5617, Spring 2019 computer networking and communication

anduo wang, Temple University
TTLMAN 401 A, R 17:30-20:00

final exam

May 2

- 17:45-19:45, in class
- close book
- cover everything discussed this semester
  - before the midterm 30%
  - after the midterm 70%

MPLS, the 2.5 layer

Tag Switching Architecture Overview

https://ieeexplore.ieee.org/document/650179/

tag switching =

- # of tags a switch maintains

a label swapping forwarding paradigm + network layer routing

- tags
  - are simple, well suited- to high-performance forwarding
  - simplify integration of routers and anachronous transfer mode switches

- tags
  - have a wide spectrum of forwarding granularity, enabling diverse routing functionalities
    - multicast
    - more flexible routing
    - scale routing with hierachy

tag switching supports a high-quality, scalable routing system

tag switching =

forwarding component

- uses tag information (tags) carried by the packets and the tag forwarding information base (***TFIB***) maintained by a tag switch to perform packet forwarding

control component

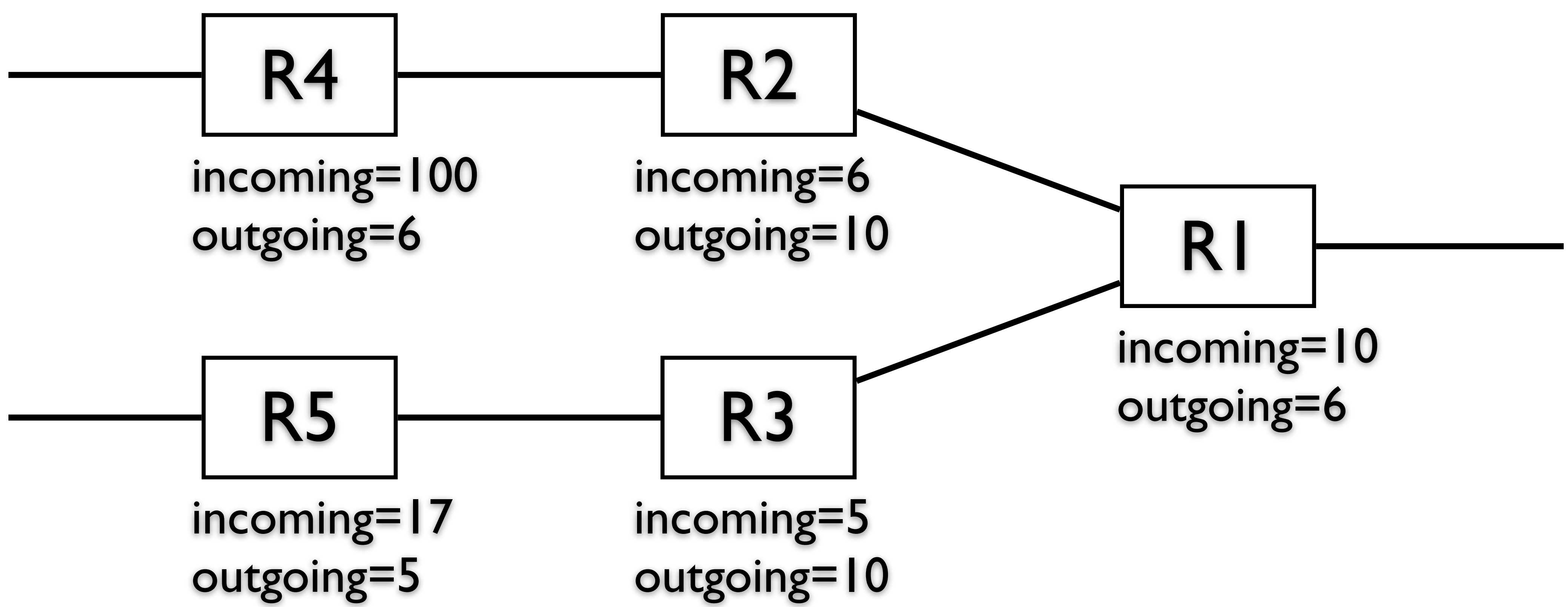
- various routing modules
  - each provides a particular set of control functionalities
- maintain correct TFIBs among a group of interconnected tag switches

forwarding component

forwarding — label swapping

a tag switch uses the *tag as an index* in its TFIB

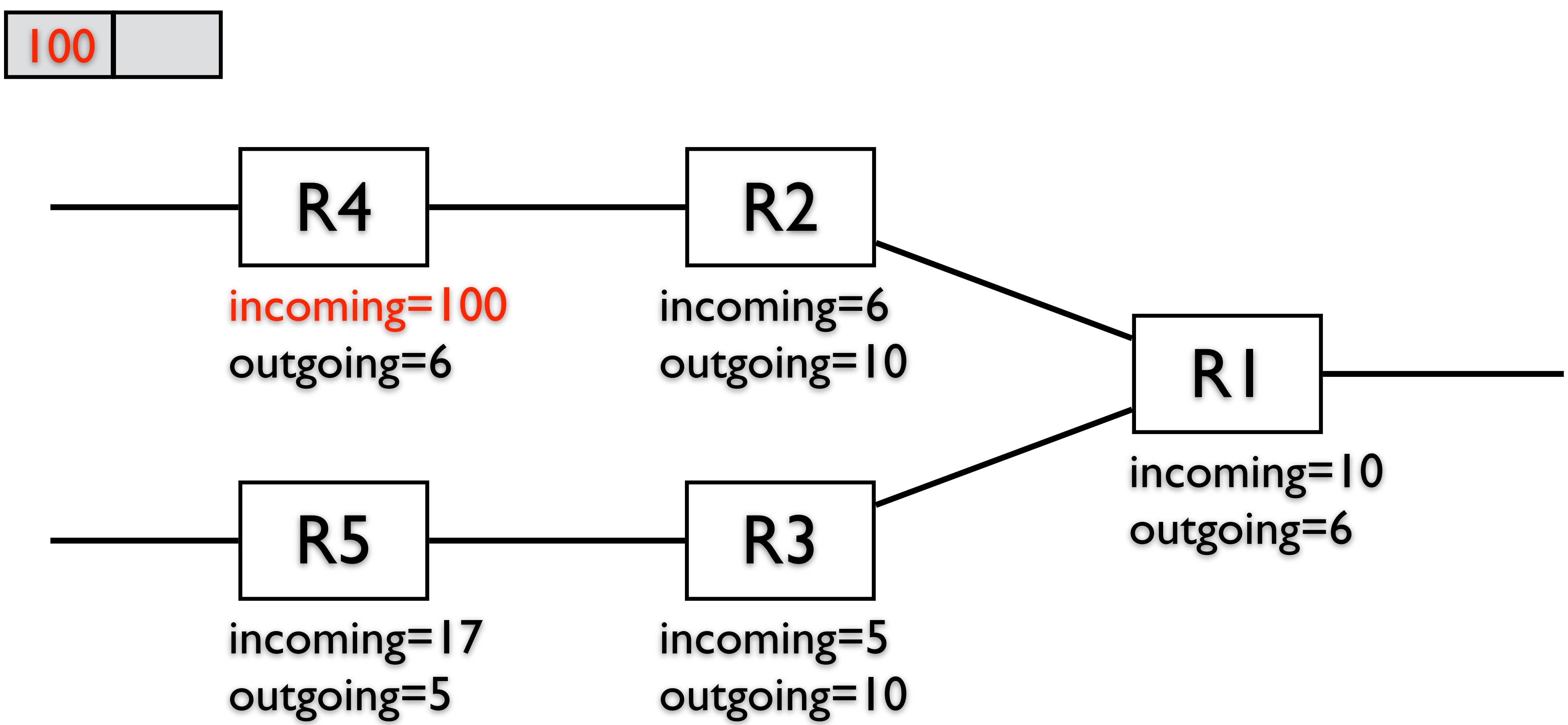
- <incoming tag, outgoing tag, outgoing interface …>



