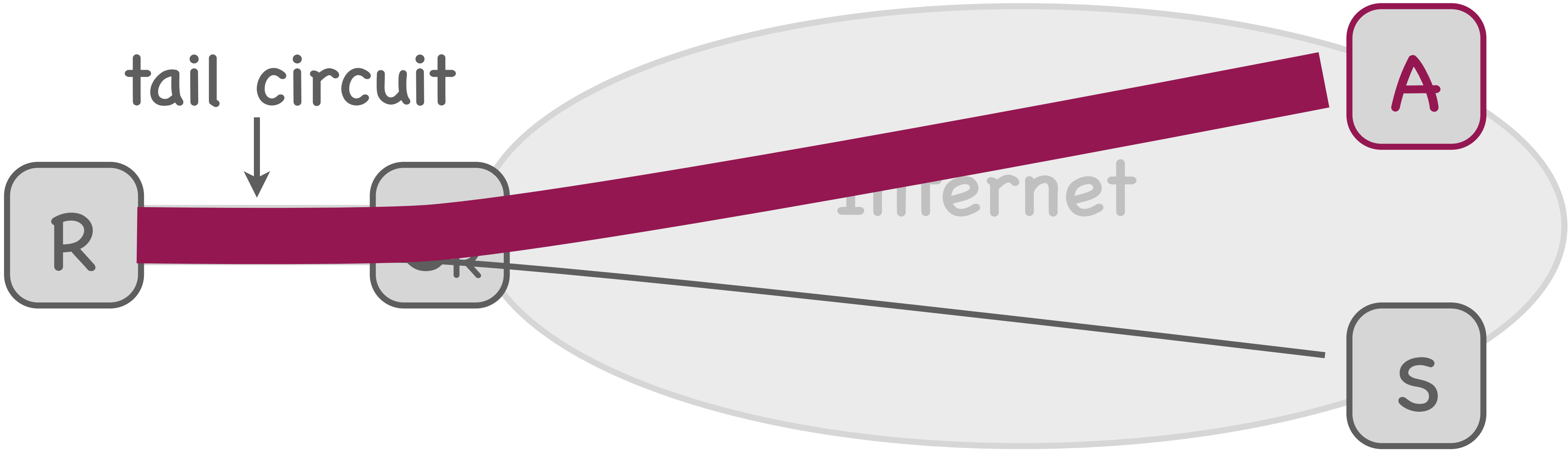
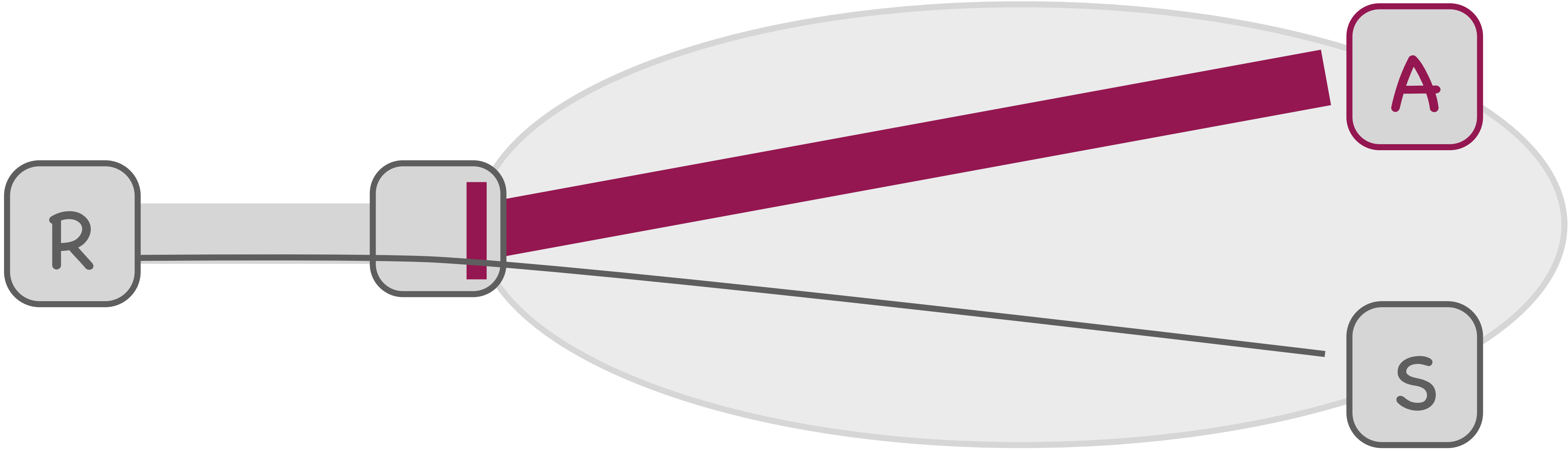
Blocking Flooding Attacks

Bandwidth flooding



Target: tail-circuit bandwidth

Network filtering

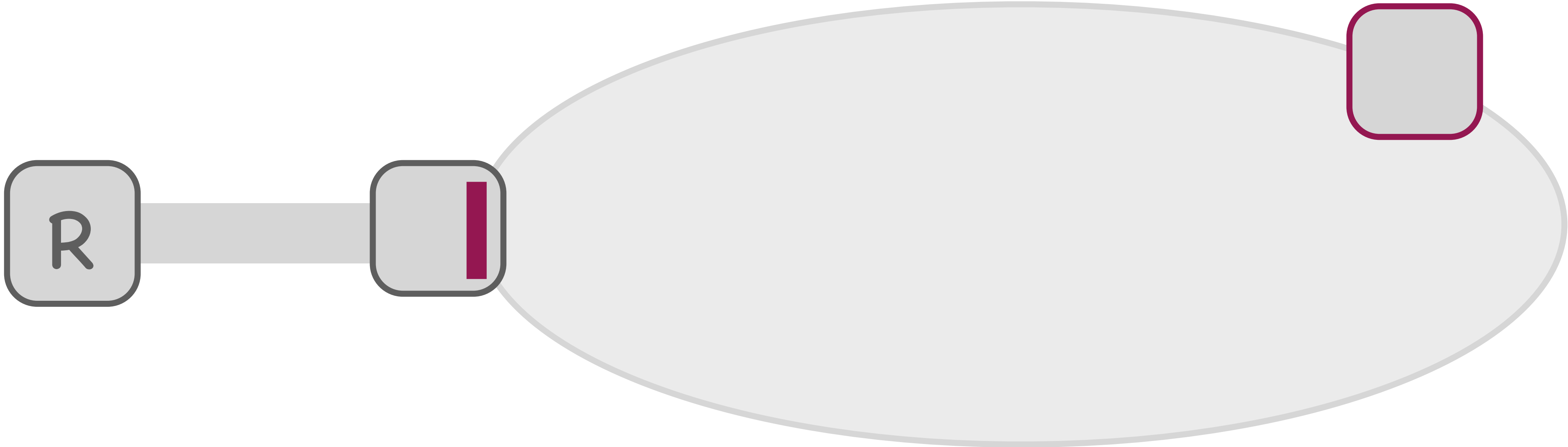


State: {A, R}

```
Code: if ( {packet.src, packet.dst} in State)
                  block packet;
```

Block attackers at the receiver's gateway

State

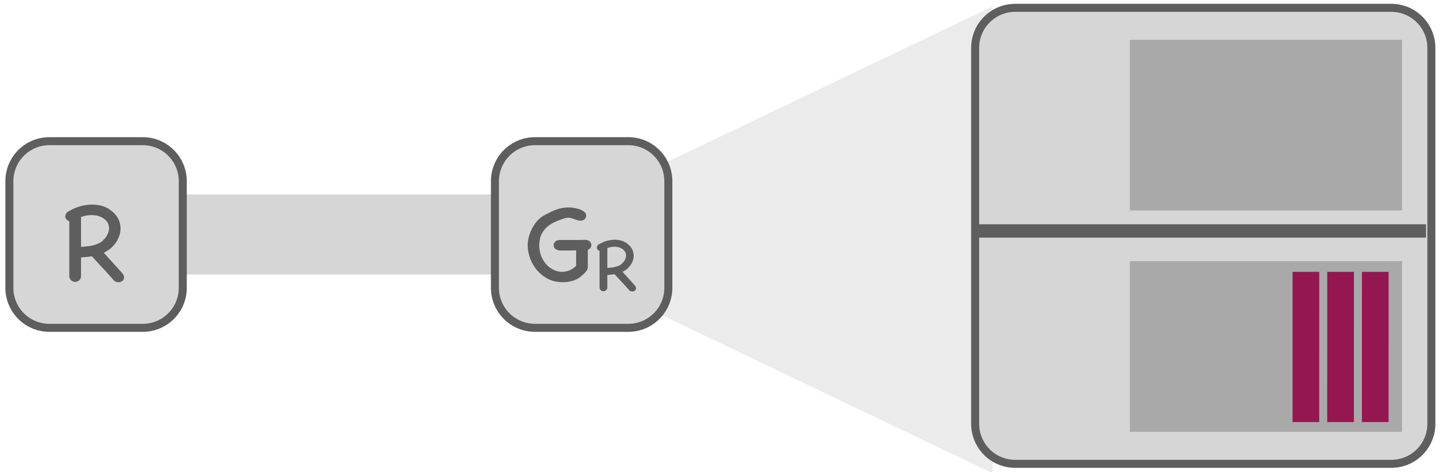


State: {attacker, receiver} pairs

Where: receiver's gateway

Managed: locally

Internet routers

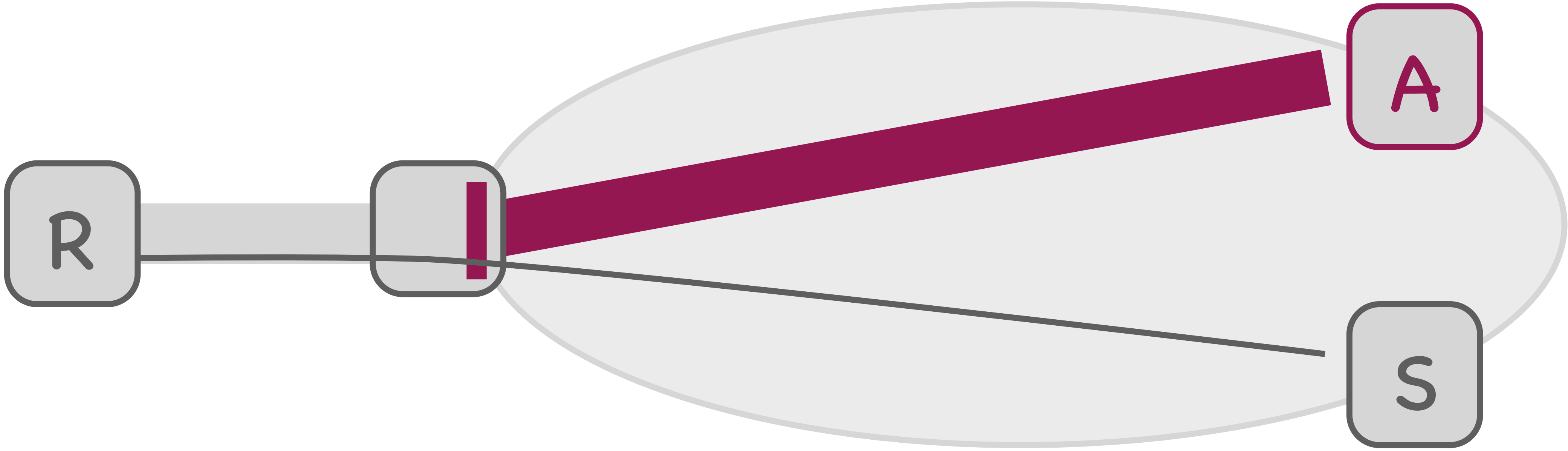


control plane
(cheap) DRAM

data plane
(expensive) SRAM

Network filtering is expensive

Network filtering

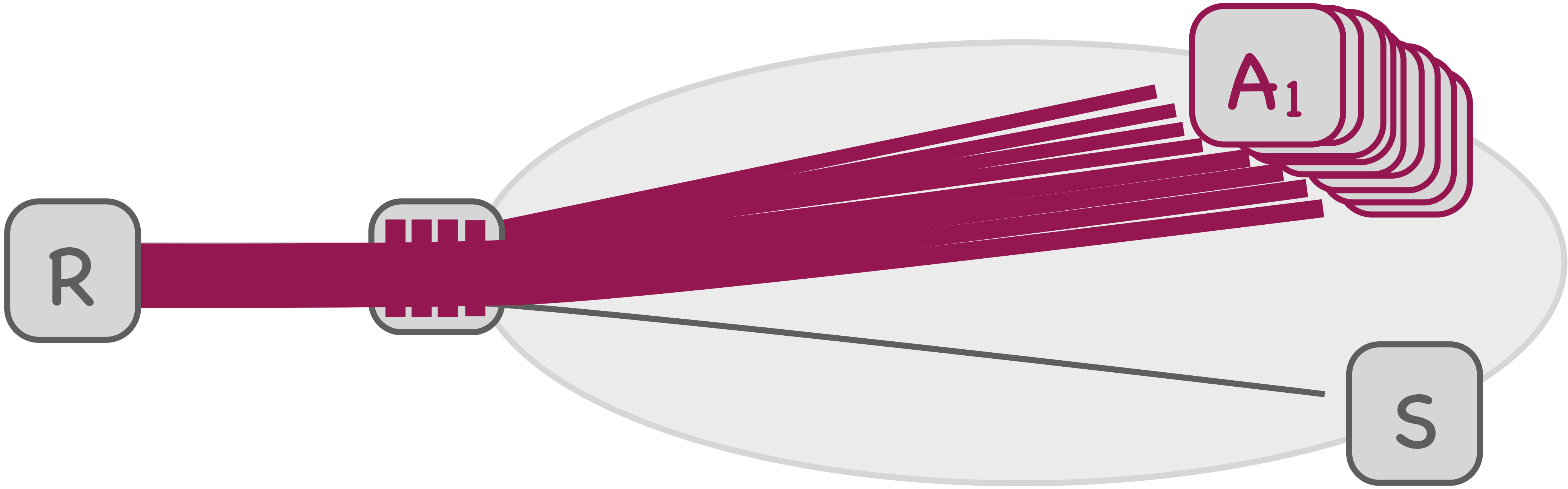


State: {A, R}

```
Code: if ( {packet.src, packet.dst} in State)
                block packet;
```

