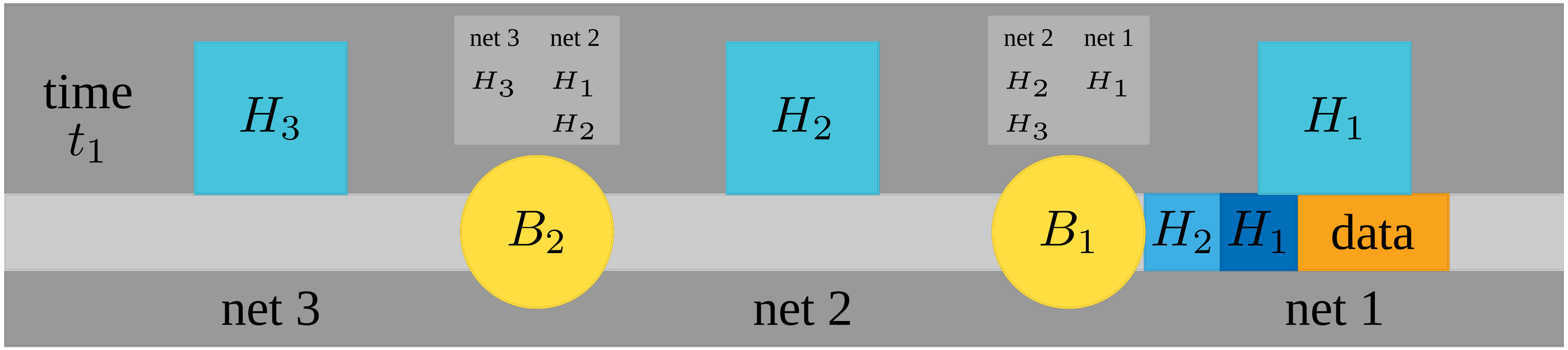


Repeater (3)



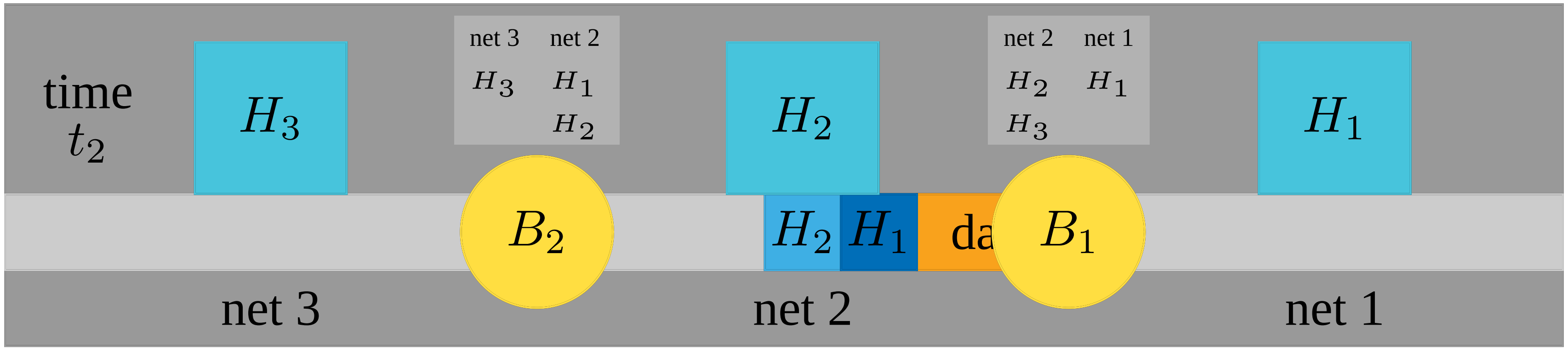
- also called **hub**

Bridge (1)

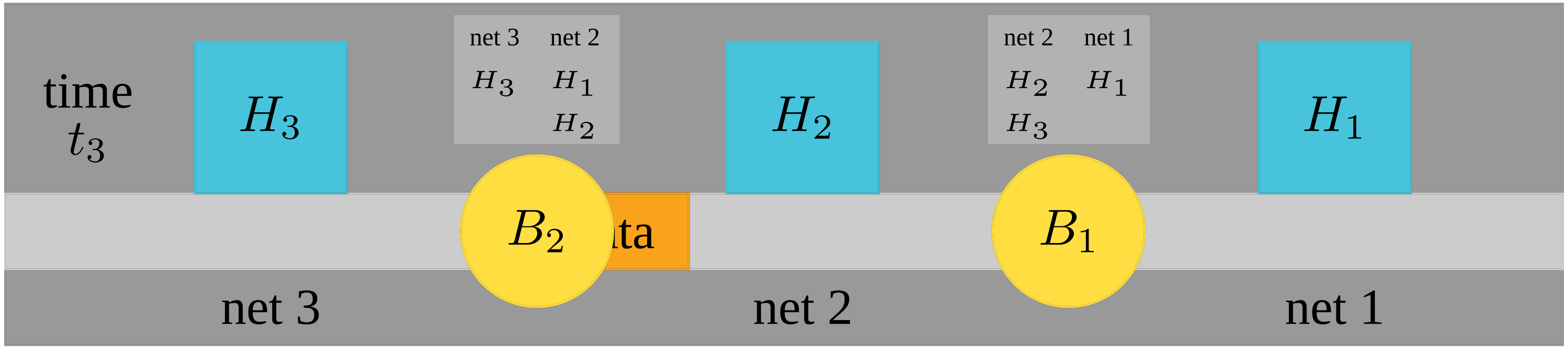


- copies frames as required; store-and-forward

Bridge (2)

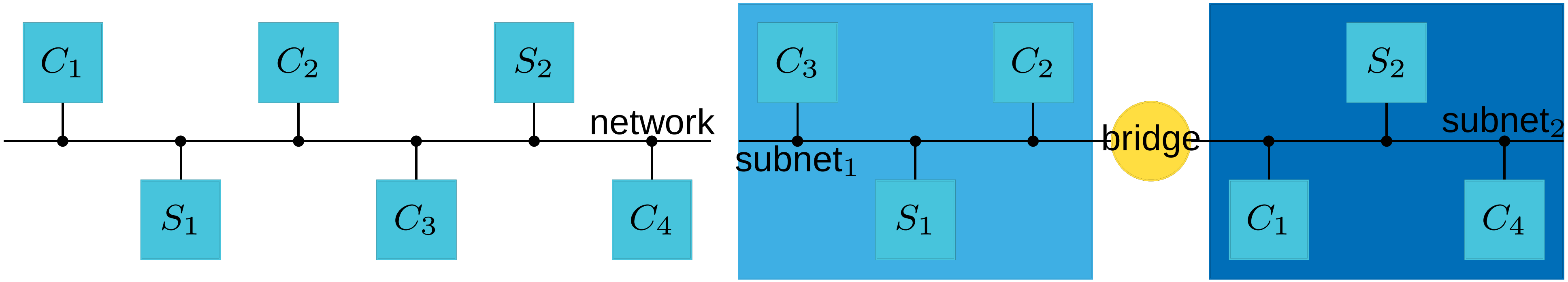

time
t_2
H_3
net 3
net 2
H_3
H_1
H_2
B_2
H_2
H_2
H_1
da
net 2
net 1
H_2
H_1
H_3
B_1
H_1
net 3
net 2
net 1

Bridge (3)