forwarding — label swapping

a tag switch uses the *tag as an index* in its TFIB

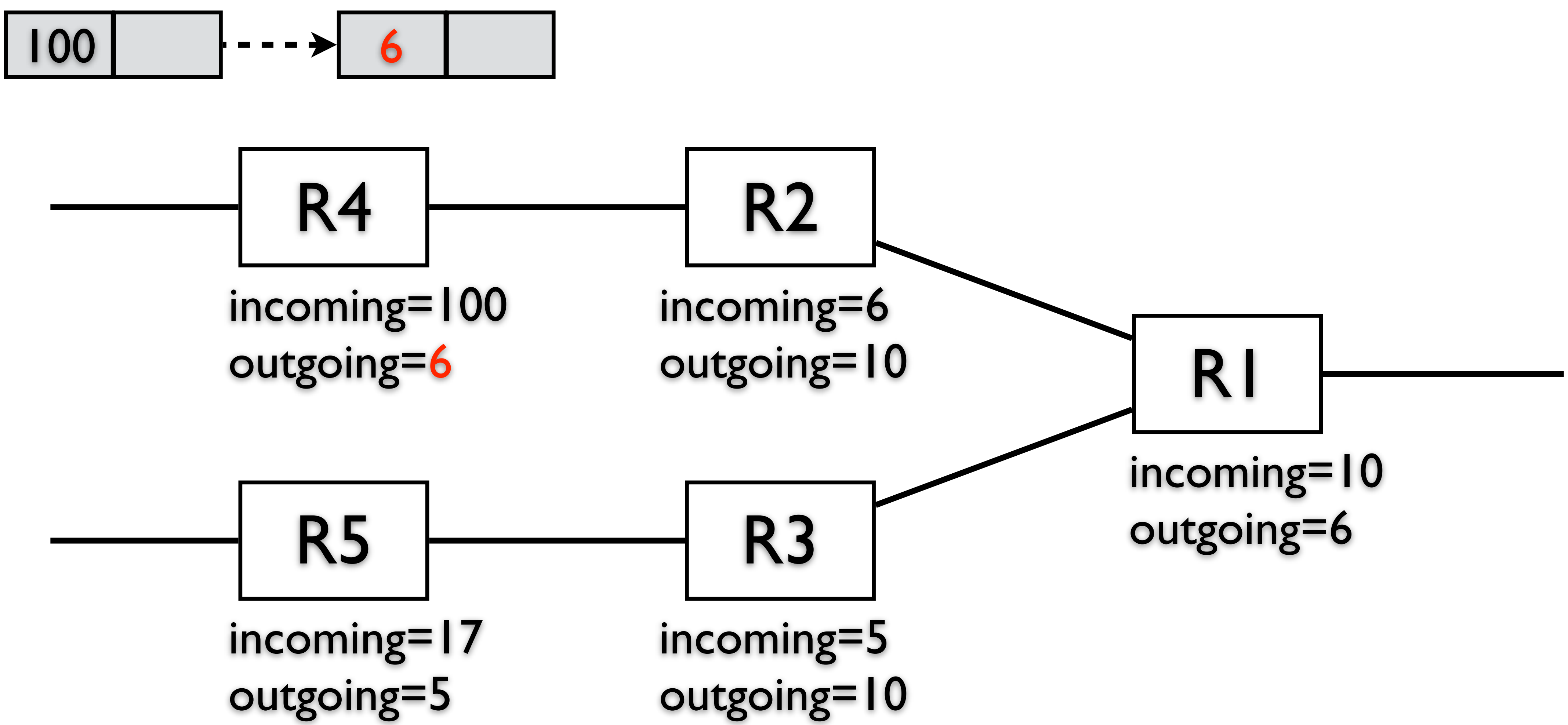
- <incoming tag, outgoing tag, outgoing interface …>



forwarding — label swapping

replaces the tag with the outgoing tag

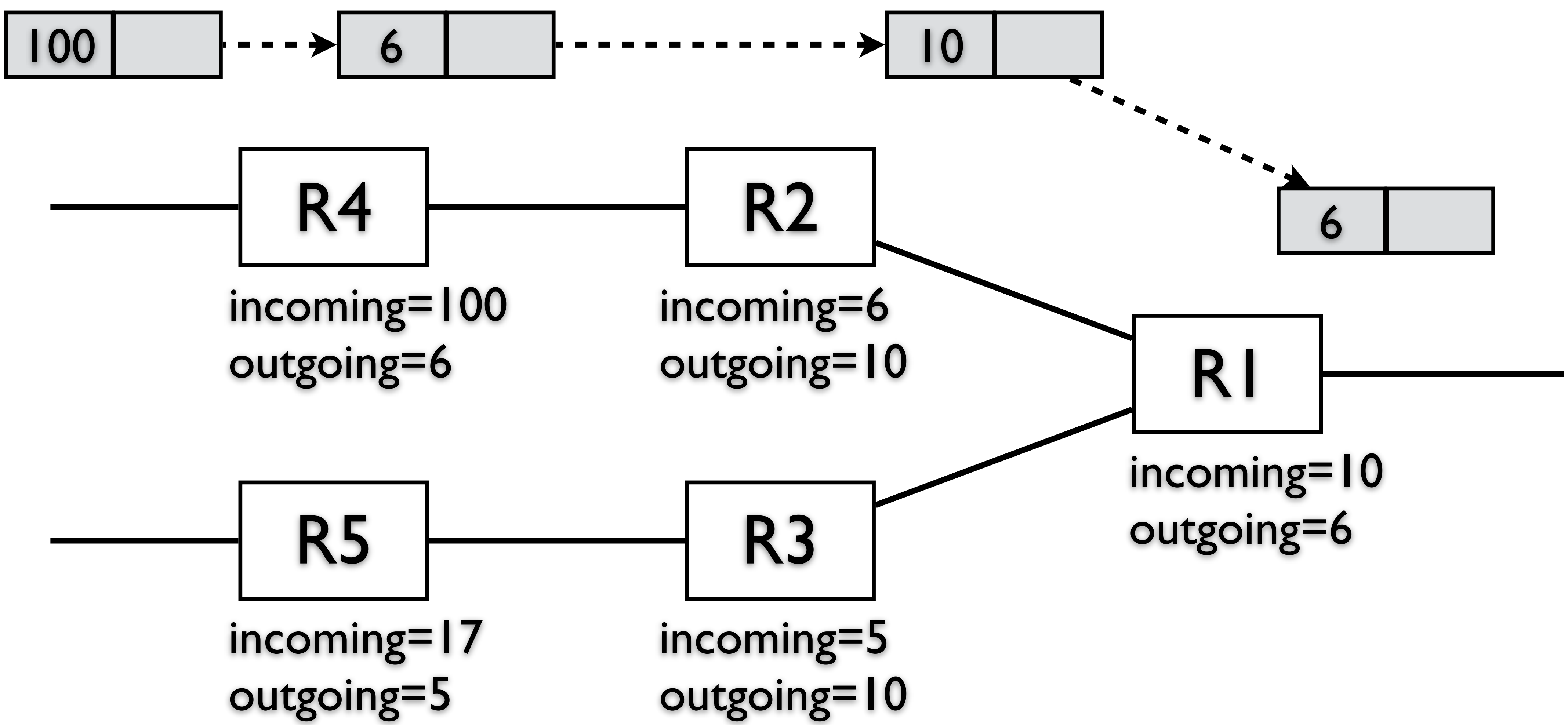
- <incoming tag, outgoing tag, outgoing interface ...>



forwarding — label swapping

replaces the tag with the outgoing tag

- <incoming tag, outgoing tag, outgoing interface …>



high forwarding performance

label swapping enables high performance

- exact match algorithm using fixed length (20 bit)
- fairly short tag as an index

high forwarding performance

label swapping enables high performance

- exact match algorithm using fixed length (20 bit)
- fairly short tag as an index

compare: longest prefix match

high forwarding performance

label swapping enables high performance

- exact match algorithm using fixed length (20 bit)
- fairly short tag as an index