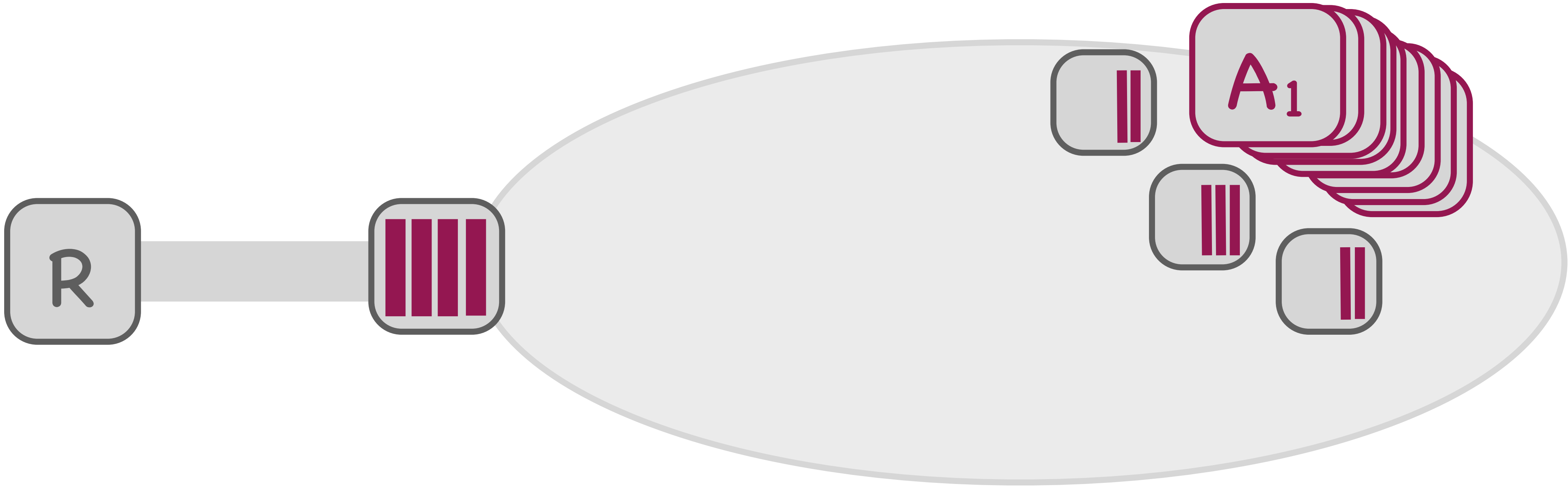
Block attackers at the receiver's gateway