- also called **switched hub**

Worksheet: Bridges Reduce Network Load



	C_1	C_2	C_3	C_4
S_1	5	22	22	5
S_2	17	5	5	19

Total load= 5+22+22+5+17+5+5+19=100%

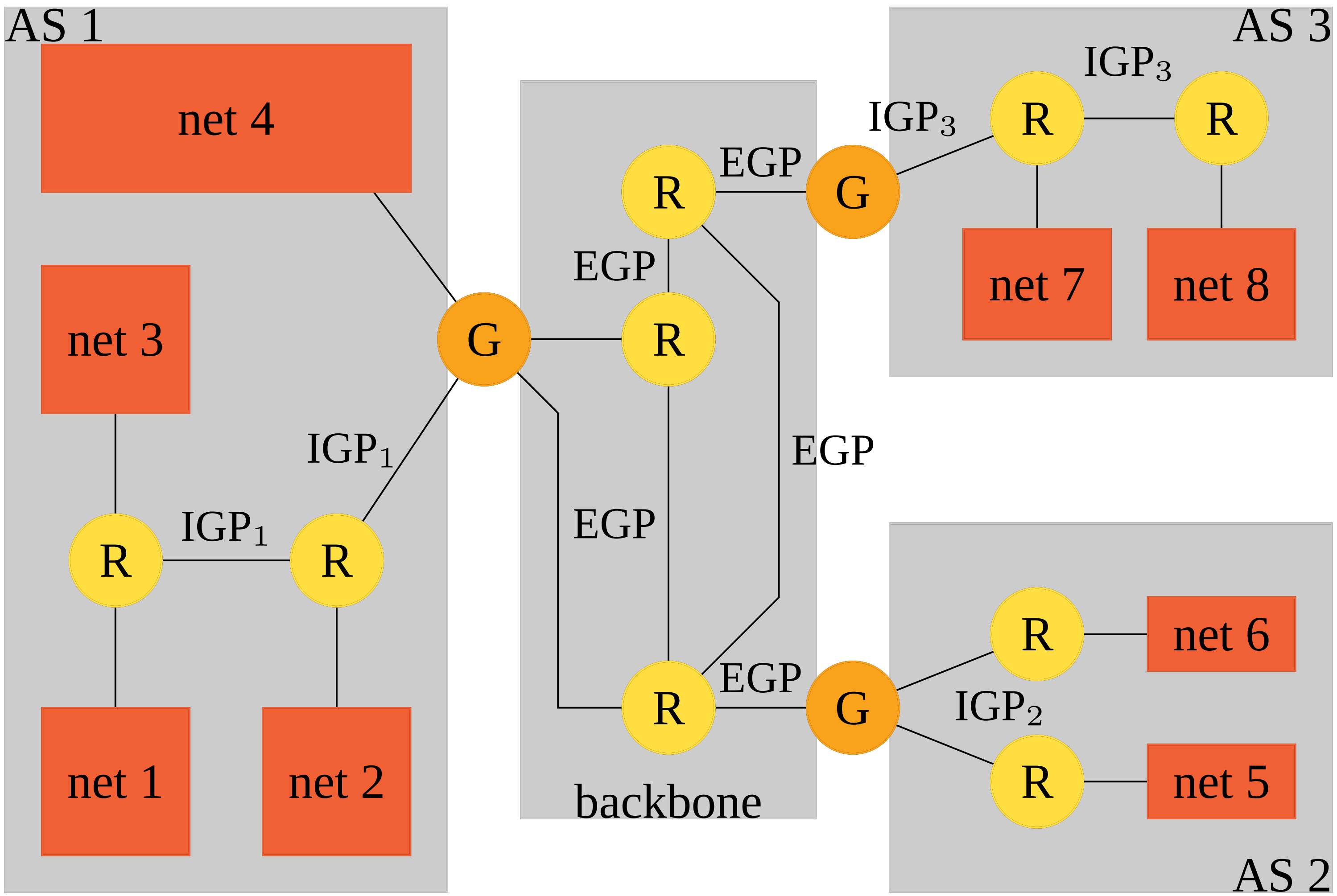
Subnet 1=Total load-Subnet 2=100-(17+19)=64%

Subnet 2=Total load-Subnet 1=100-(22+22)=56%

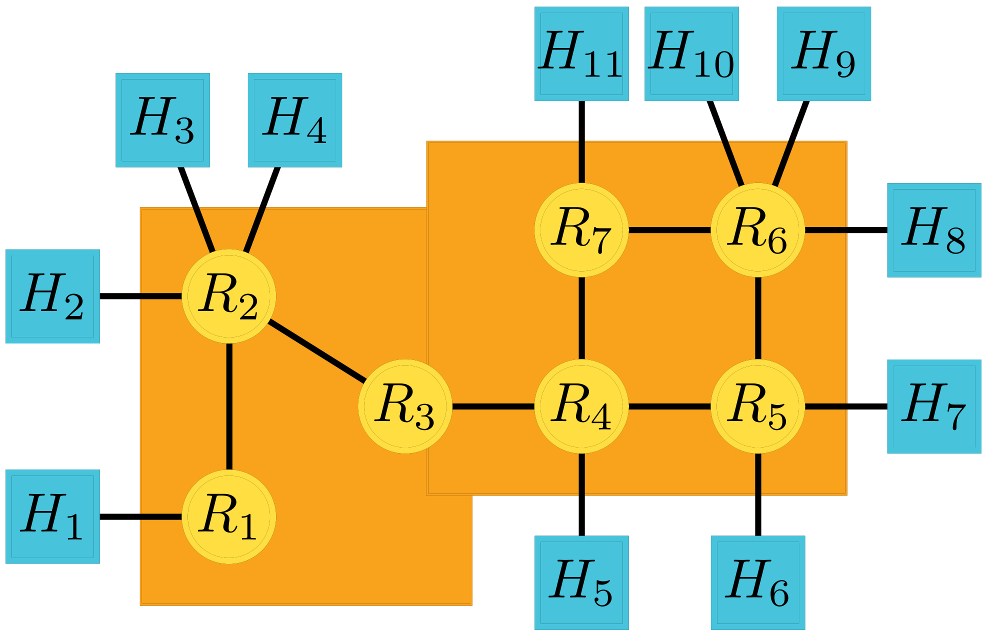
Bridges and Network Load

		To					
		H_1	H_2	H_3	H_4	H_5	H_6
From	H_1	0	300	10	0	20	0
	H_2	300	0	10	0	100	10
	H_3	10	10	0	200	5	95
	H_4	10	5	300	0	5	15
	H_5	60	815	15	5	0	5
	H_6	0	15	500	5	15	0

Using Routing Algorithms & Internetworking



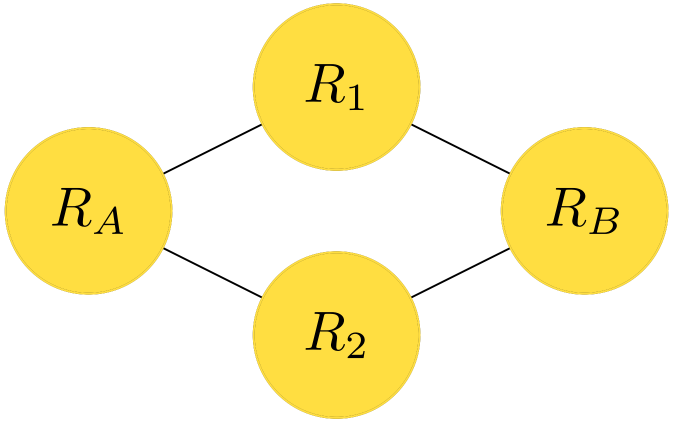
Routing Algorithm: Objectives



correctness	it finds a valid route
robustness	should some part of the network fail, it still finds a route
simplicity	must be able to analyse
stability	no ‘stampedes’ down a link
fairness	each host gets a share if network busy
optimality	route minimises the network resources used