} compare: longest prefix match

simple enough to allow straightforward hardware implementation

decoupled from the control component

label swapping is independent of tag's forwarding behavior

- same forwarding paradigm for unicast/multicast
  - unicast: a unicast entry has a single <outgoing tag …> subentry
  - multicast: … one or more sub-entries

label swapping is independent of network-layer

- same forwarding paradigm that supports a variety of network-layer protocols

decoupled from the control component

label swapping is independent of tag's forwarding behavior

- same forwarding paradigm for unicast/multicast
  - unicast: a unicast entry has a single <outgoing tag …> subentry
  - multicast: … one or more sub-entries

label swapping is independent of network-layer

- same forwarding paradigm that supports a variety of network-layer protocols

new routing (control) functions can be added without disturbing the forwarding paradigm (or re-optimization)

control component

tag binding

binding between a tag and network-layer route

- create a tag binding
  - allocating a tag, binding it to a route
- distribute the tag binding information among tag switches

tag binding

binding between a tag and network-layer route

- create a tag binding
  - allocating a tag, binding it to a route
- distribute the tag binding information among tag switches

distribution and maintenance of tag binding information is consistent with that of the associated routing information

- unicast: like OSPF
- multicast: periodic refresh

tag binding examples

different tag binding scheme realizes different control functionalities

- destination-based routing
- flexible route (explicit routes)
- hierarchy of routing knowledge (BGP)

destination-based routing

a switch allocates tags and binds them to address prefixes in its FIB

- downstream allocation
  - the tag carried in a packet is generated and bound to a prefix by the switch at the downstream end of a link

destination-based routing

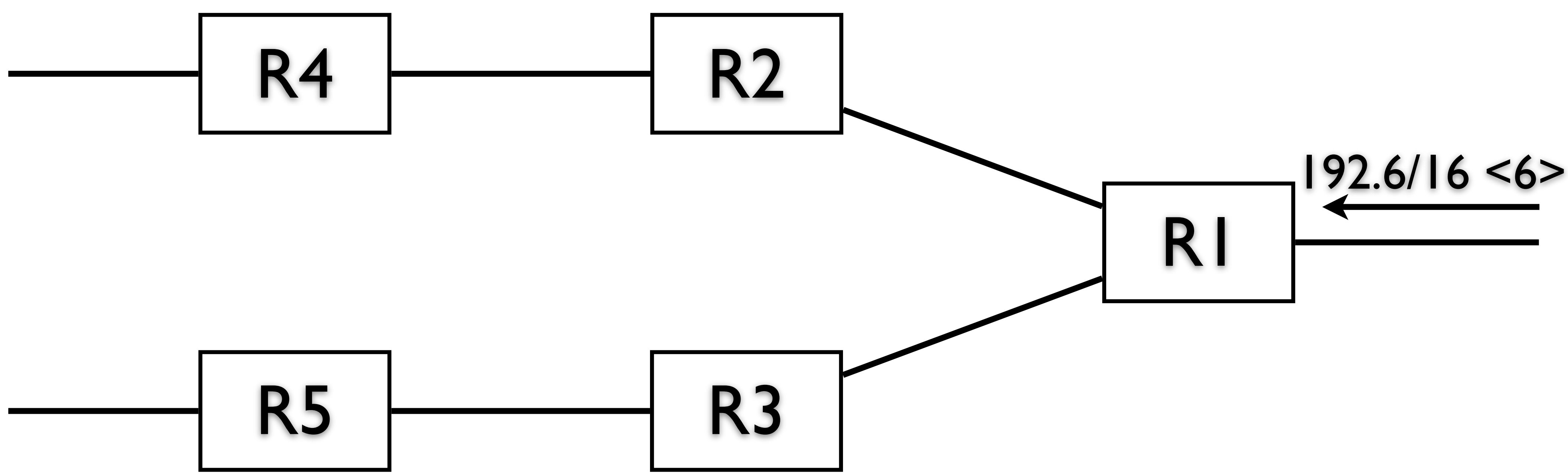
downstream allocation

- the tag carried in a packet is generated and bound to a prefix by the switch at the downstream end of a link
- for each route in the (downstream) switch's FIB
  - allocates a (incoming) tag
  - creates an entry in its TFIB
  - advertises the binding between the (incoming) tag and the route to the (upstream) other adjacent switches