Distributed flooding



Target: filtering resources + tail circuit

Distributed filtering

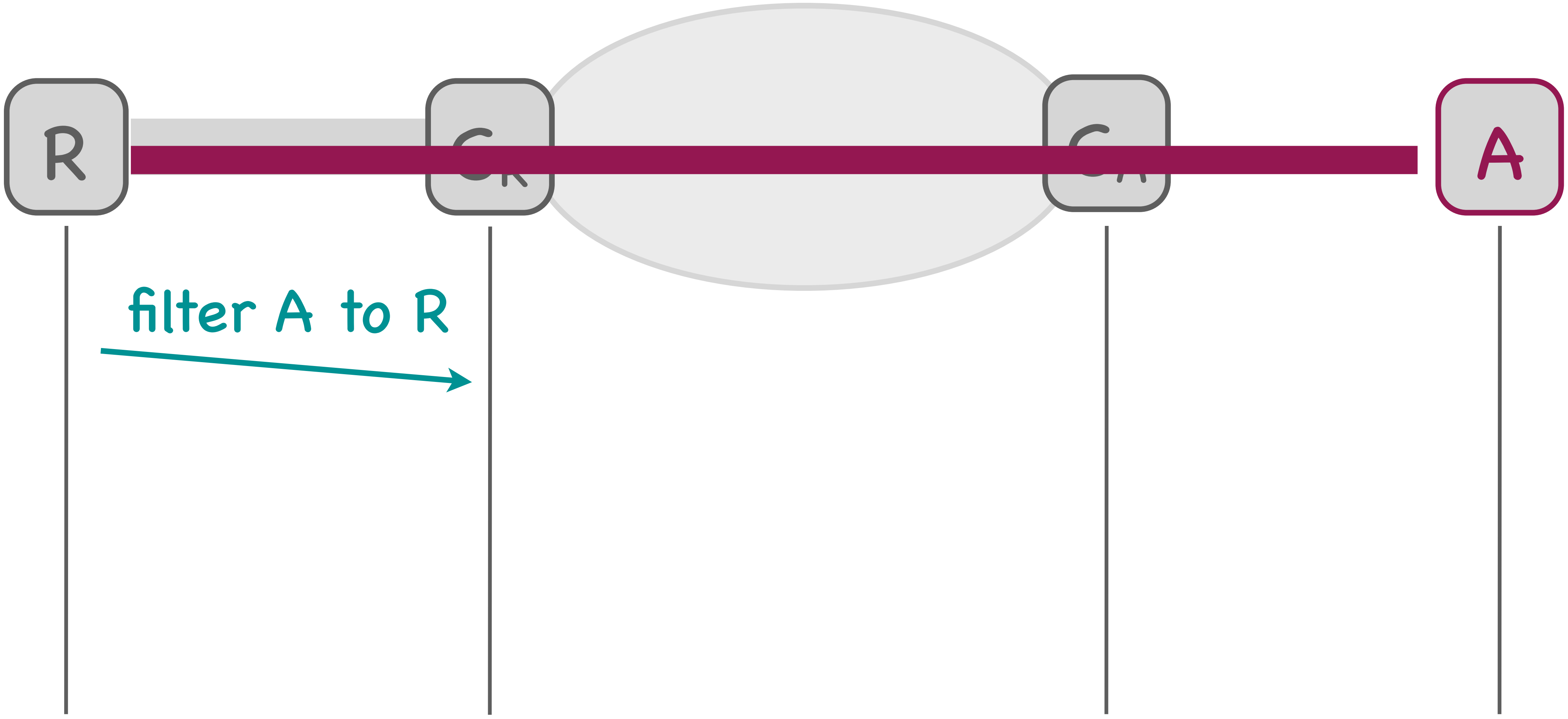


Identify routers close to attack sources

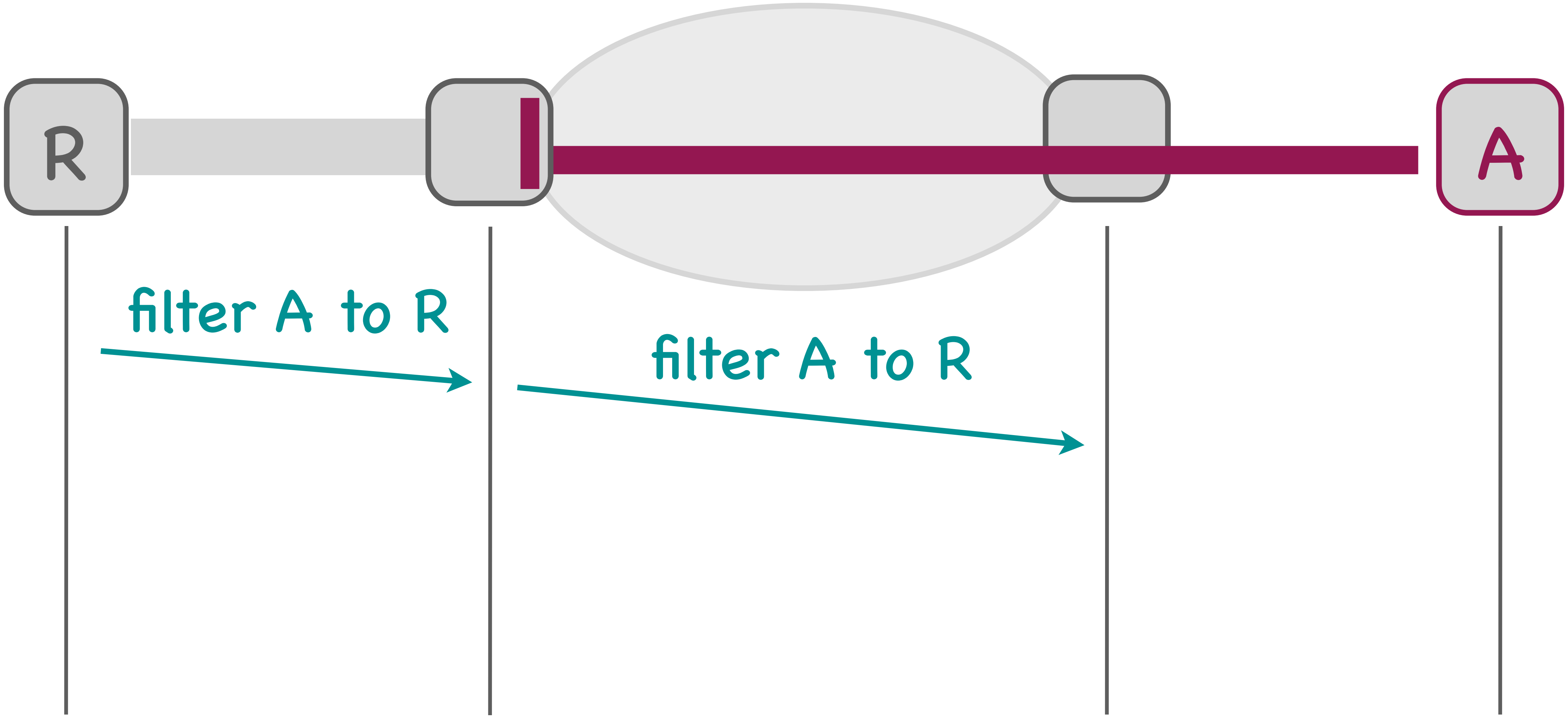
Ask them to block attack traffic

Need a filter-propagation protocol

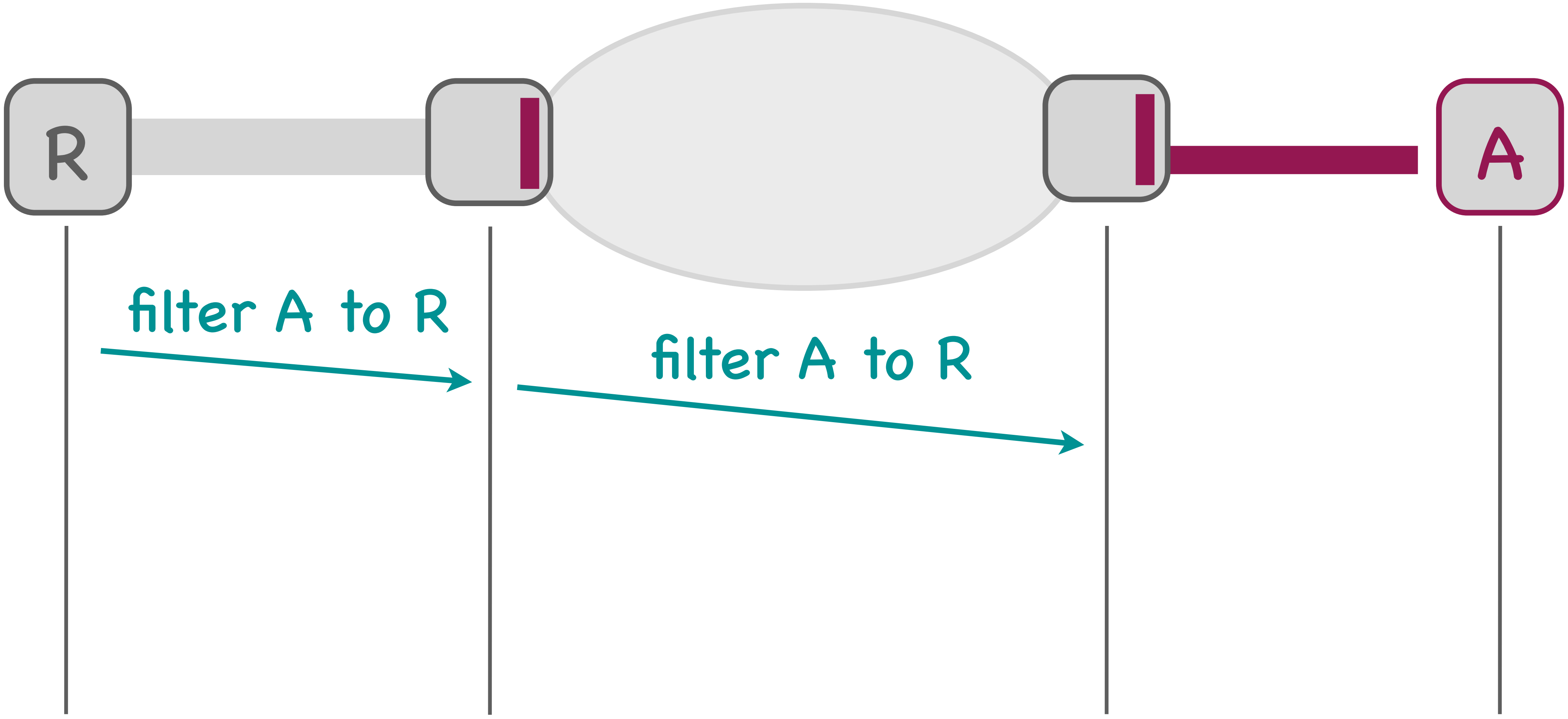
Filter propagation



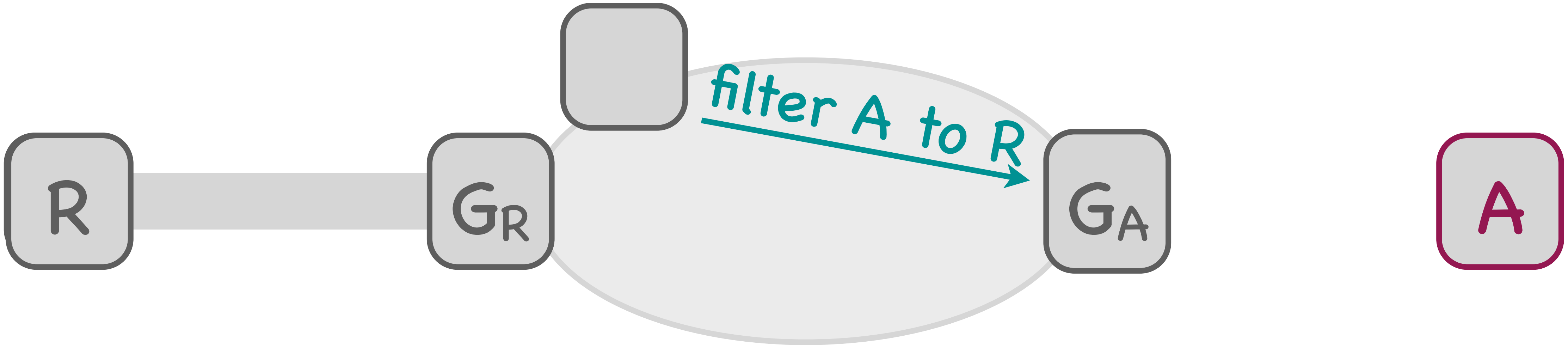
Filter propagation



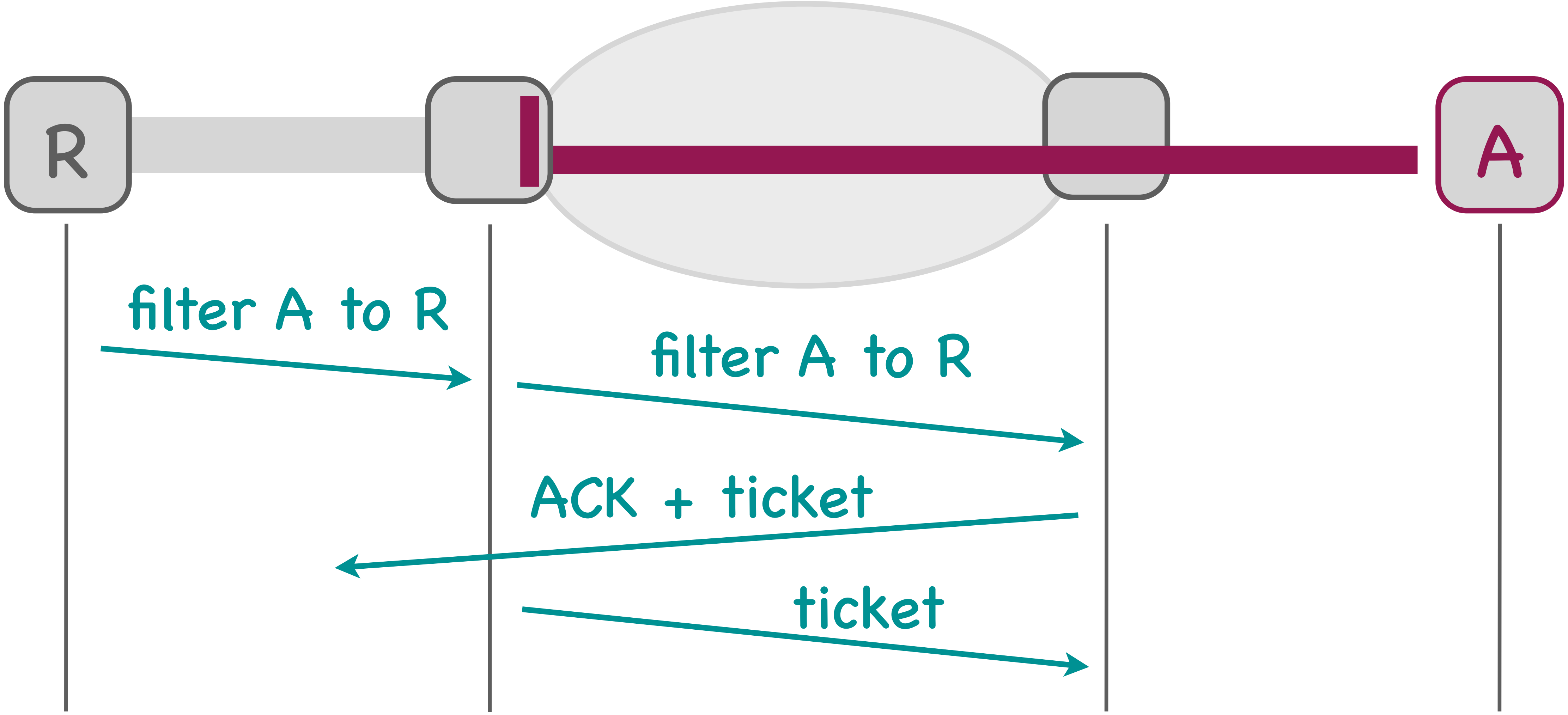
Filter propagation



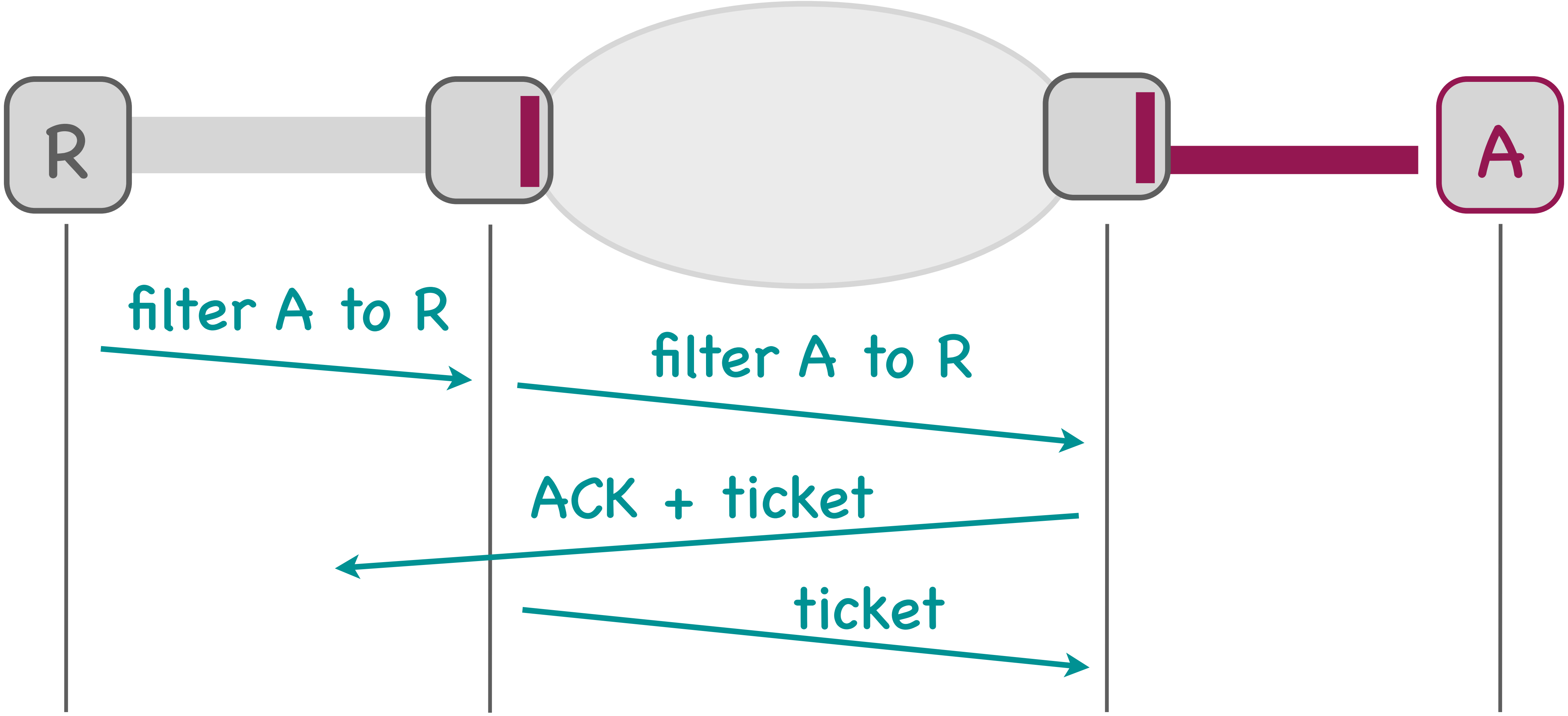
Malicious filtering requests?