Costing a route

Often there will be a choice of how to get from A to B

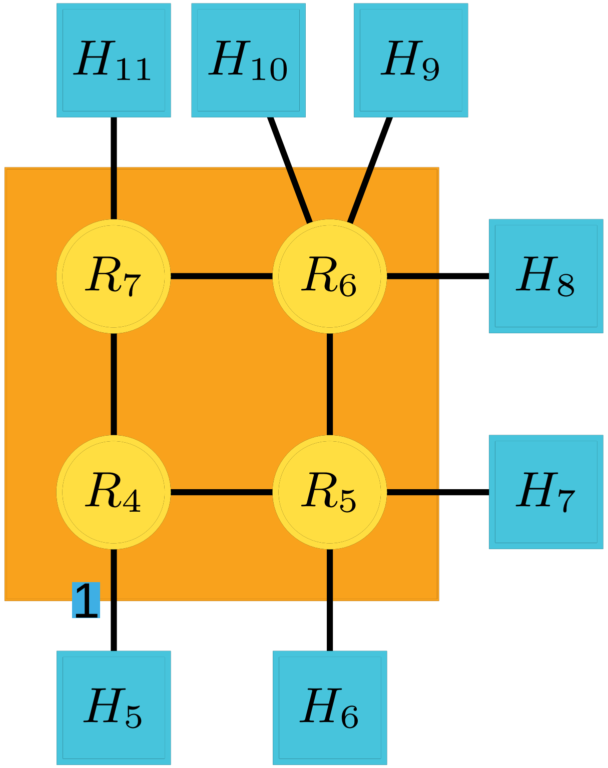


bandwidth	avoid links with less free bandwidth
delay	take links which take shorter time
price	use links with less charge per byte
priority	prefer some data over a particular link

Two Basic Approaches

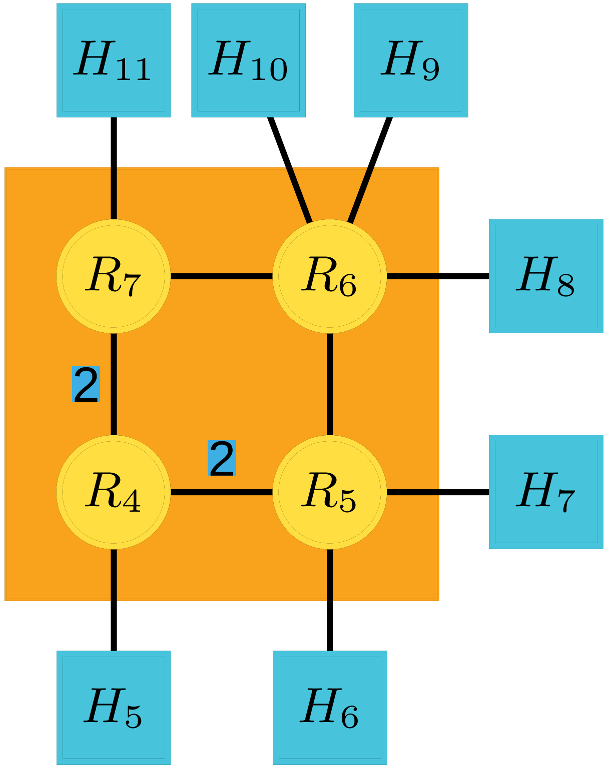
- **adaptive routing** has each router adapting its routing according to network load.
  - **isolated adaptive routing** makes decisions purely on information about the local node
  - **distributed adaptive routing** pass information between the nodes
- **nonadaptive routing**
  - routers always taking the same action to route packets
  - *aka* **static routing**

Flood Routing (1)

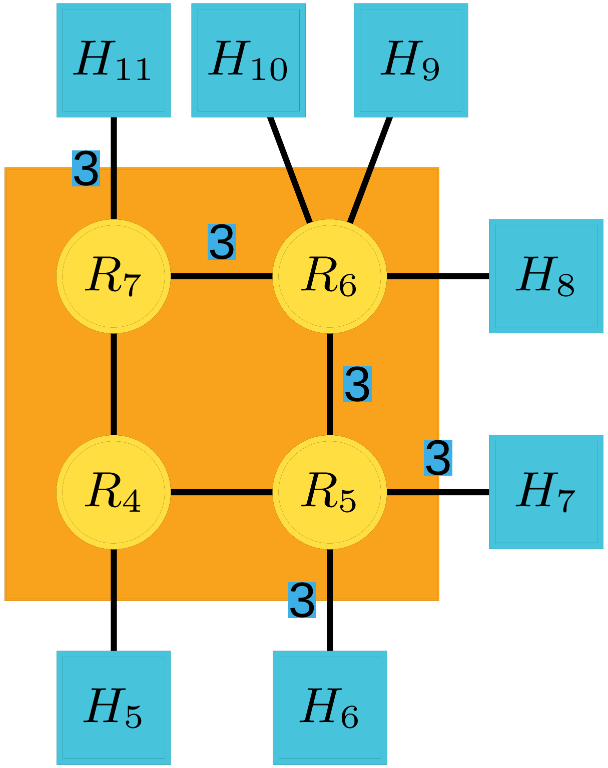


- Send the data down all links except that it arrived on

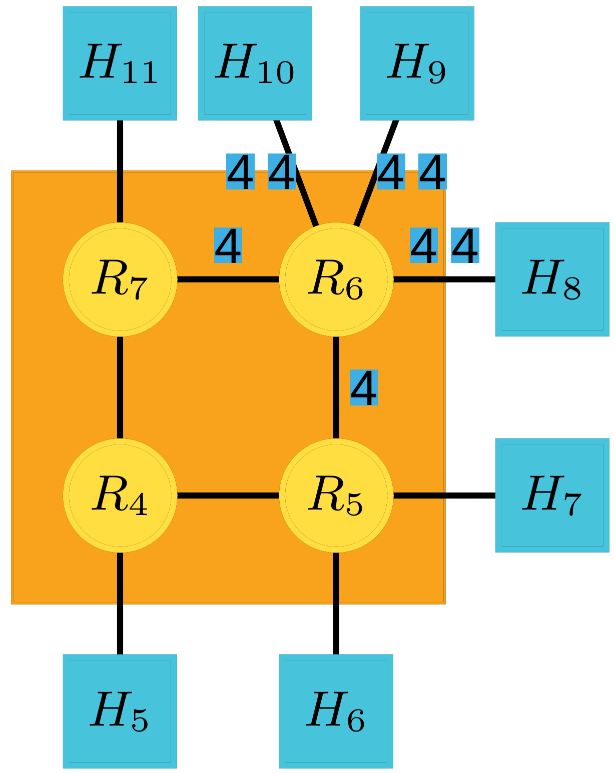
Flood Routing (2)



Flood Routing (3)

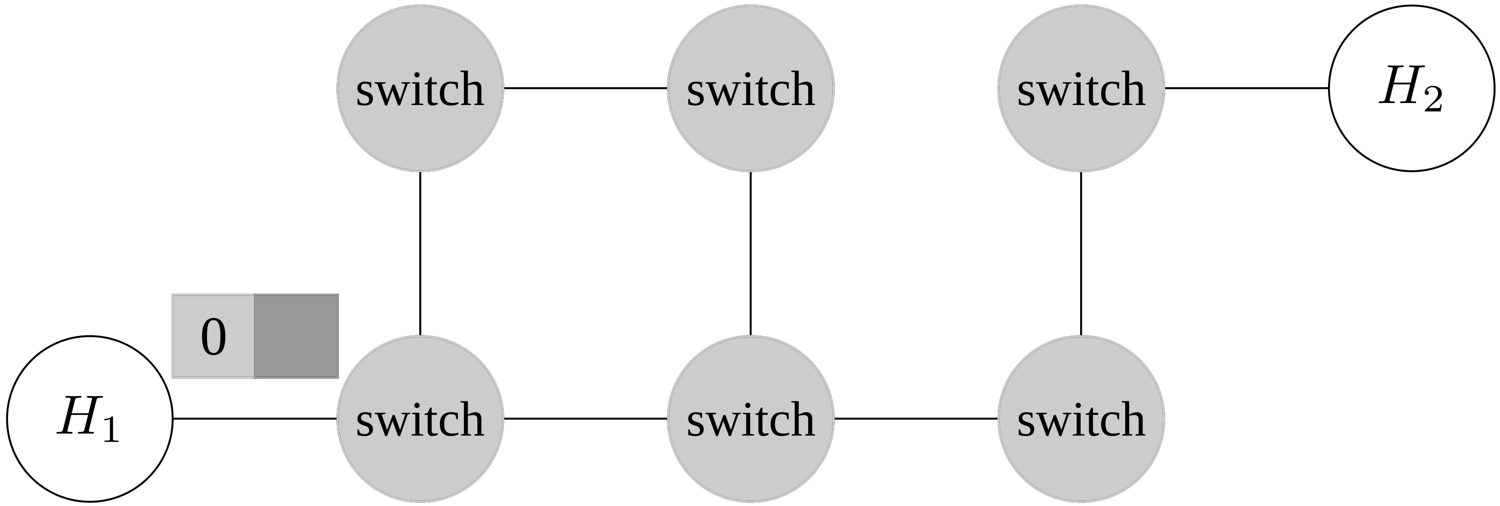


Flood Routing (4)