destination-based routing

downstream allocation

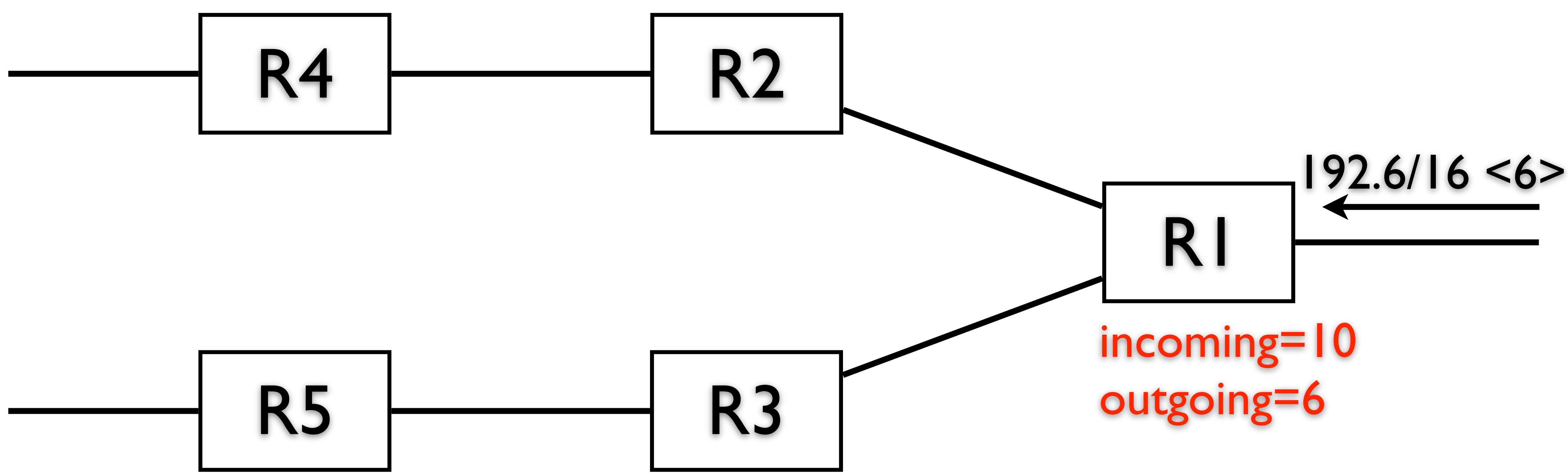
- R1 receives 192.6/16 bound to tag <6>



destination-based routing

R1 receives 192.6/16 with tag <6>

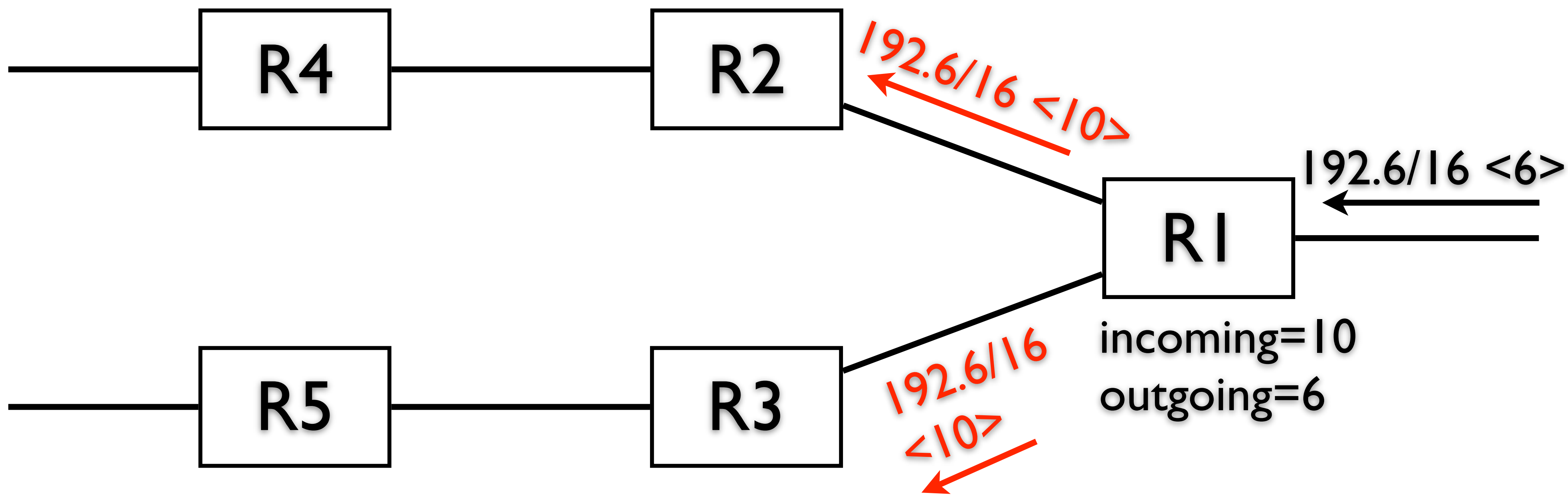
- creates an entry in TFIB, set outgoing tag to <6>
- generates a local tag <10>, set incoming tag to <10>



destination-based routing

R1 receives 192.6/16 with tag <6>

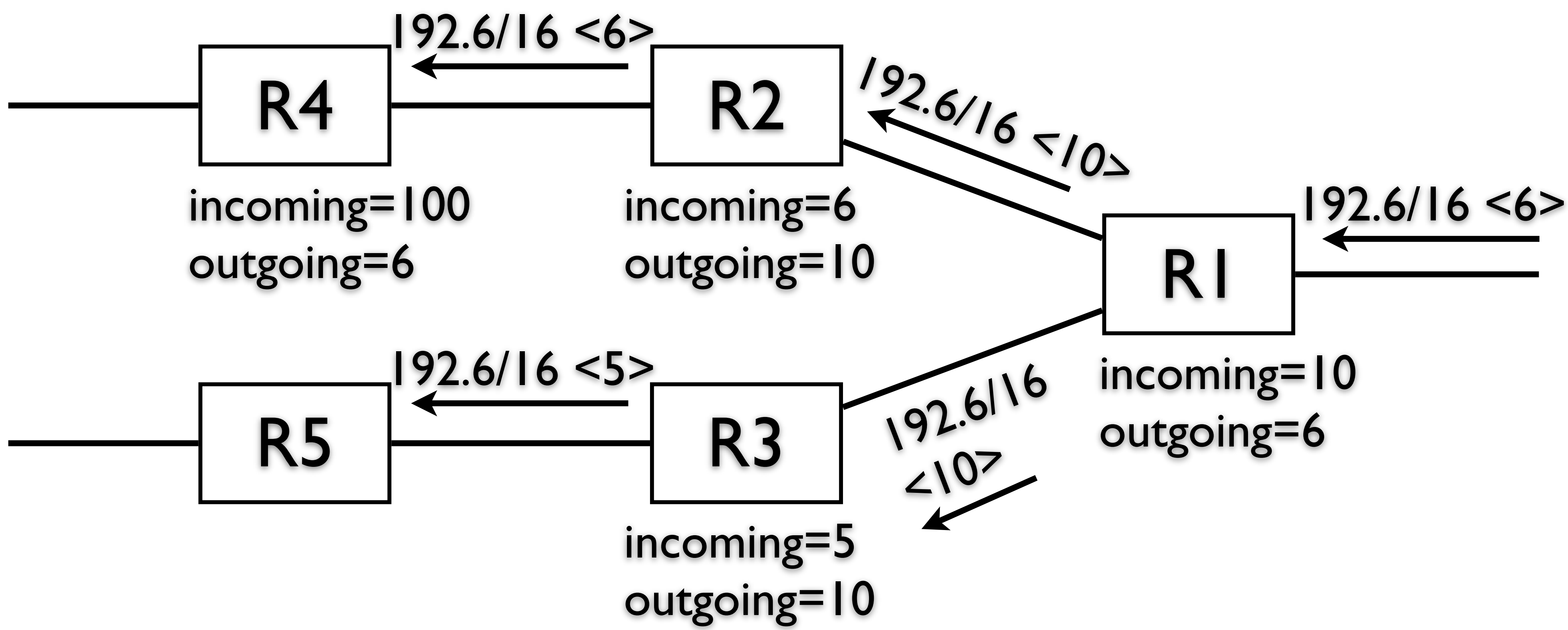
- set outgoing tag to <6>, set incoming tag to <10>
- advertises 192.6/16 with <10> to others



destination-based routing

similarly, R2, R3, R4

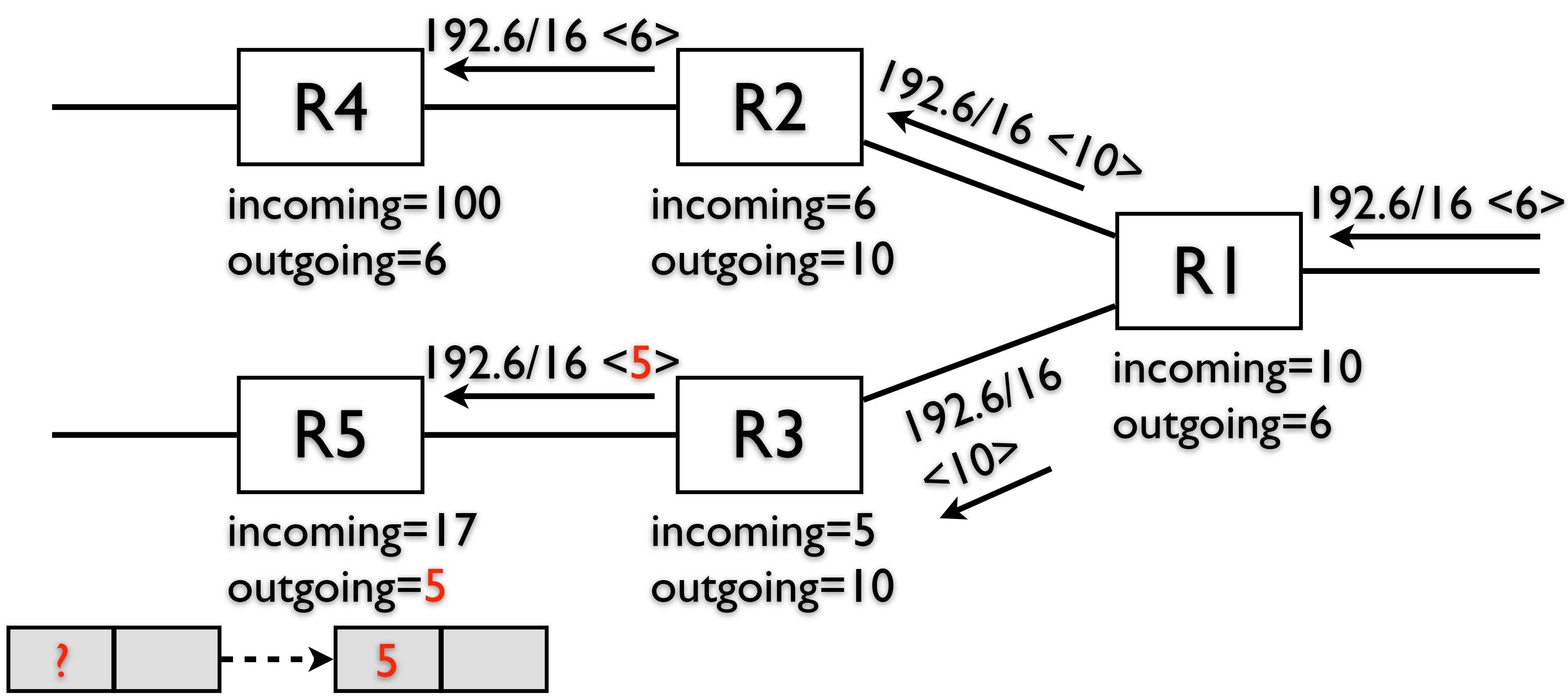
- receive tag binding, create TFIB entries, re-advertise