Filter propagation continued

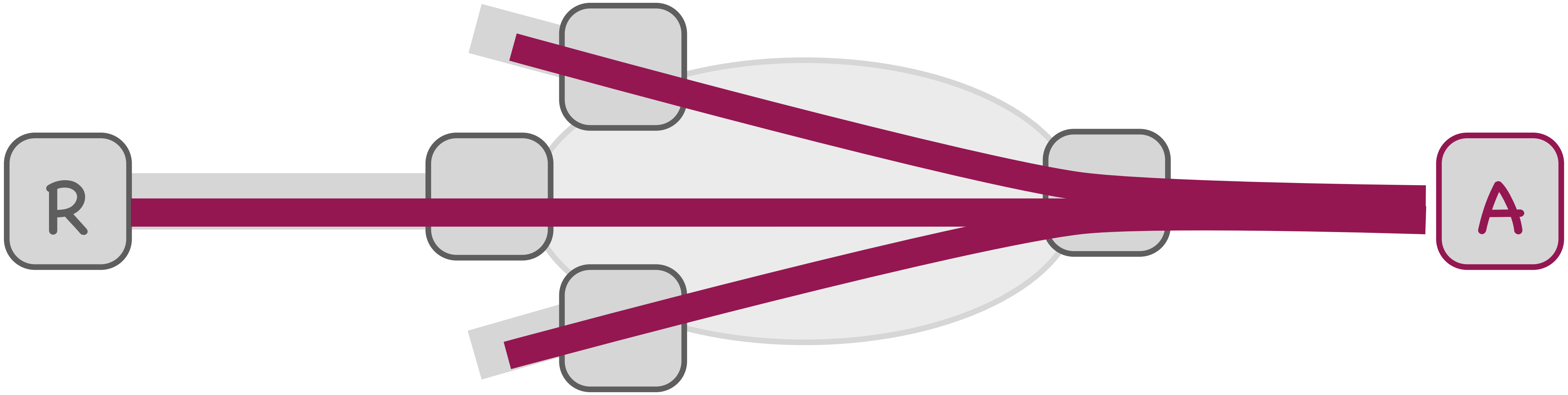


Filter propagation continued

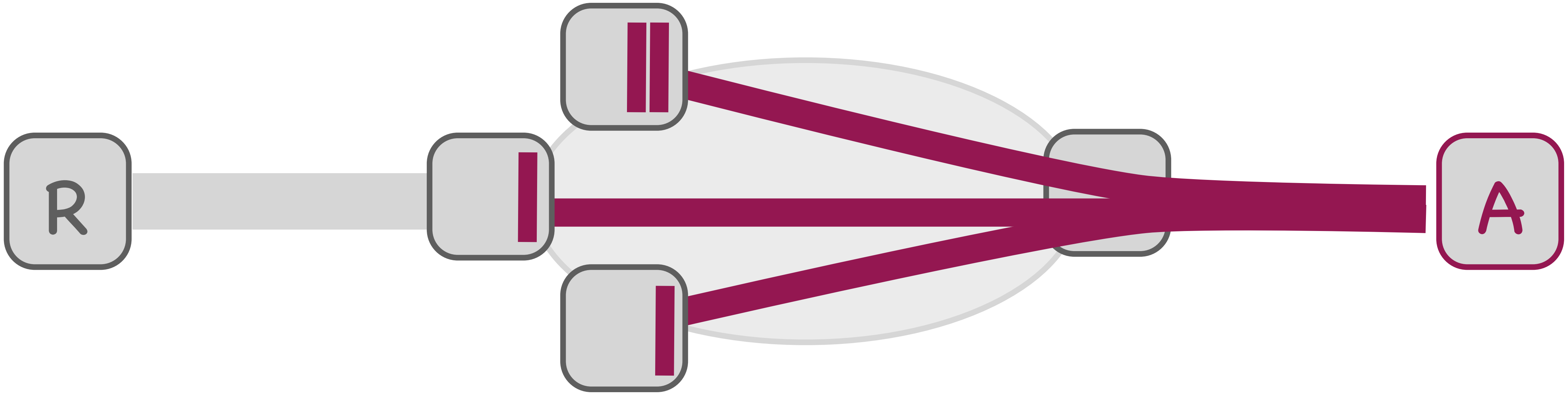


G_R proves it is on the path by 3-way handshake

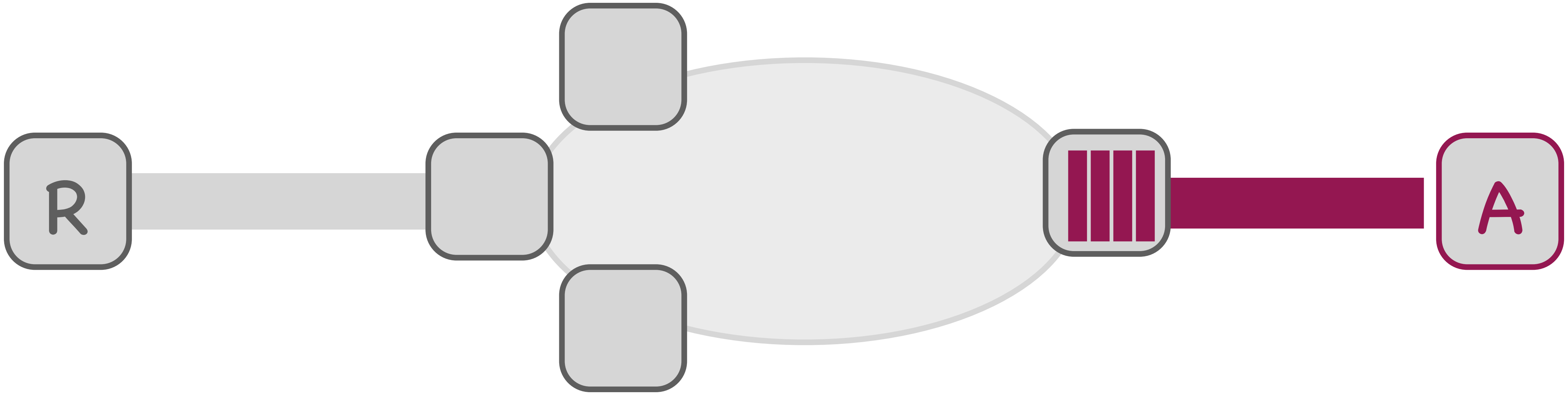
Busy attackers?



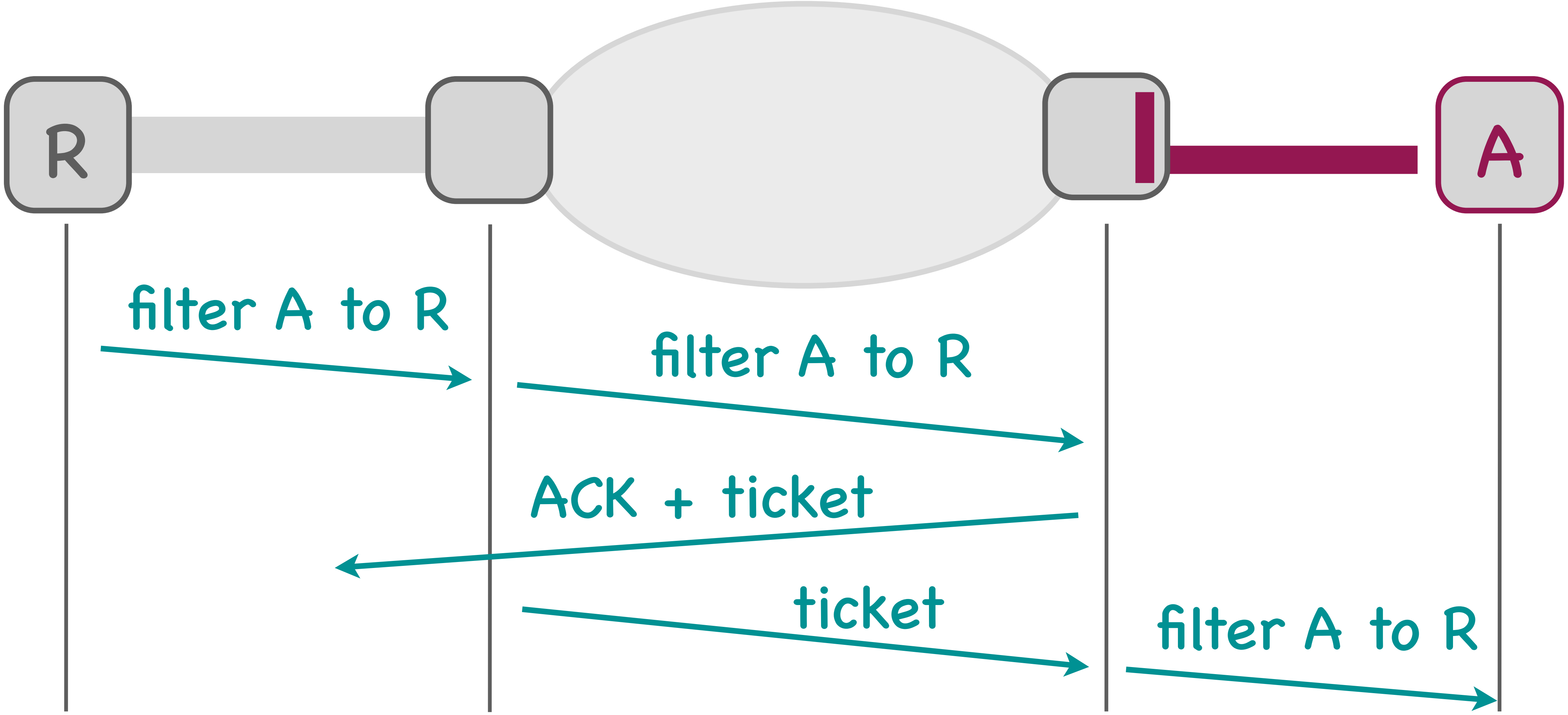
Busy attackers?



Busy attackers?