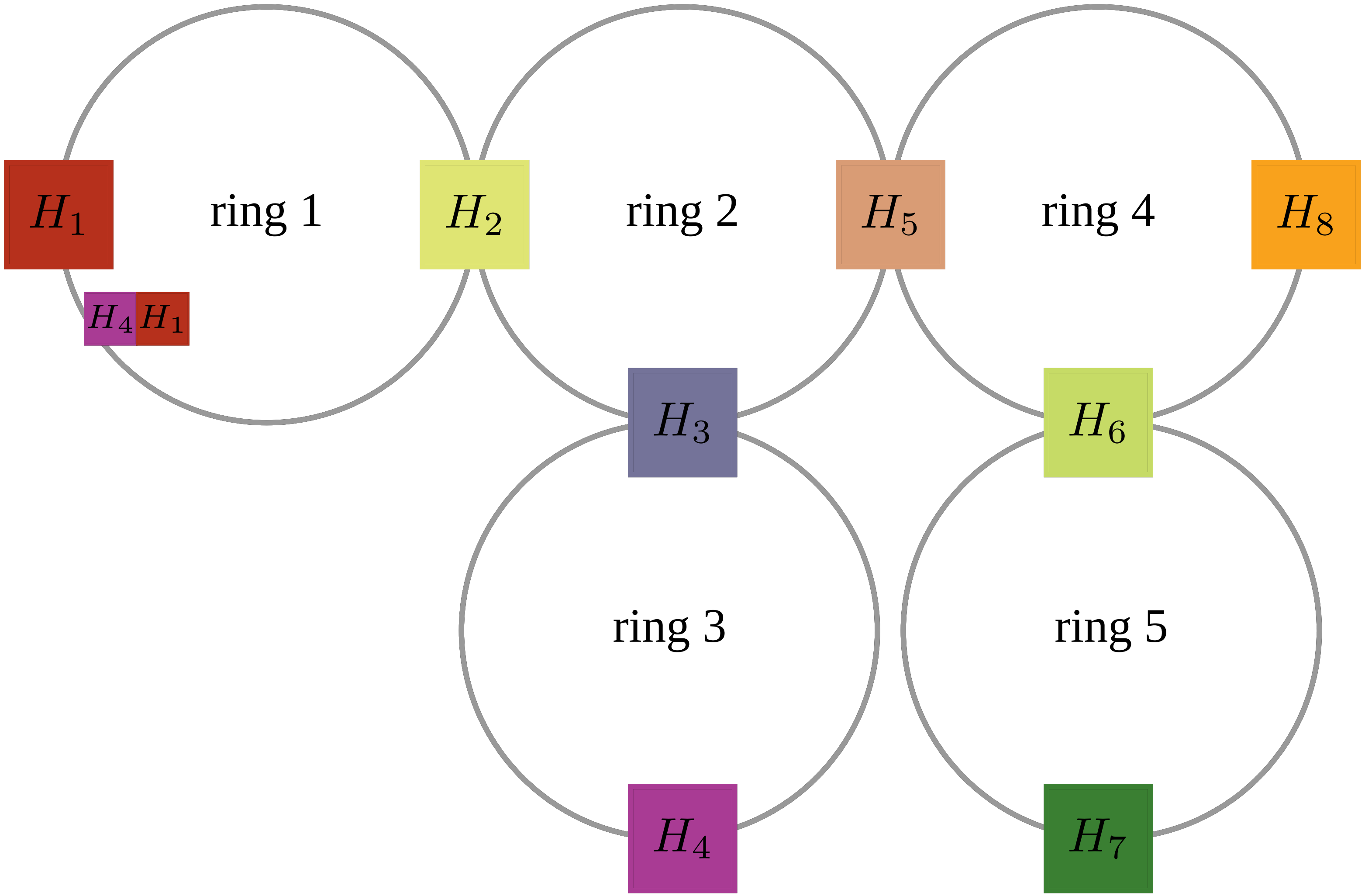


- Avoid loops by
  - recording hop count
  - recording where packet has been

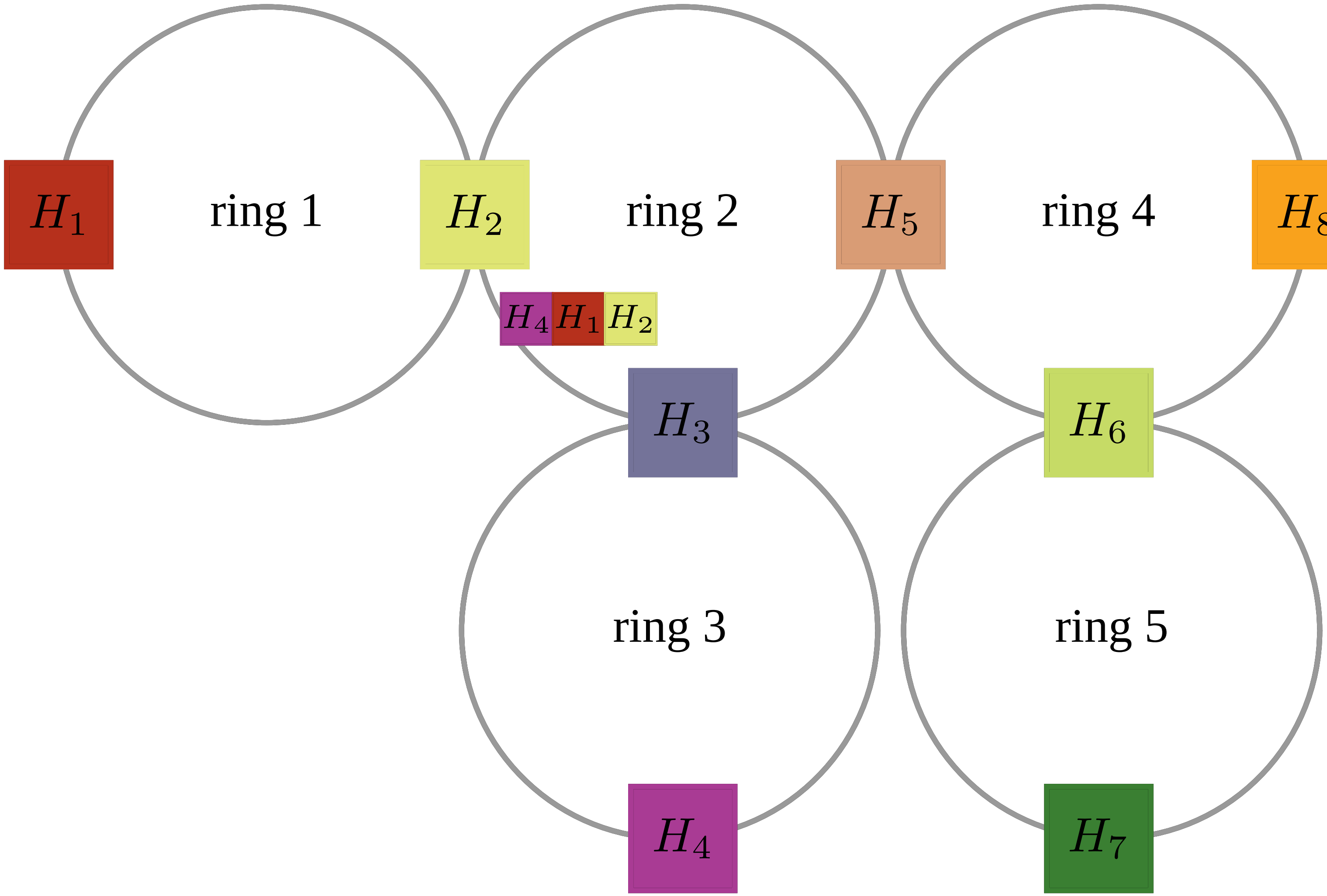
Worksheet: Flood Routing



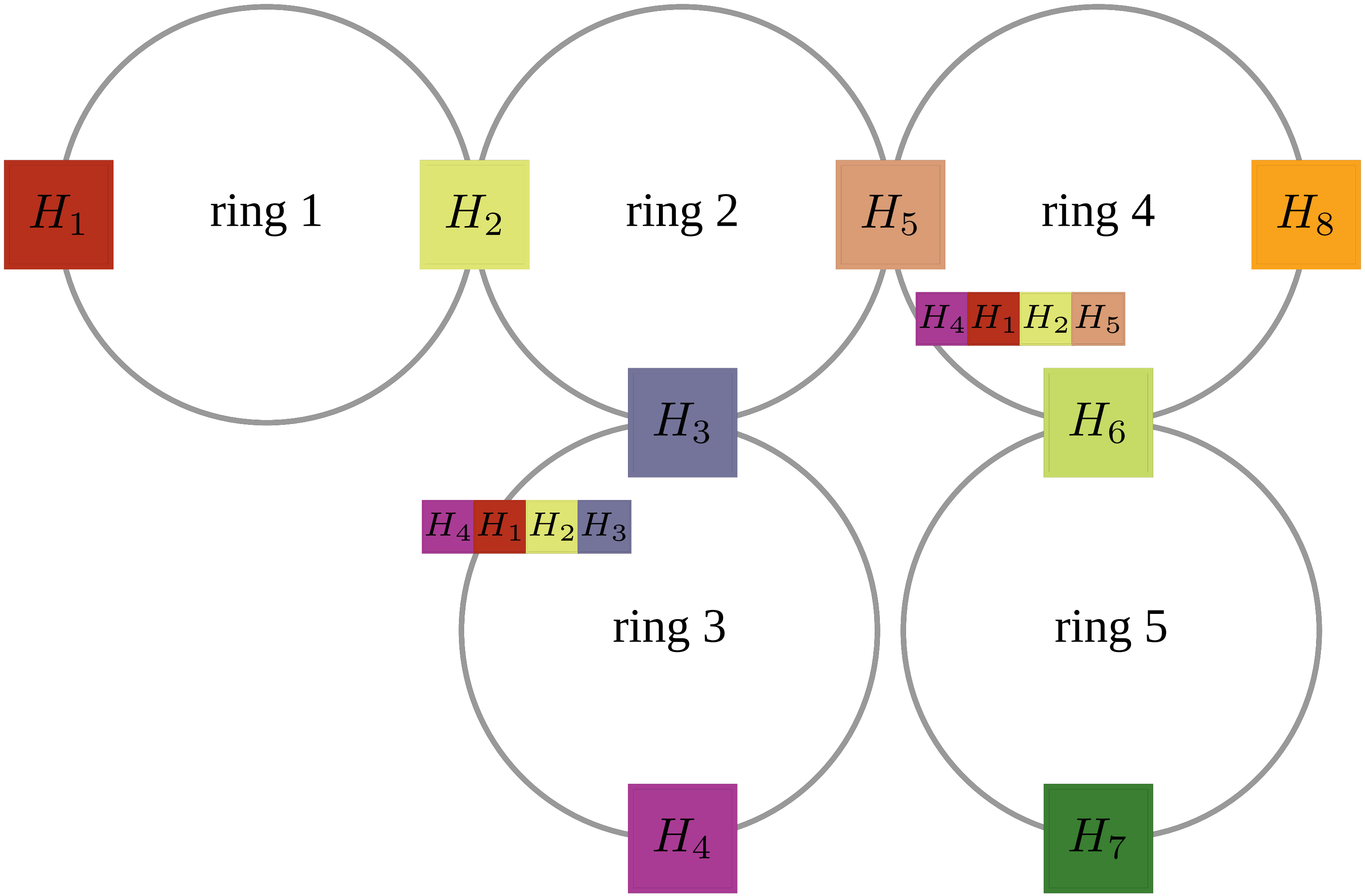
Source Route