destination-based routing

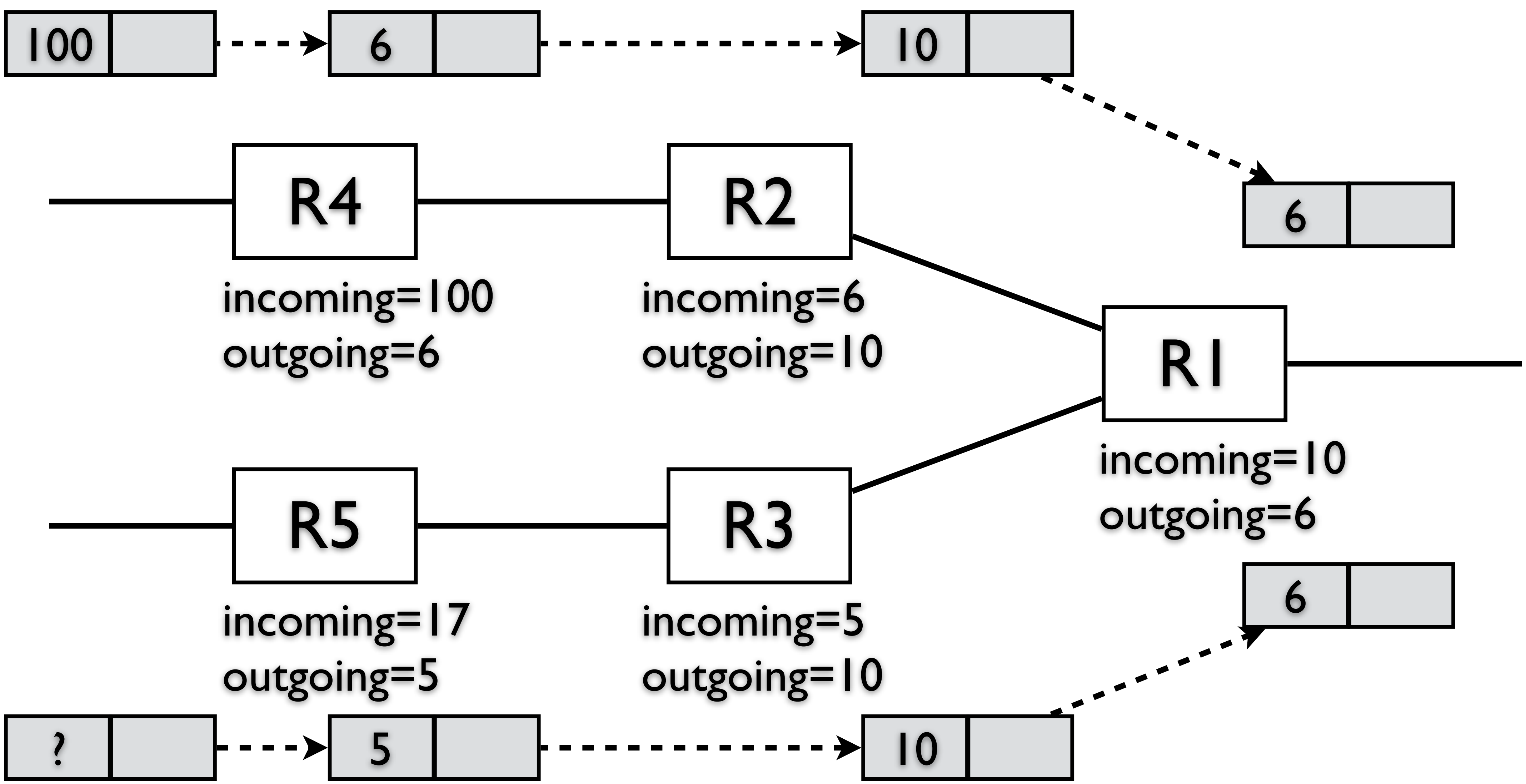
R5, router left to which is not a tag switch

- R5 also augments its FIB with outgoing tag <5>



destination-based routing

a switch allocates tags and binds them to address prefixes in its FIB



observation — routes aggregation

tag allocation is topology-driven

- if a tag switch forwards multiple packets to the same next-hop neighbor
  - only a single (incoming) tag is needed
- if a tag switch receives a set of routes associated with a single tag
  - only a single (incoming) tag is needed

scaling properties

tag switching used for destination-based routing

of tags a switch maintains

# of routes in the FIB

scaling properties

tag switching used for destination-based routing

of tags a switch maintains << # of routes in the FIB

scaling properties

tag switching used for destination-based routing

of tags a switch maintains   <<   # of routes in the FIB

tag associated with routes, rather than flows

- much less state required
- no need to perform flow classicification

scaling properties

tag switching used for destination-based routing

of tags a switch maintains << # of routes in the FIB

tag associated with routes, rather than flows

- much less state required
- no need to perform flow classicification

more robust & stable destination-based routing in the presence of traffic pattern change

observation — normal destination-based forwarding still needed

when a tag is added to a previously untagged packet

- first hop router requires normal FIB forwarding

when a tag switch aggregates a set of routes into a single tag, but the routes do not share a common next hop

- again, look up the normal FIB

flexible routing (explicit routes)

provides forwarding along the paths different from the path determined by destination-based routing

- install tag binding in tag switches that do not correspond to the destination based routing paths

hierarchical routing (BGP)

Internet routing (BGP)

- 2-tier routing scheme, collection of routing domains

tag switching

- decouples interior (intra-) and exterior (inter-) routing
- significantly reduces load on non-border switches
- only border maintains routing information for both interior/ exterior routing

hierarchical routing (BGP)

tag stack

- a set of tags carried by a packet organized as a stack

operations

- label swapping as before: swap tag at the top

hierarchical routing (BGP)

tag stack

- a set of tags carried by a packet organized as a stack

operations

- label swapping as before: swap tag at the top
- pop the stack
- push one more tag into the stack

hierarchical routing (BGP)

when a packet is forwarded between two border tag switches in different domains

- the tag stack only has one tag, associated with the AS-level route

hierarchical routing (BGP)

when a packet is forwarded between two border tag switches in different domains

- the tag stack only has one tag, associated with the AS-level route

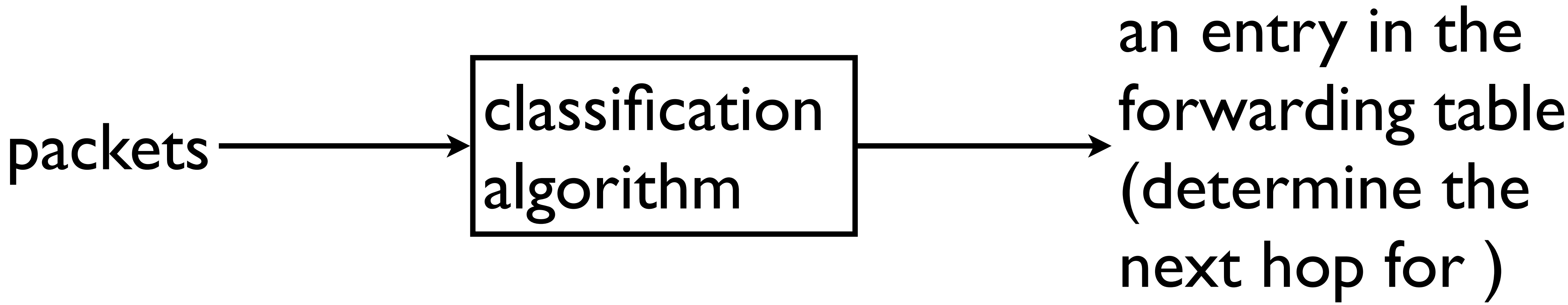
when a packet is forwarded within a domain