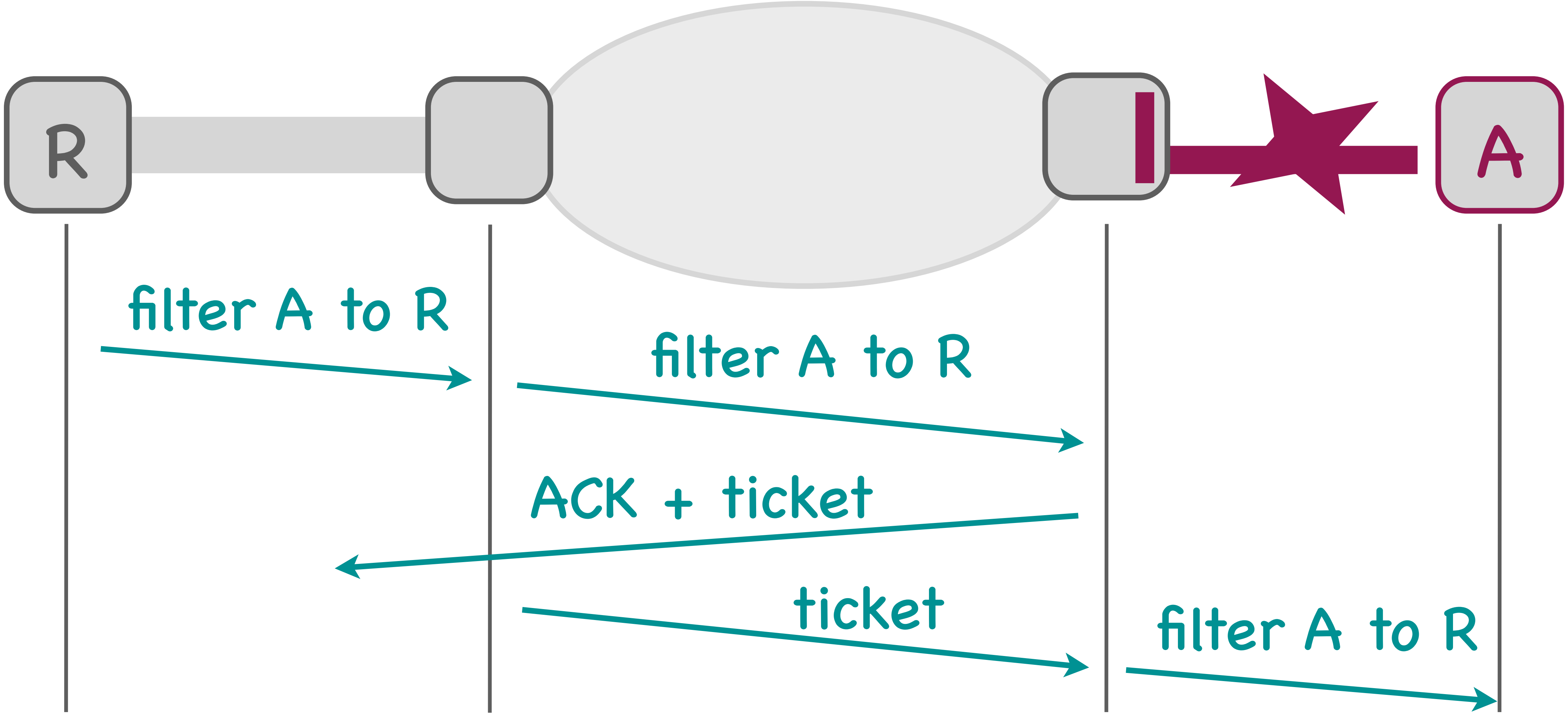


Filter propagation continued



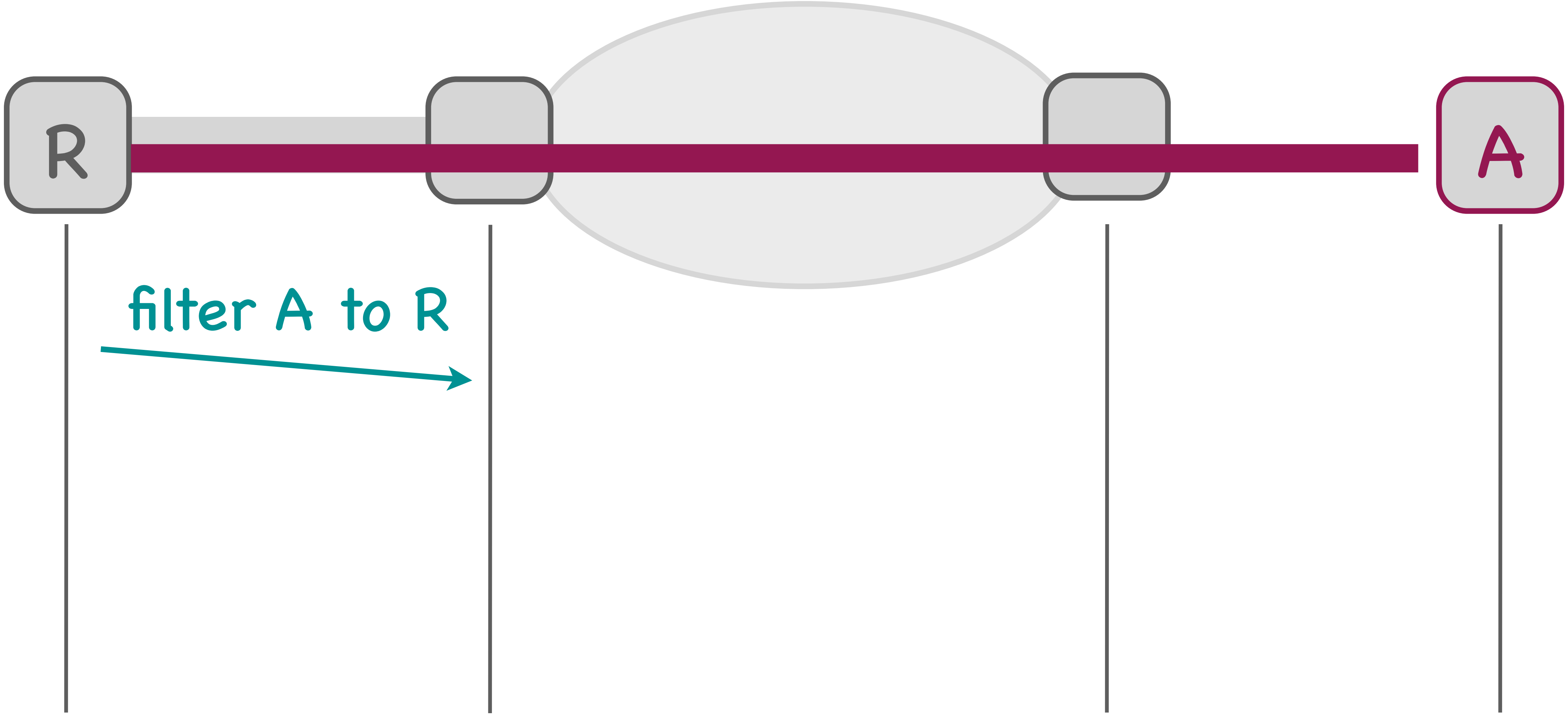
Keep in-network filters temporarily

Filter propagation continued

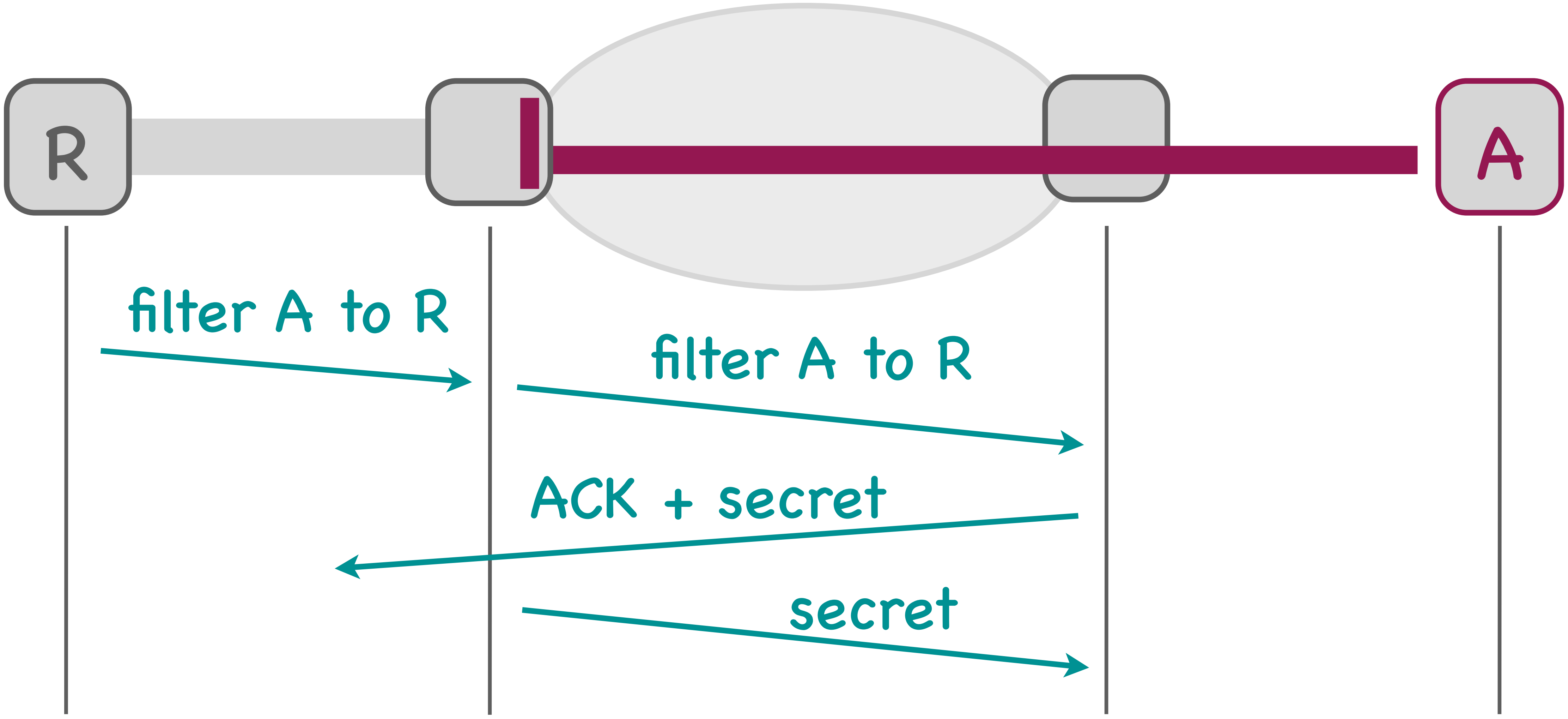


Disconnection = cheap filtering

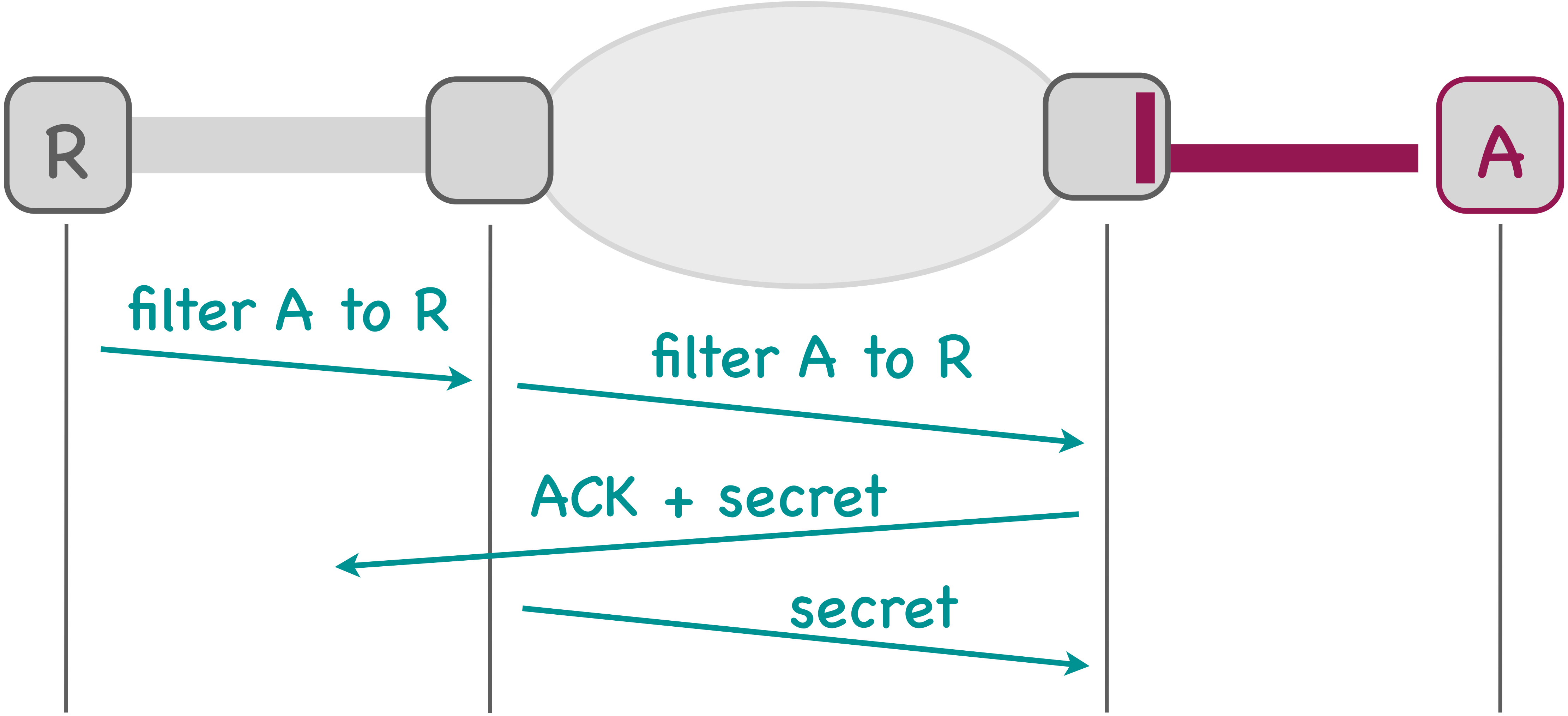
Repeat offenders?



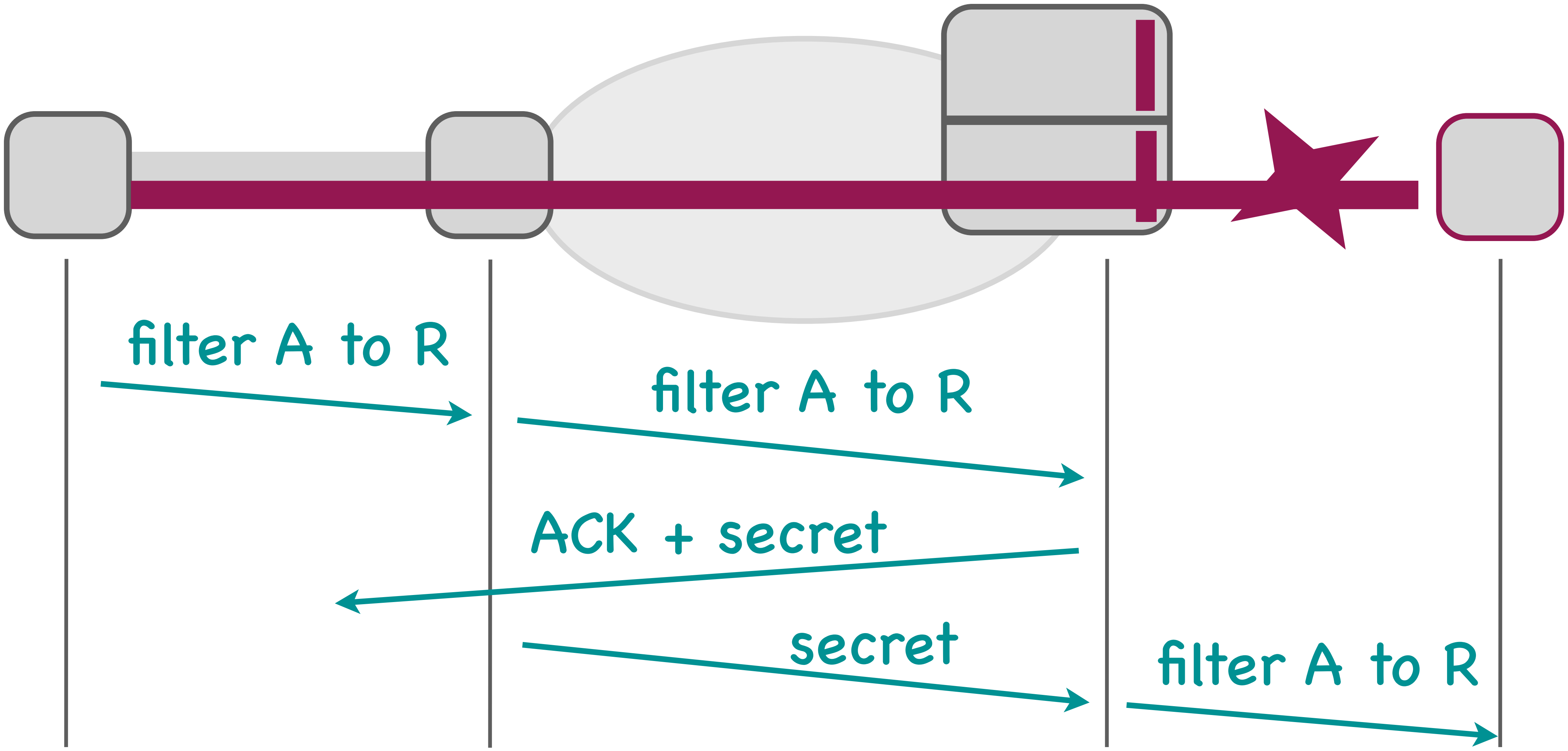
Repeat offenders?



Repeat offenders?

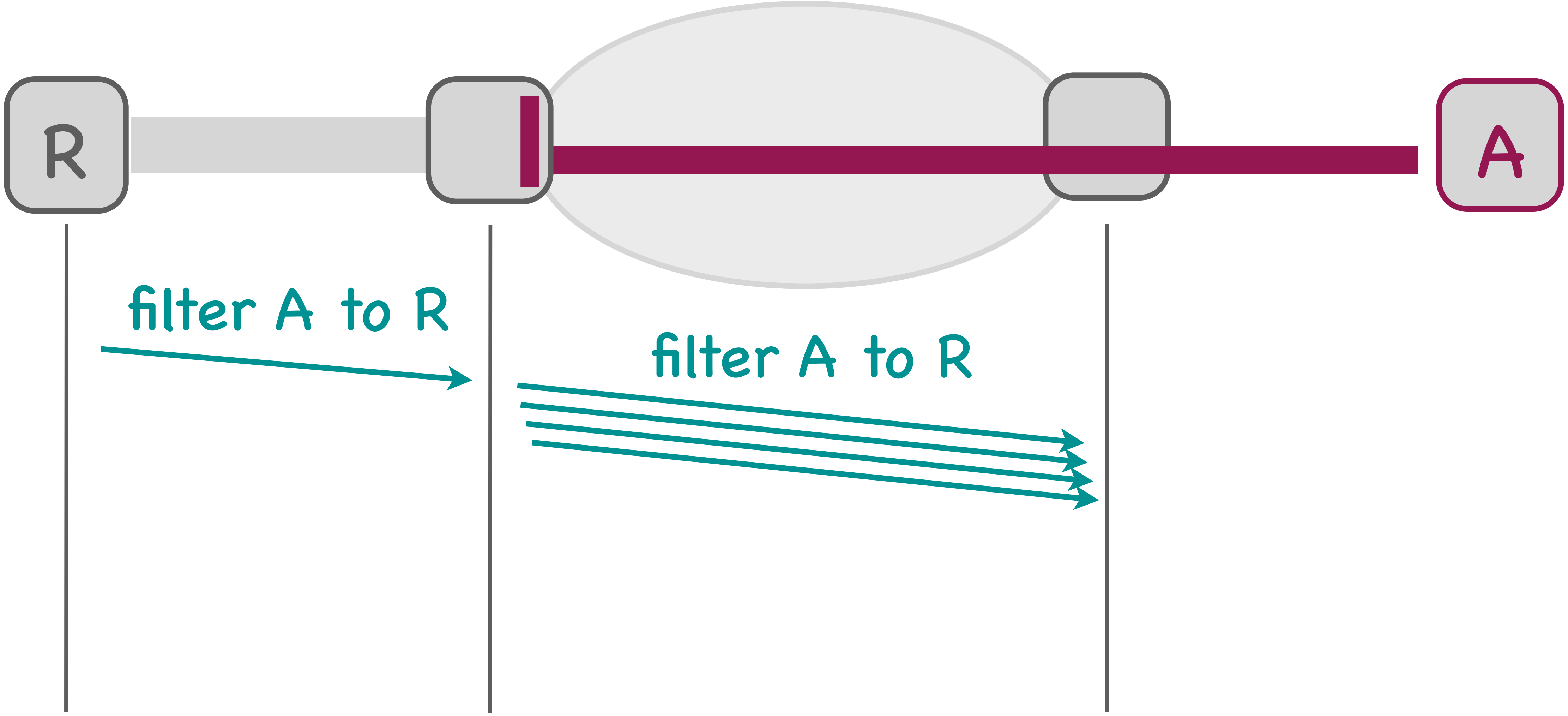


Repeat offenders?