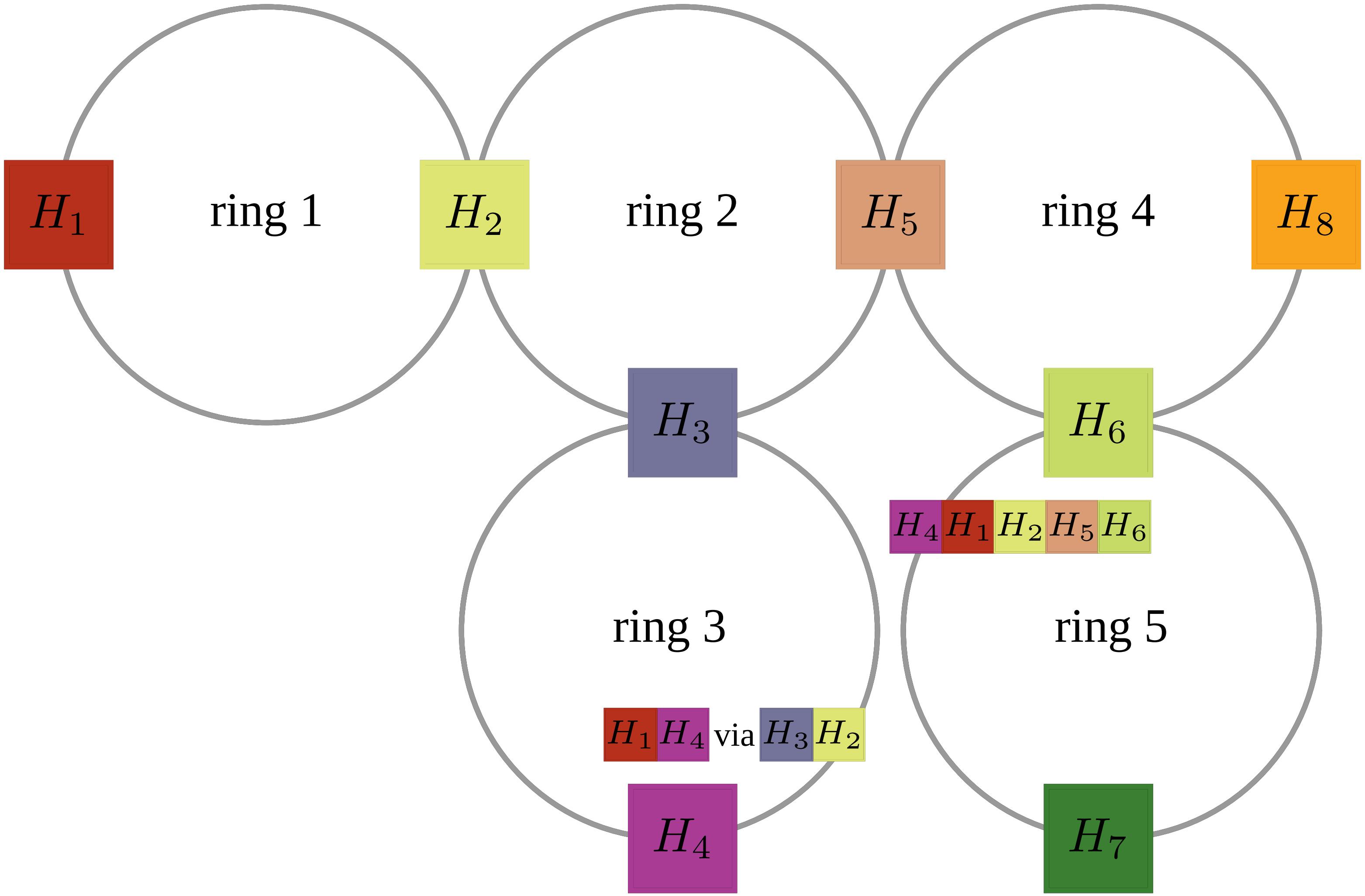
Source Route



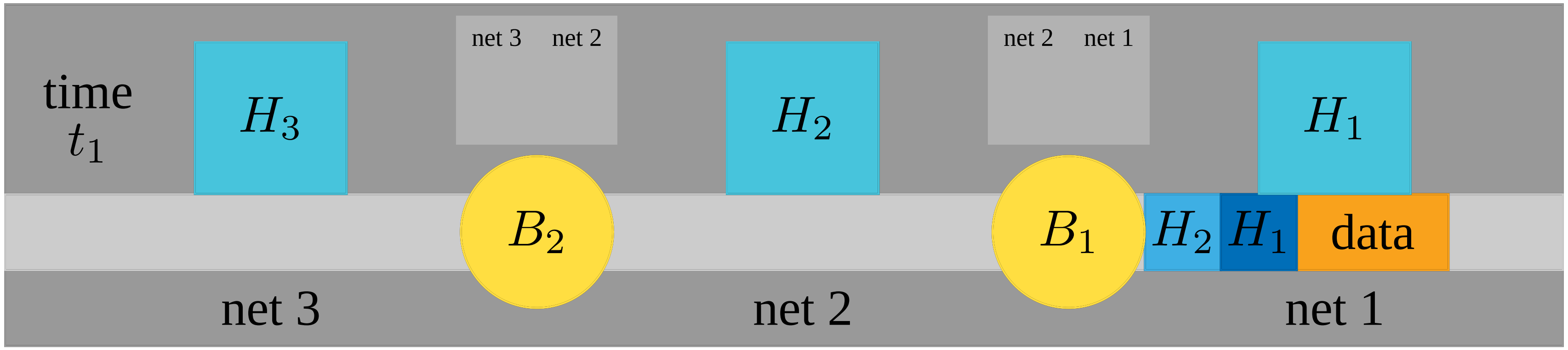
Source Route



Source Route

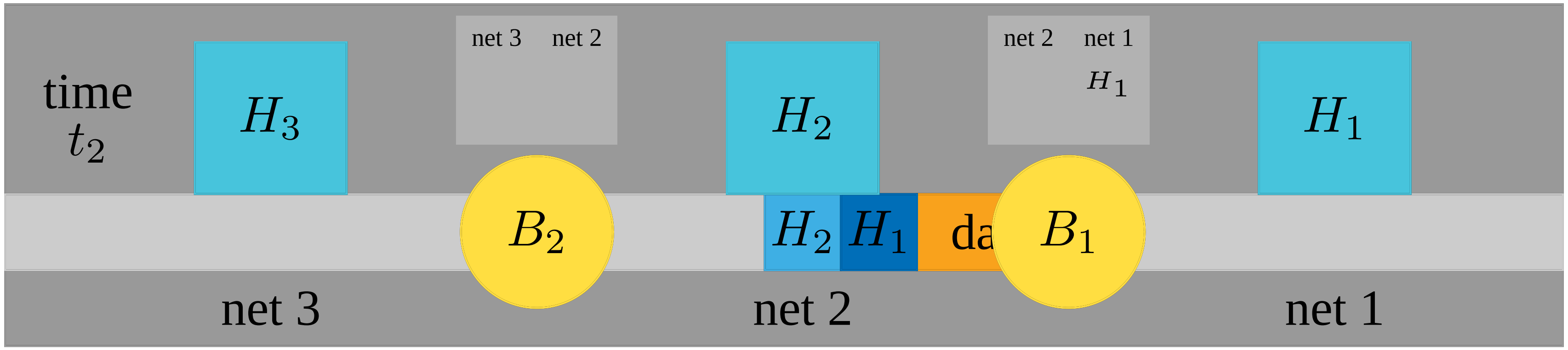


Backwards Learning and Transparent Bridges (1)



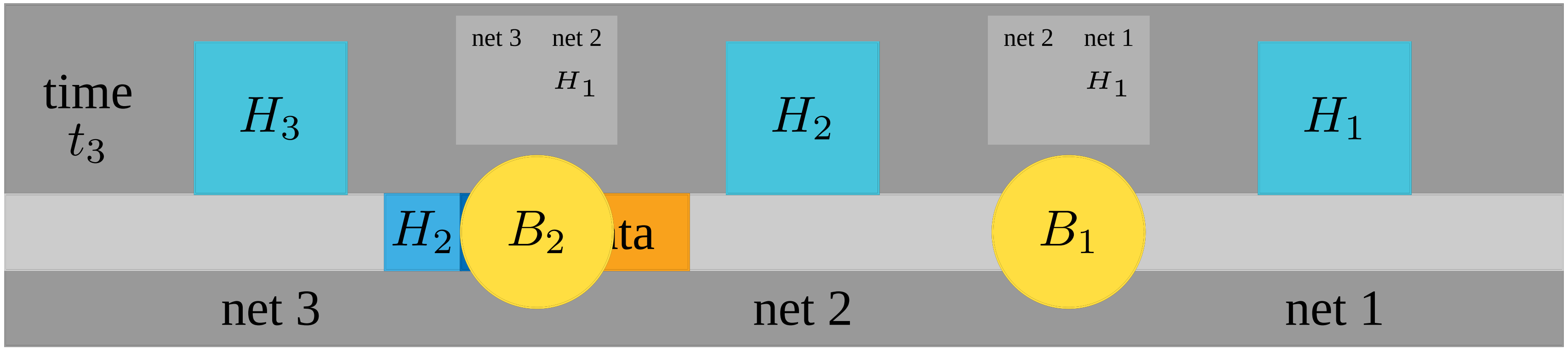
- if don’t know where host is, flood network