- ingress router: 2nd tag associated with an interior route to the egress border is pushed
- internal switches: only operate on the 2nd top tag
- egress border: pop the top (2nd) tag, uses the original tag for tag switching to routers in another domain

tag switching and forwarding equivalence classes (*FEC*)

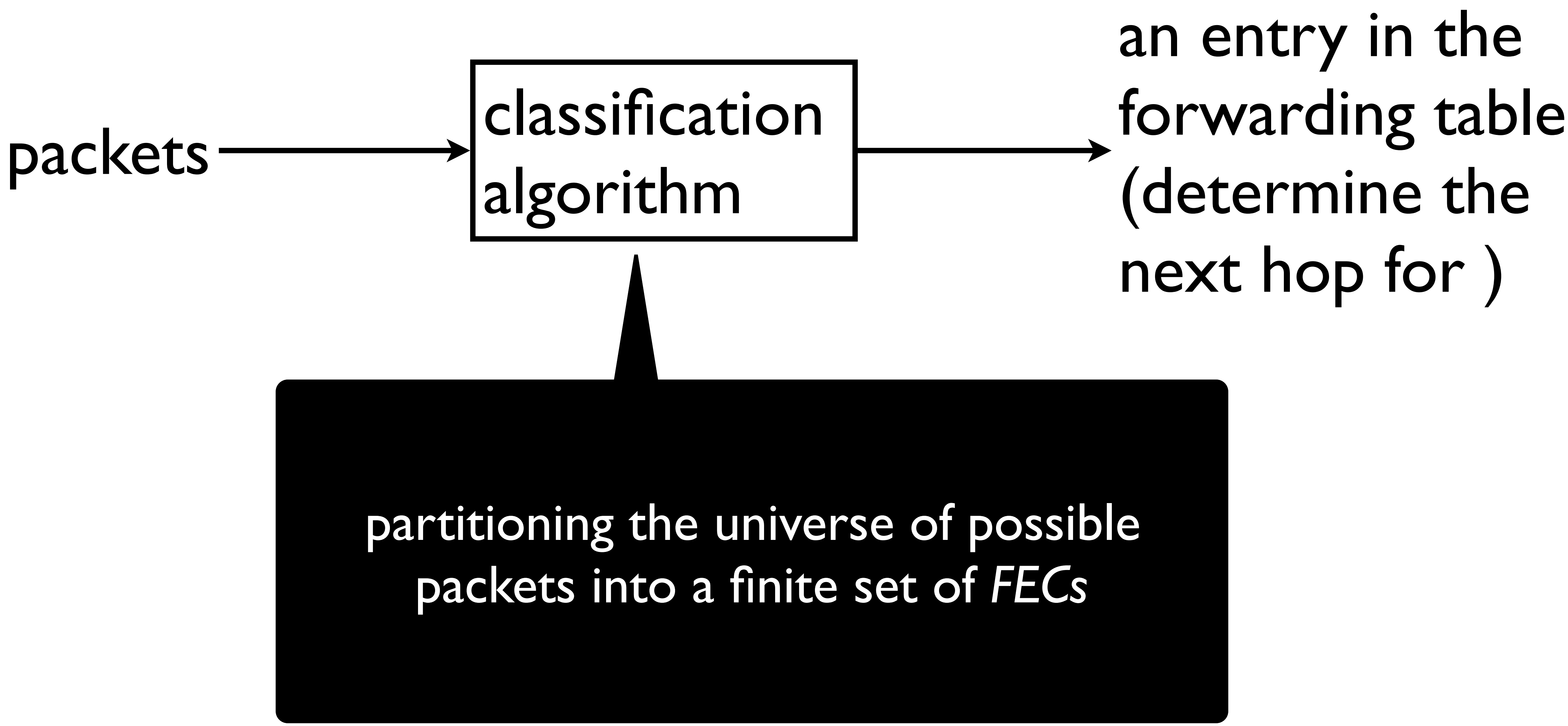
forwarding equivalence classes (*FEC*)

forwarding table in a conventional router



forwarding equivalence classes (*FEC*)

forwarding table in a conventional router



tag switching and FEC

if a pair of tag switches are adjacent, they must agree on assignments of tags to FEC

- only the first tag switch on a tag switched path needs to perform the classification algorithm

tag switching and FEC

control functionalities revisited

- 2 packets in the same FEC if they have the same prefix in the routing table that is the longest match

  destination based routing

- 2 packets in the same FEC if they are alike in some arbitrary manner by a policy

  flexible routing (explicit routes)

- 2 packets in the same FEC if they have to traverse through a common tag switch

  BGP

the power of tag switching, revisited

any number of different kind of FEC's (control schemes) can co-exist in a single switch

- as long as the result partitions the packet space seen by the tag switch

different procedures can be used by different tag switches to classify packets

a hierarchy of tags can be used

- hierarchical routing

migration strategies

inherently incrementally deployable

- tag switching performed between a pair of adjacent switches
- tag binding information distributed on a pairwise basis

transparent to legacy routers

- tag switch runs the same routing protocol, no impact on existing routers

incentive

- as more tag switches introduced — routers upgraded to enable tag switching, the scope of tag switching functionalities widens
  - e.g., internal BGP routers -> hierarchical tag switching

Fabric: A Retrospective on Evolving SDN

http://yuba.stanford.edu/~casado/fabric.pdf

many proposals towards a better network

MPLS

- simplifies hardware + improves control flexibility

SDN attempts to make further progress but suffers certain shortcomings

- can we overcome those shortcomings by adopting the insights underlying MPLS?

an ideal network

hardware

- simple (inexpensive)
- vendor-neutral
- future proof: accommodate future innovation as much as possible

control

- flexible: meet future requirements as they arise

review

original Internet, MPLS, SDN along two dimensions

- requirements
- interfaces

requirements

two sources

- hosts
- operators

hosts

- want their packets to travel to a particular destination with some QoS requirement about the nature of the services these packets receive en-route to the destination

operators

- TE, tunneling, virtualization, isolation, …

interfaces

places where control information pass between network entities

- host-network
  - *how hosts inform the network of their requirements*
  - e.g., packet header (destination address), …

interfaces

places where control information pass between network entities

- host-network
  - *how hosts inform the network of their requirements*
  - e.g., packet header (destination address), …
- operator-network
  - how operator informs the network of their requirements
  - e.g., per-box configuration command
- packet-switch
  - how a packet identifies itself to a switch
  - e.g., packet header as an index into the forwarding table

Original Internet VS. *MPLS* VS. *SDN*

	host-network interface	operator-network interface	packet-switch interface
original Internet	destination address	none	destination address
MPLS	packet header (inspected by edge tag switch)	none	label (used by internal tag switch)
SDN	packet header (Openflow)	fully programmatic interface (network abstractions)	packet header (Openflow)