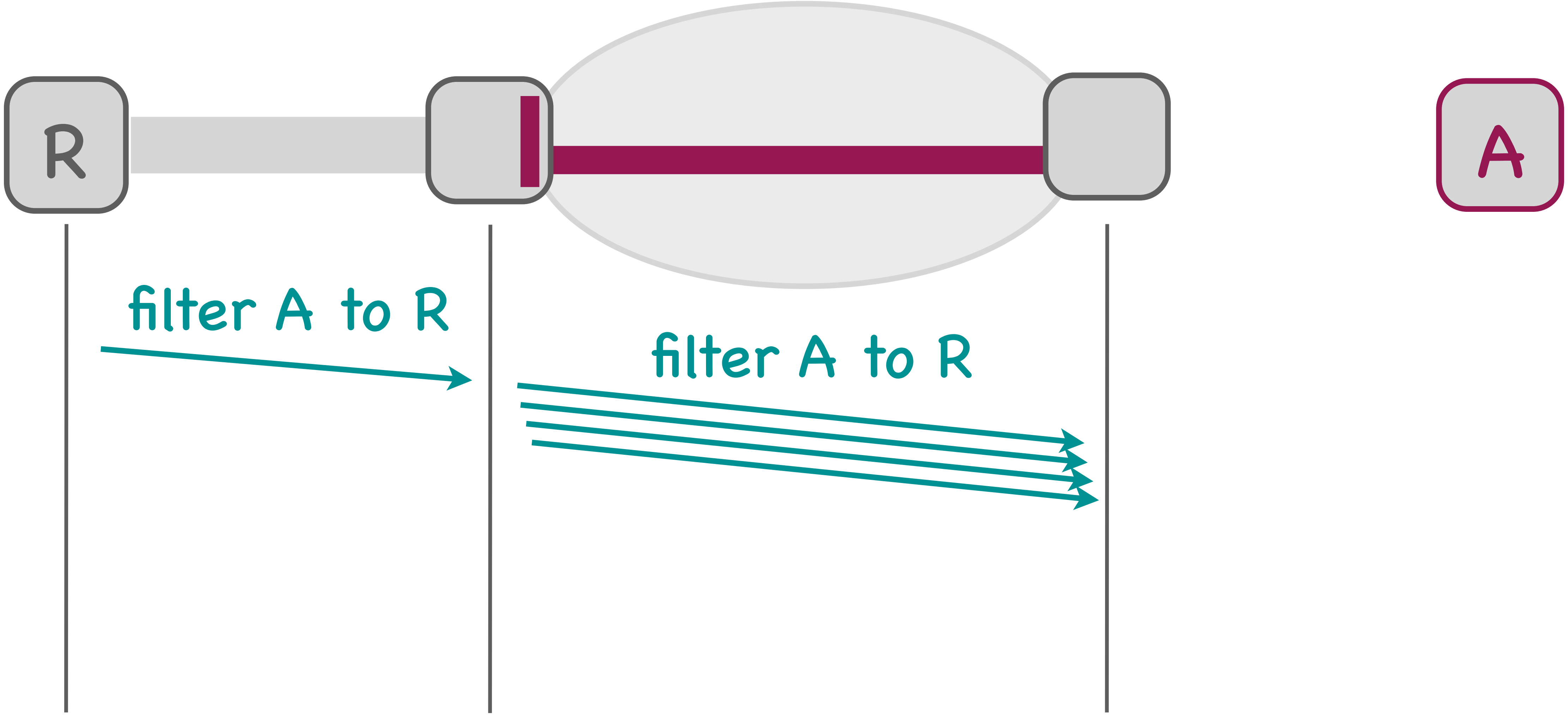


Keep filtering state in the control plane

Non-cooperative networks?

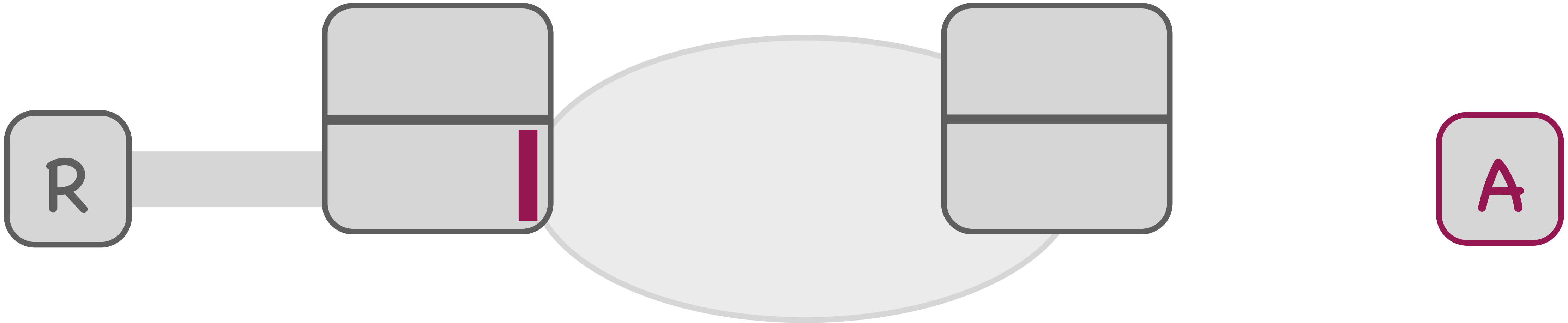


Non-cooperative networks?