- bridges learn where you are

Backwards Learning and Transparent Bridges (2)



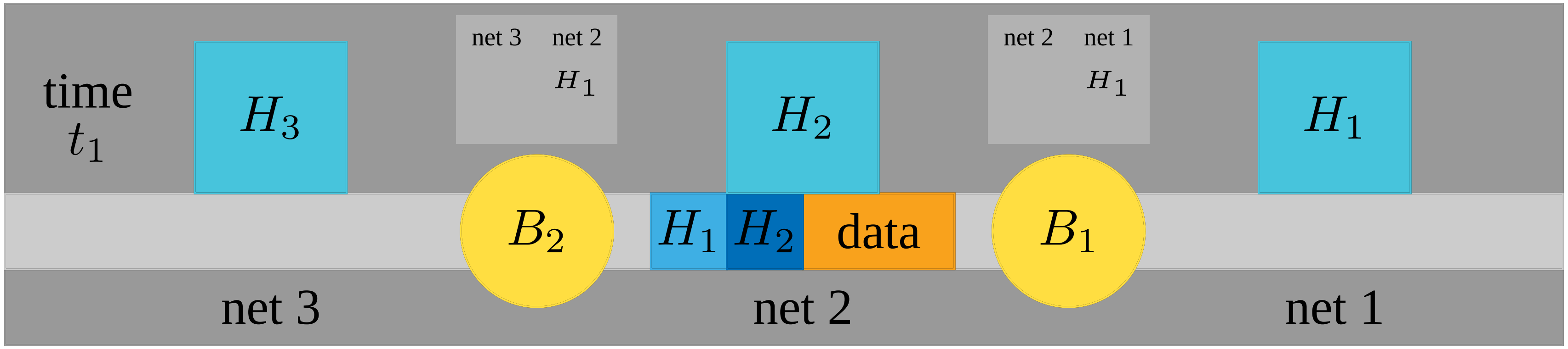
- if don’t know where host is, flood network
- bridges learn where you are

Backwards Learning and Transparent Bridges (3)