shortcomings of SDN

not fulfill the promise of simple hardware

- Openflow far complex than the tens of bits MPLS

host generality expected to increas

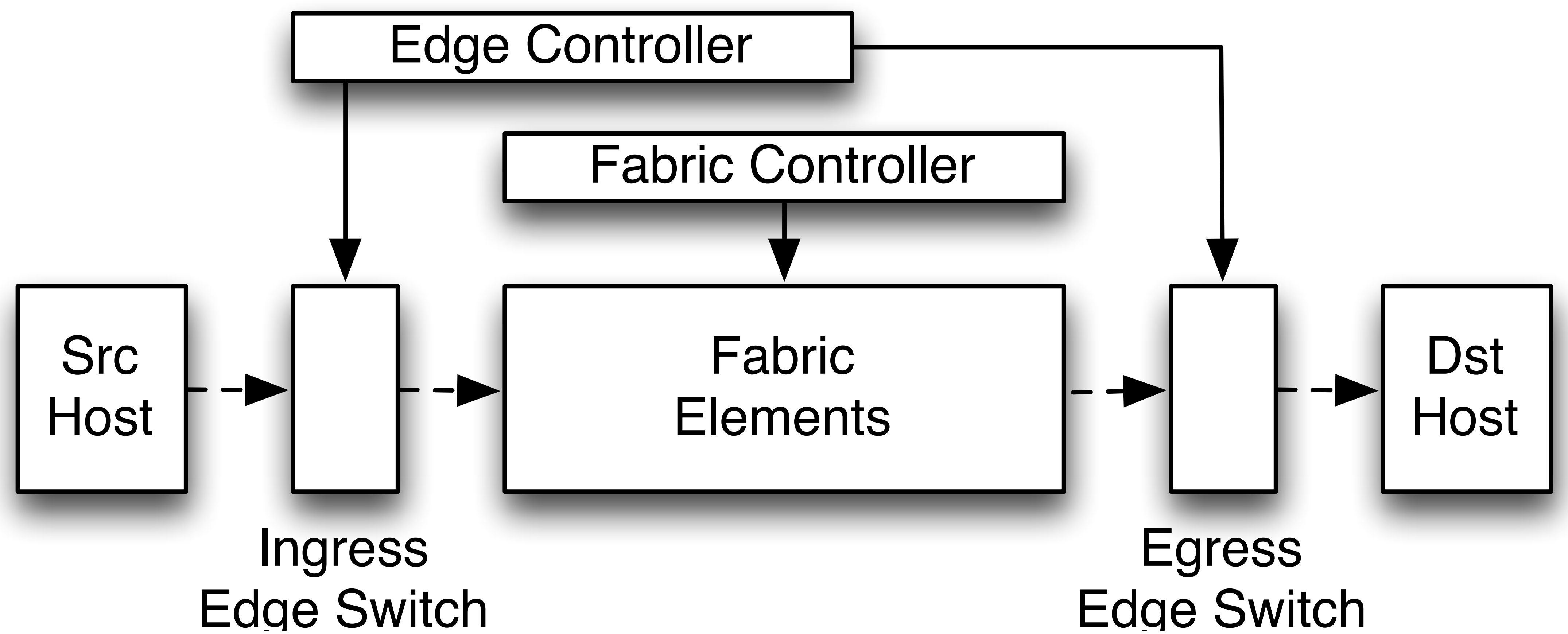
- in turn means the generality of the host-network interface will increase, but the increased generality must also be present to every switch

unnecessary coupling the host requirements to the network core behavior

extending SDN with MPLS inspiration

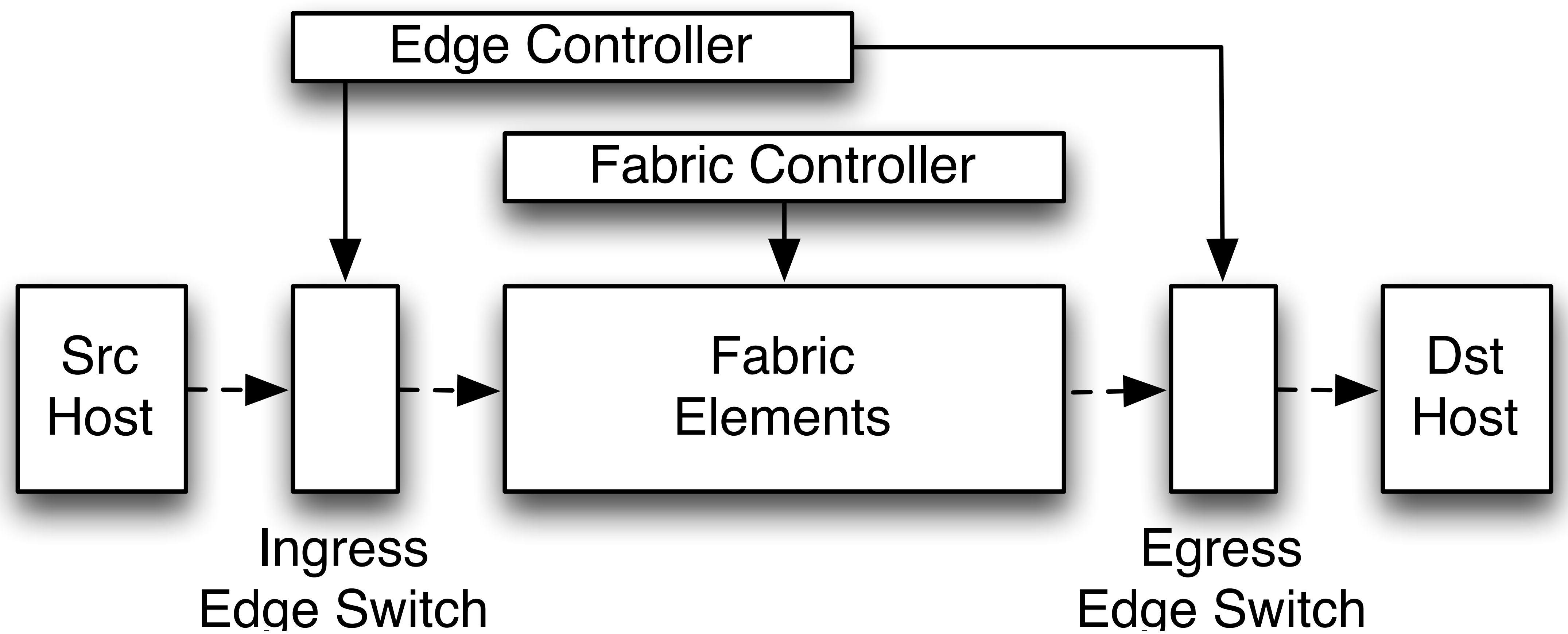
SDN architecture should incorporate "fabric"

- fabric is a transport element



extending SDN with MPLS inspiration

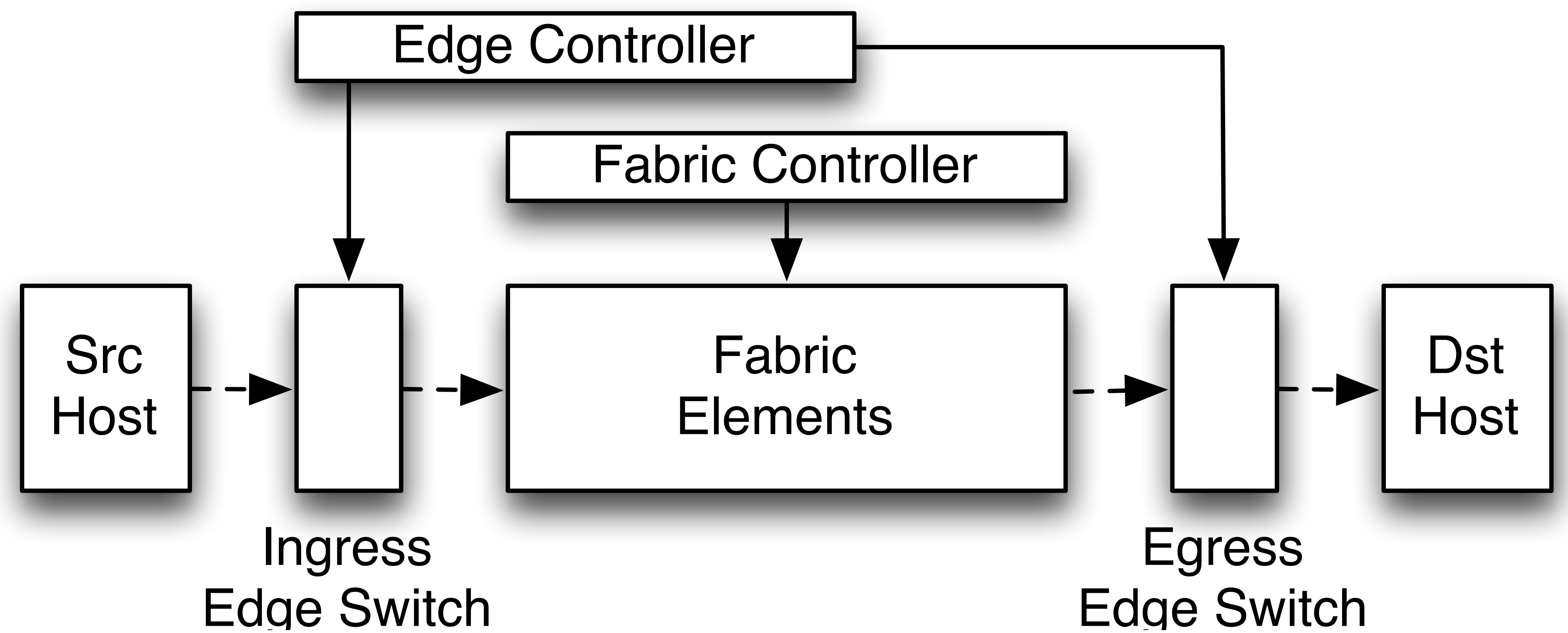
three components: hosts, edge (ingress, egress), fabric (core)