... get disconnected from R

State

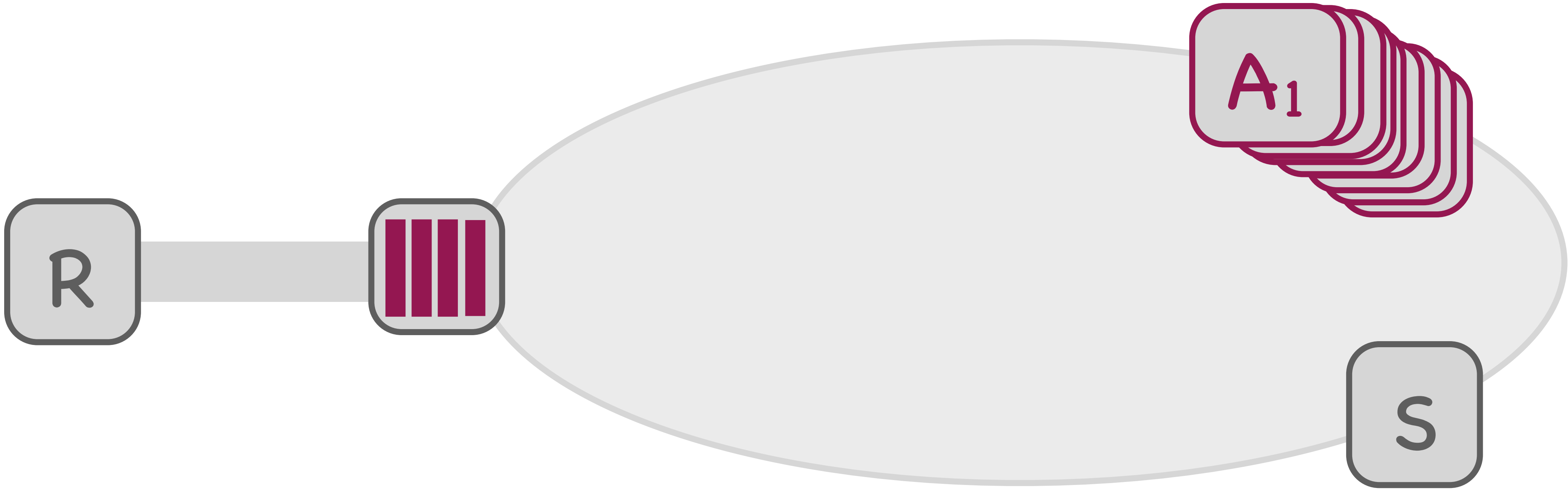


State: {attacker, receiver} pairs

Where: control plane of attacker's gateway

Managed: filter-propagation protocol

Distributed flooding



Target: filtering resources + tail circuit

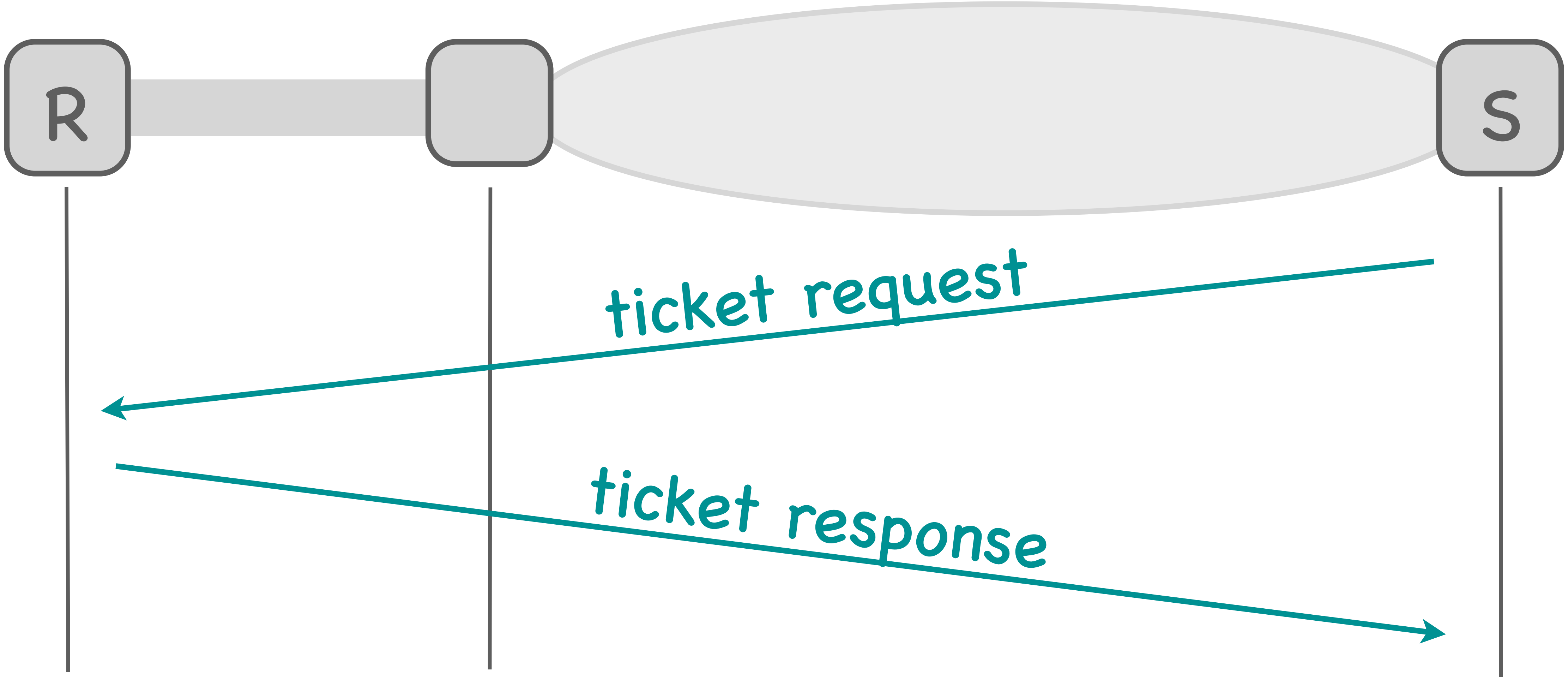
Ticket-based authorization

Give tickets to well behaved senders

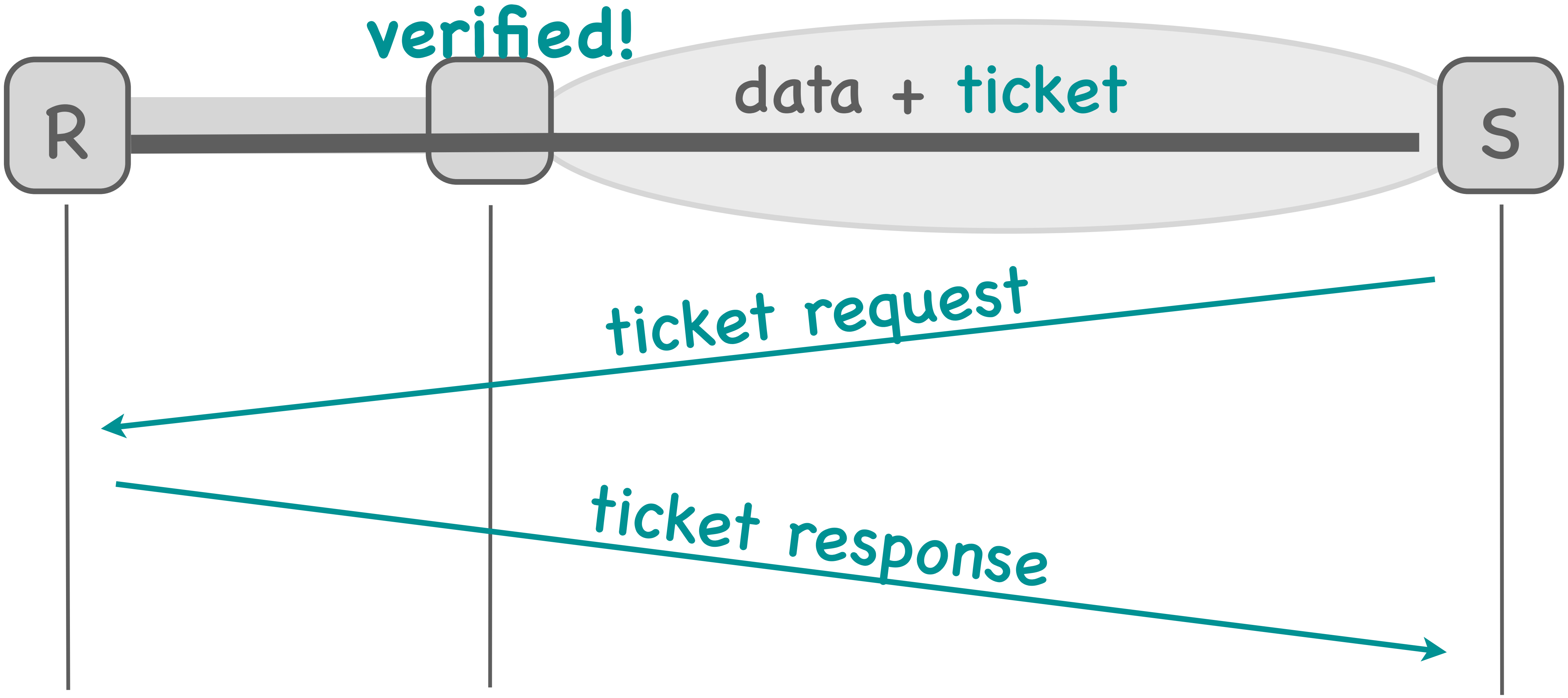
Verify tickets inside the network

Need ticket distribution and verification

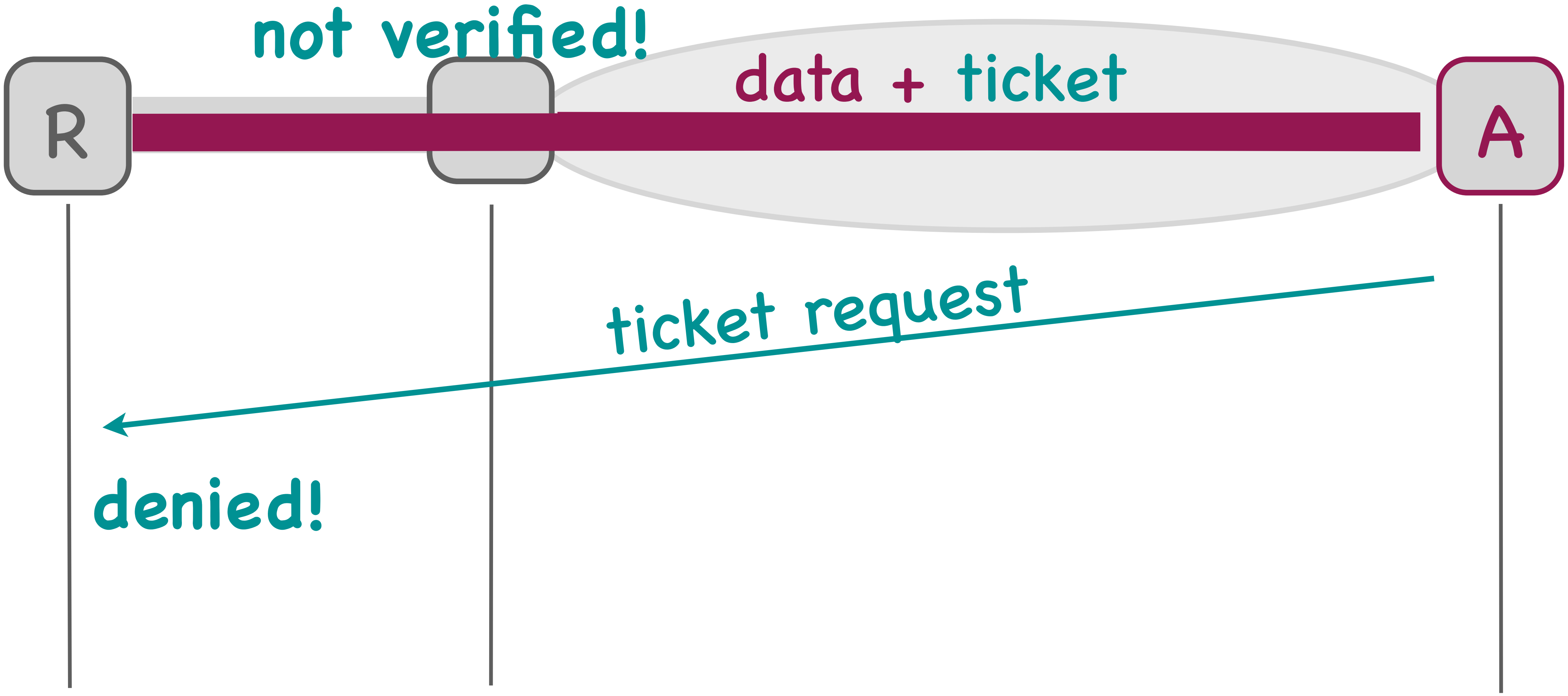
Ticket distribution



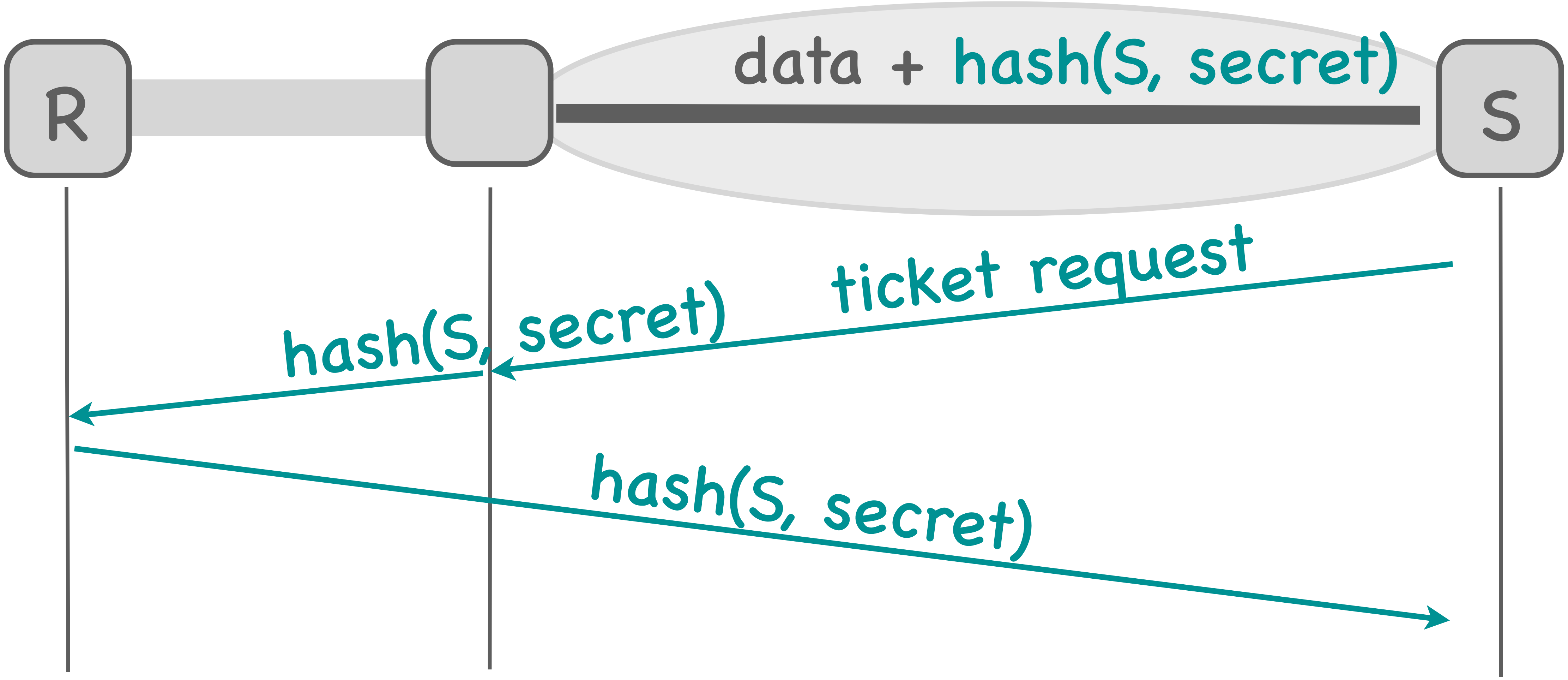
Ticket verification



Ticket verification

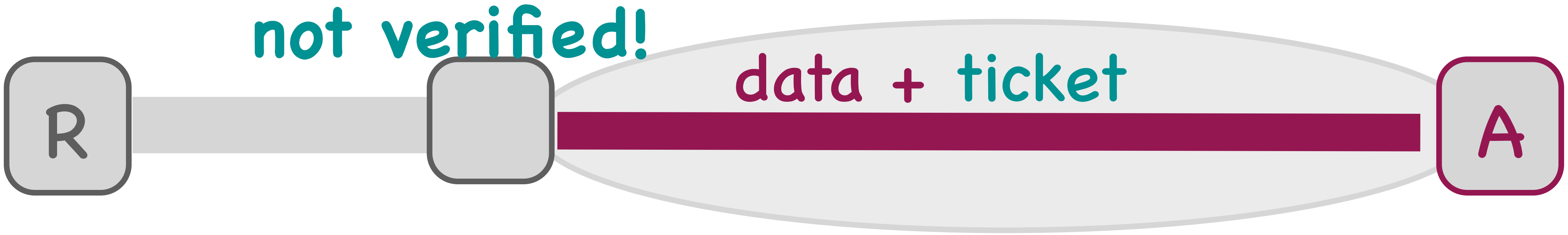


Ticket construction



S cannot guess the value of a valid ticket

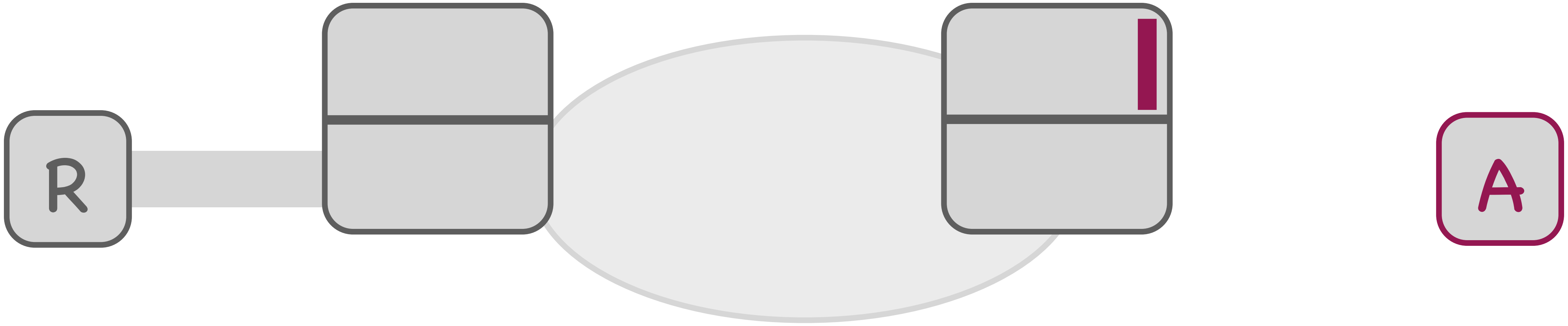