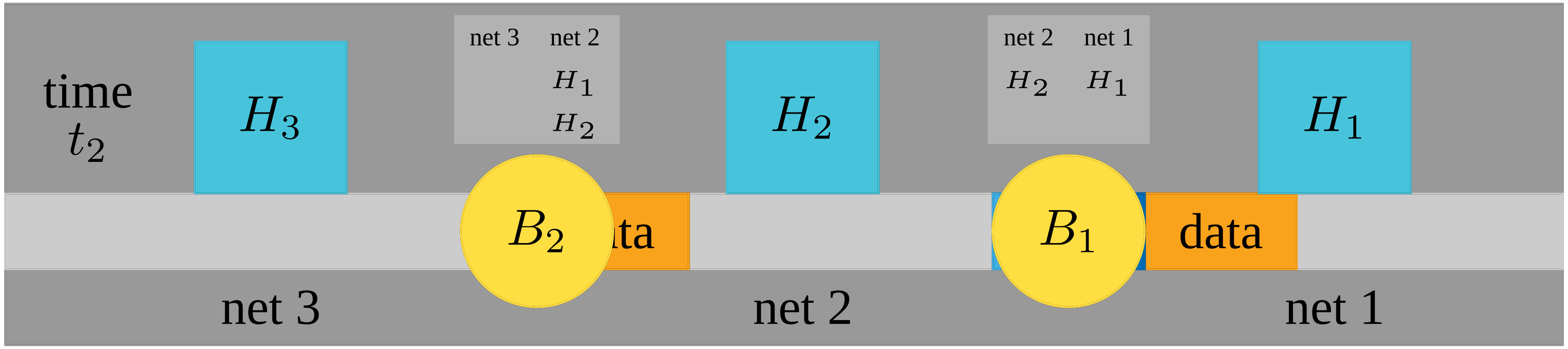
- if don’t know where host is, flood network
- bridges learn where you are

Backwards Learning and Transparent Bridges (4)



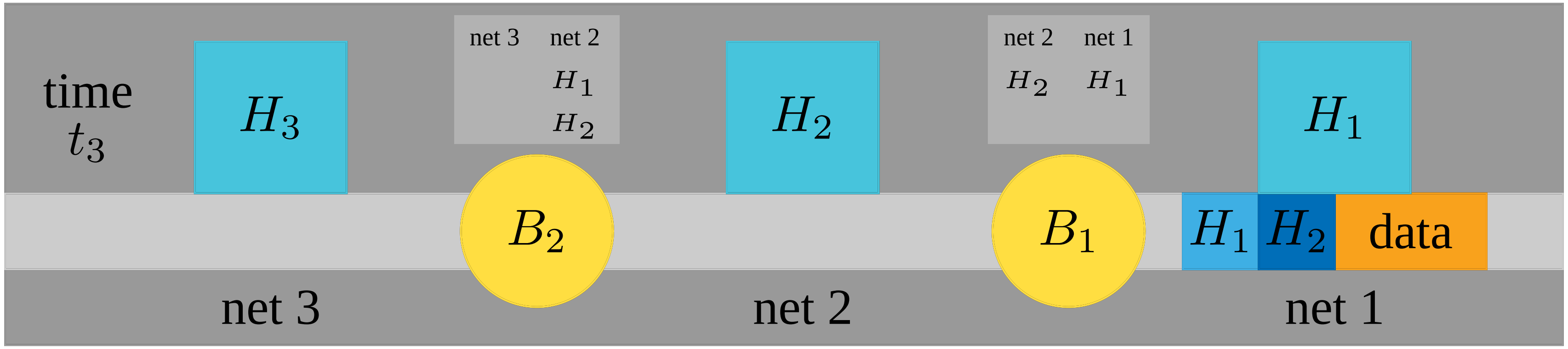
- reply teaches bridges where distination was

Backwards Learning and Transparent Bridges (5)



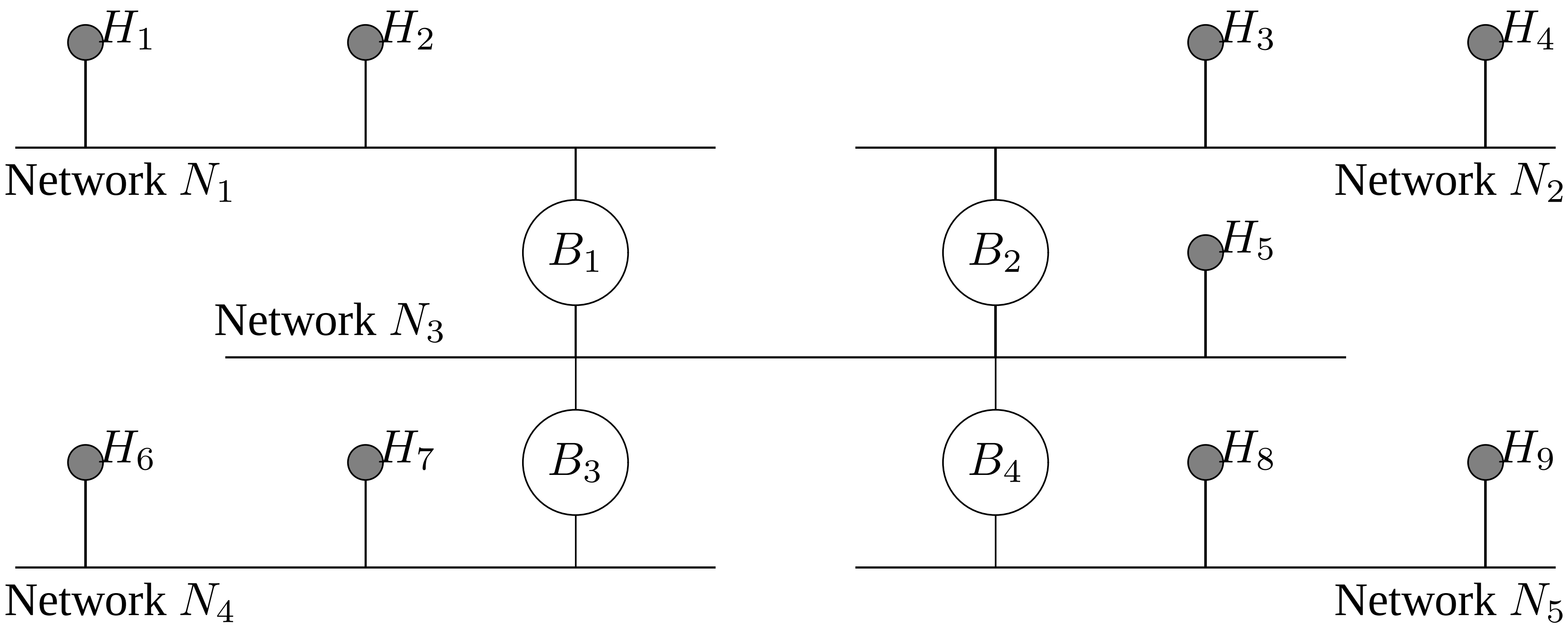
- reply teaches bridges where distination was

Backwards Learning and Transparent Bridges (6)