extending SDN with MPLS inspiration

host

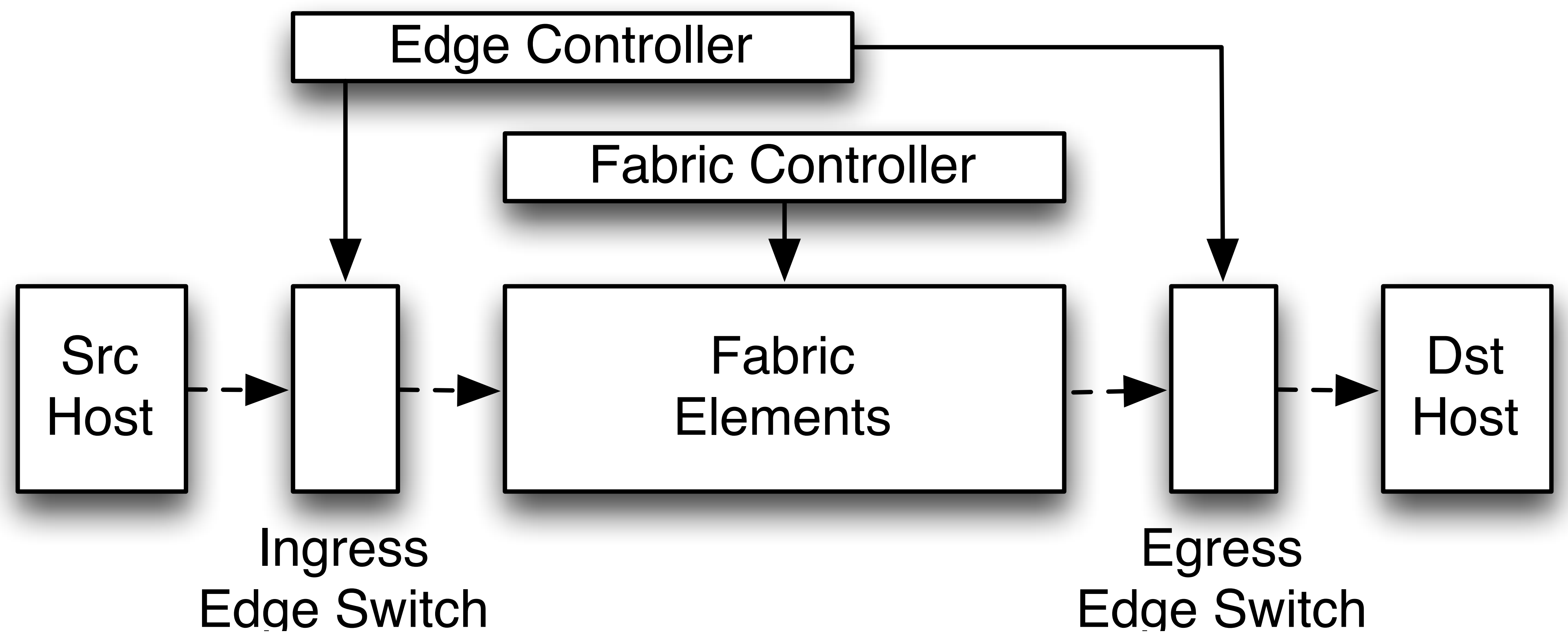
- generator and destination of traffic



extending SDN with MPLS inspiration

edge

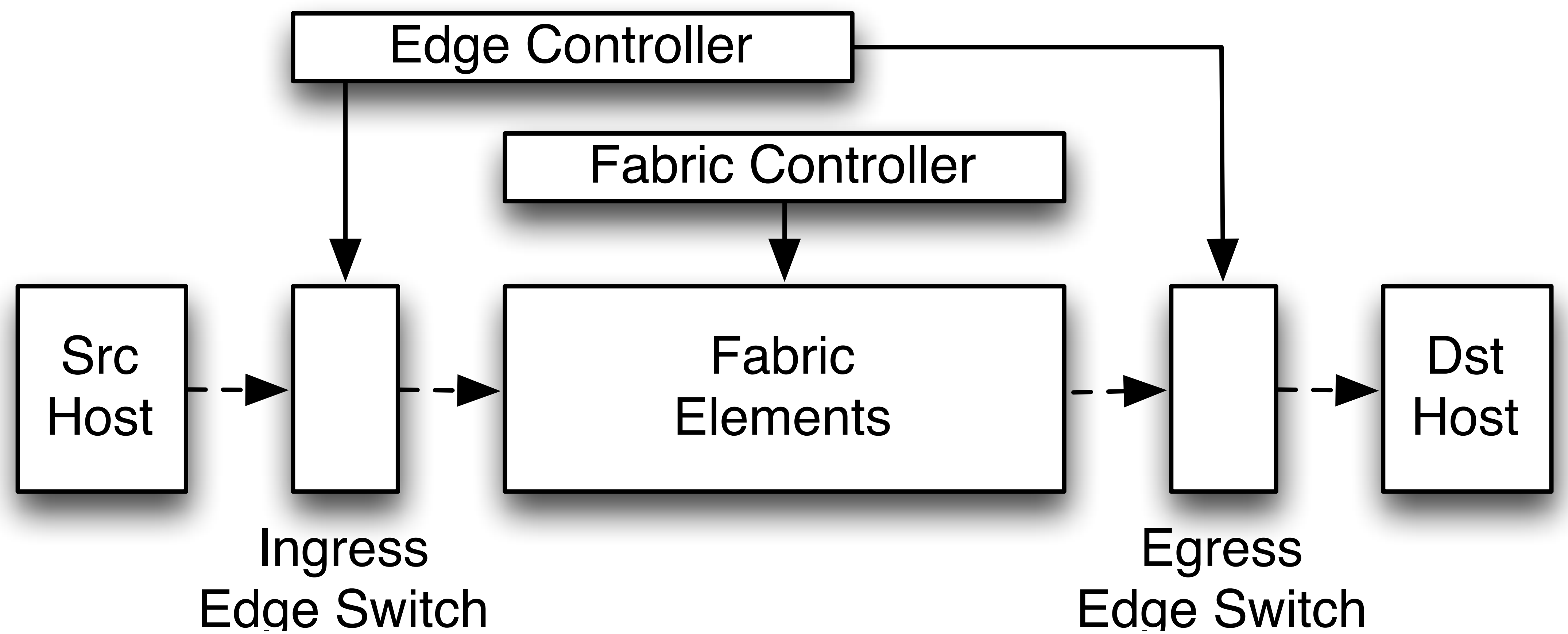
- (ingress + edge controller) provide the host-network interface
- edge controller provides operator-network interface



extending SDN with MPLS inspiration

fabric

- packet-switch interface (packet transfer alone)



extending SDN with MPLS inspiration

edge implements network policy and manage end-host addressing while the fabric interconnects as fast and cheaply as possible