- reply teaches bridges where distination was

Worksheet: Backwards Learning



Host	Network
H_1	N_1

B_1

Host	Network
H_1	N_3

B_2

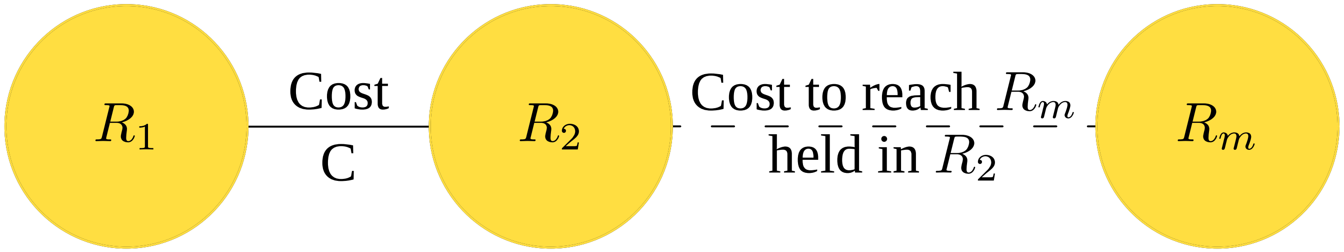
Host	Network
H_1	N_3

B_3

Host	Network
H_1	N_3

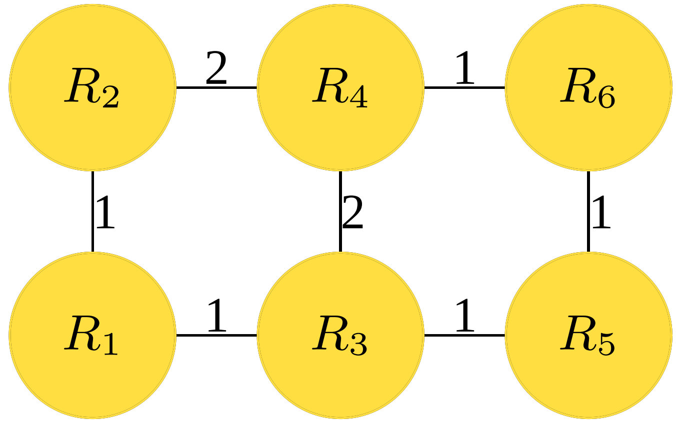
B_4

Distance Vector



- Cost C=cost to neighbour+value in neighbour's distance vector
- Recompute local distance vector as distribute to neighbours

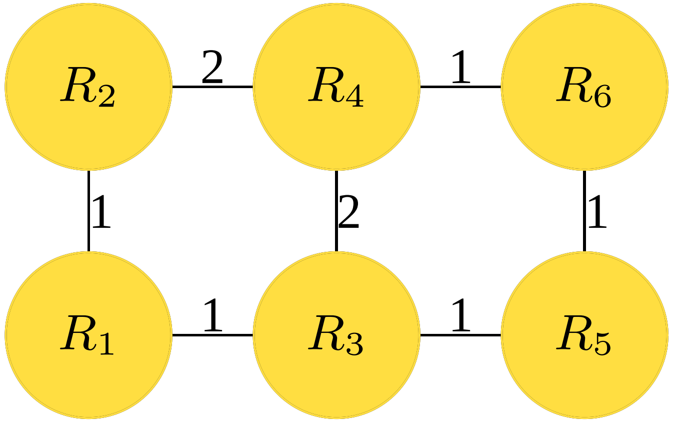
Distance Vector (1)



		To					
		R_1	R_2	R_3	R_4	R_5	R_6
	R_1	0	0	0	0	0	0
	R_2	0	0	0	0	0	0
From	R_3	0	0	0	0	0	0
	R_4	0	0	0	0	0	0
	R_5	0	0	0	0	0	0
	R_6	0	0	0	0	0	0

		To					
		R_1	R_2	R_3	R_4	R_5	R_6
	R_1	0	1	1	1	1	1
	R_2	1	0	1	1	1	1
From	R_3	1	1	0	1	1	1
	R_4	1	1	1	0	1	1
	R_5	1	1	1	1	0	1
	R_6	1	1	1	1	1	0

Distance Vector (2)

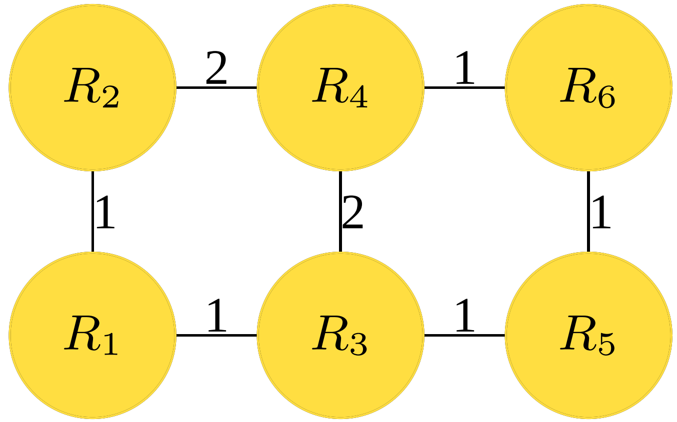


From \ To	R_1	R_2	R_3	R_4	R_5	R_6
R_1	0	1	1	1	1	1
R_2	1	0	1	1	1	1
R_3	1	1	0	1	1	1
R_4	1	1	1	0	1	1
R_5	1	1	1	1	0	1
R_6	1	1	1	1	1	0

⇒

From \ To	R_1	R_2	R_3	R_4	R_5	R_6
R_1	0	1	1	2	2	2
R_2	1	0	2	2	2	2
R_3	1	2	0	2	1	2
R_4	2	2	2	0	2	1
R_5	2	2	1	2	0	1
R_6	2	2	2	1	1	0

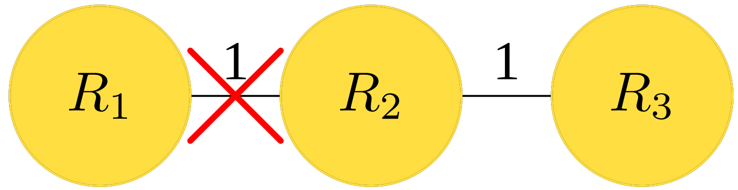
Distance Vector (3)



		To					
		R_1	R_2	R_3	R_4	R_5	R_6
	R_1	0	1	1	2	2	2
	R_2	1	0	2	2	2	2
From	R_3	1	2	0	2	1	2
	R_4	2	2	2	0	2	1
	R_5	2	2	1	2	0	1
	R_6	2	2	2	1	1	0

		To					
		R_1	R_2	R_3	R_4	R_5	R_6
	R_1	0	1	1	3	2	3
	R_2	1	0	2	2	3	3
From	R_3	1	2	0	2	1	2
	R_4	3	2	2	0	2	1
	R_5	2	3	1	2	0	1
	R_6	3	3	2	1	1	0

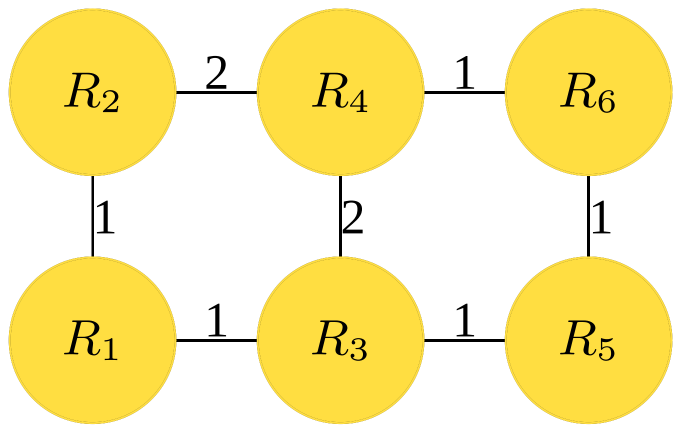
Distance Vector: Poor Response to Failures



		To		
		R_1	R_2	R_3
	R_1	0	1	2
From	R_2	1	0	1
	R_3	2	1	0

		To		
		R_1	R_2	R_3
	R_1	0	∞	∞
From	R_2	3	0	1
	R_3	2	1	0

Link State



From \ To	R_1	R_2	R_3	R_4	R_5	R_6
R_1	0	1	1	?	?	?
R_2	1	0	?	2	?	?
R_3	1	?	0	2	1	?
R_4	?	2	2	0	?	1
R_5	?	?	1	?	0	1
R_6	?	?	?	1	1	0

From \ To	R_1	R_2	R_3	R_4	R_5	R_6
R_1	0	1	1	3	2	3
R_2	1	0	2	2	3	3
R_3	1	2	0	2	1	2
R_4	3	2	2	0	2	1
R_5	2	3	1	2	0	1
R_6	3	3	2	1	1	0