Stateless filtering



State: -

```
Code: if ( not verify(ticket) )
                block packet;
```

State

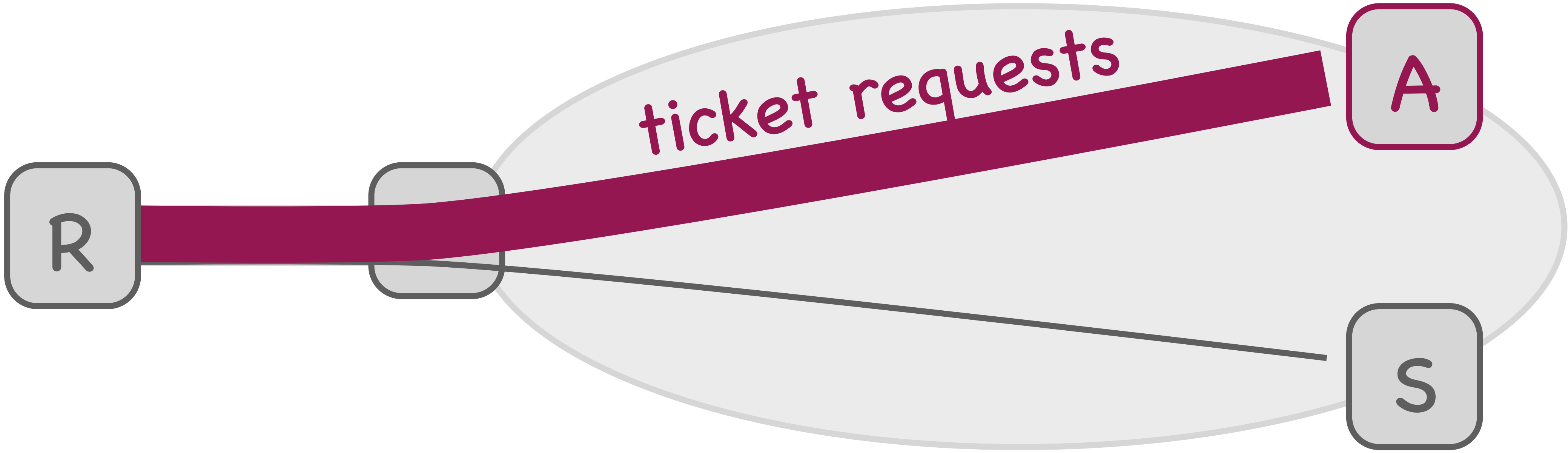


State: {sender, receiver} pairs

Where: senders

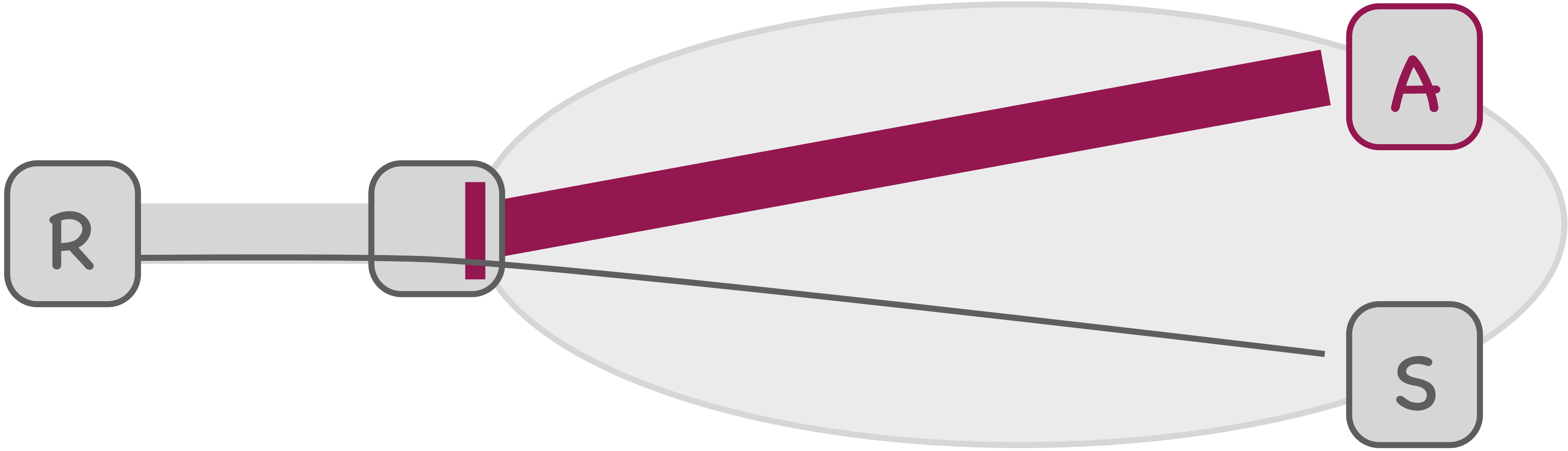
Managed: ticket-distribution protocol

Denial of ticket



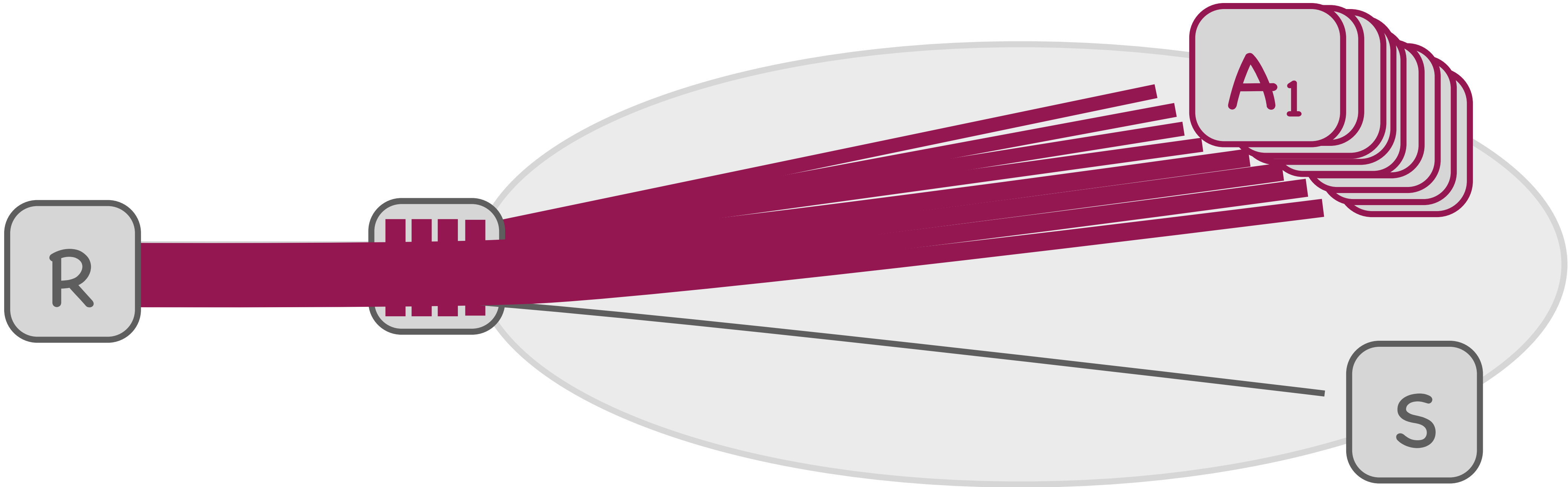
Target: tail circuit + ticket distribution

Tickets + network filtering



Block attackers in the network

Distributed denial of ticket

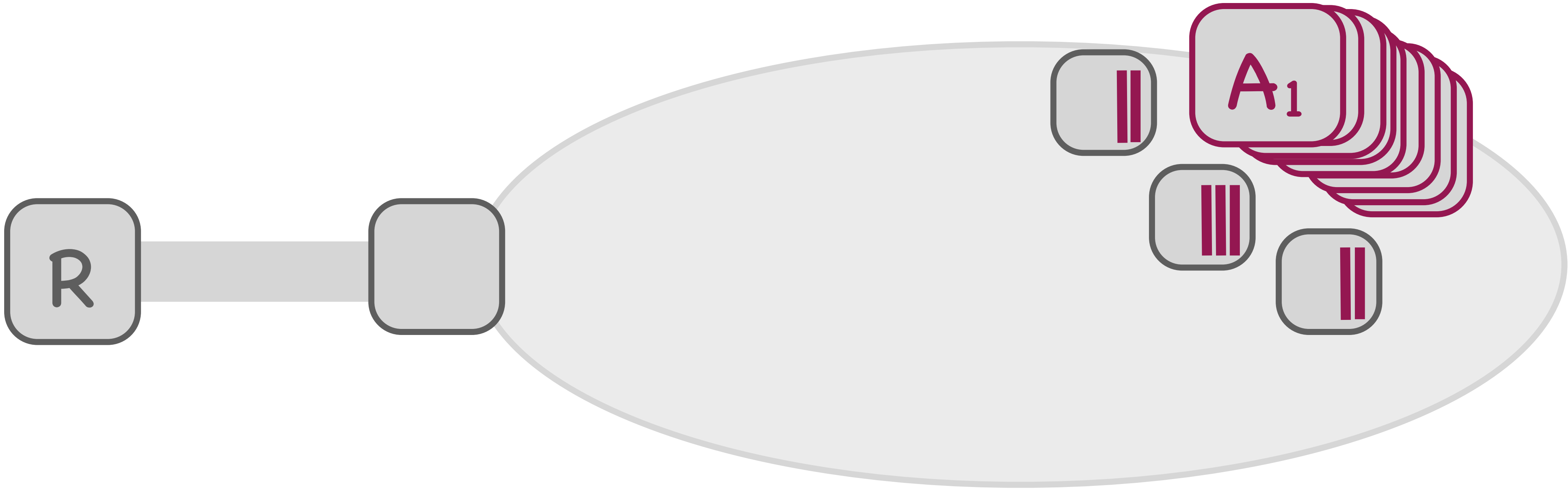


Target: filtering resources

+ tail circuit

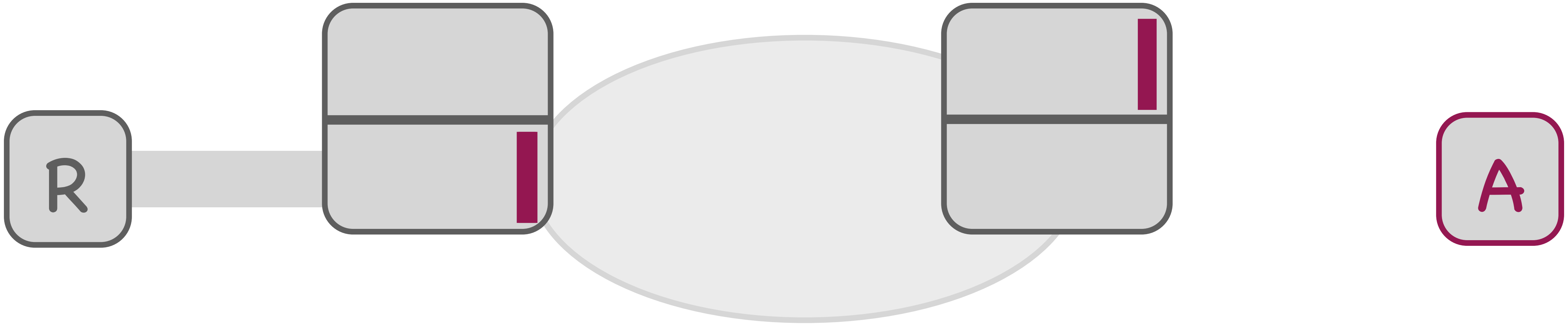
+ ticket distribution

Tickets + distributed filtering



Need a filter-propagation protocol

State

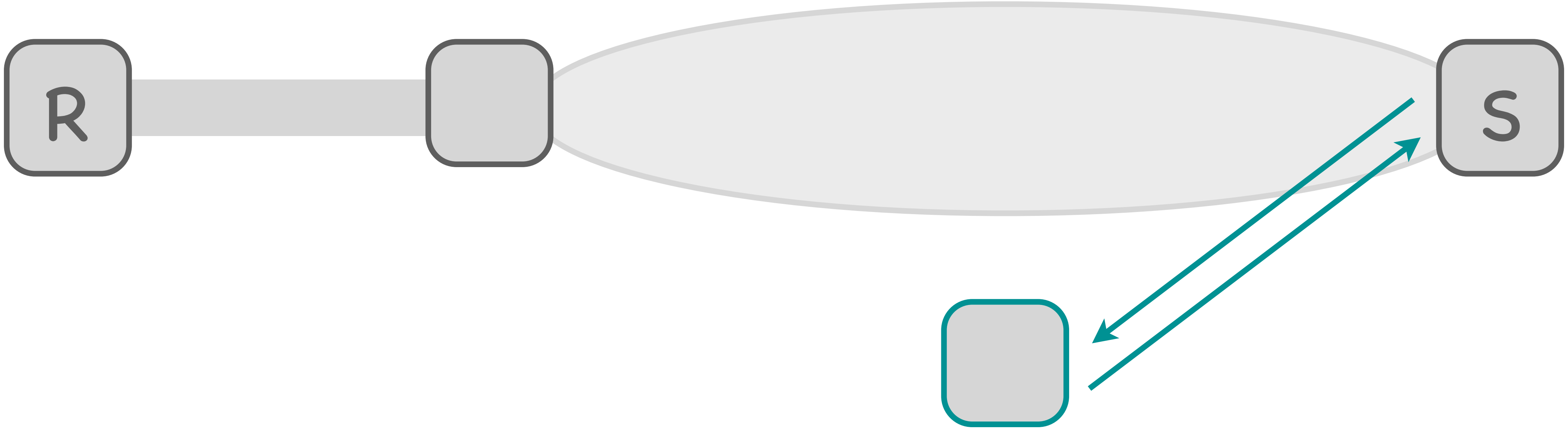


State: {sender/attacker, receiver} pairs

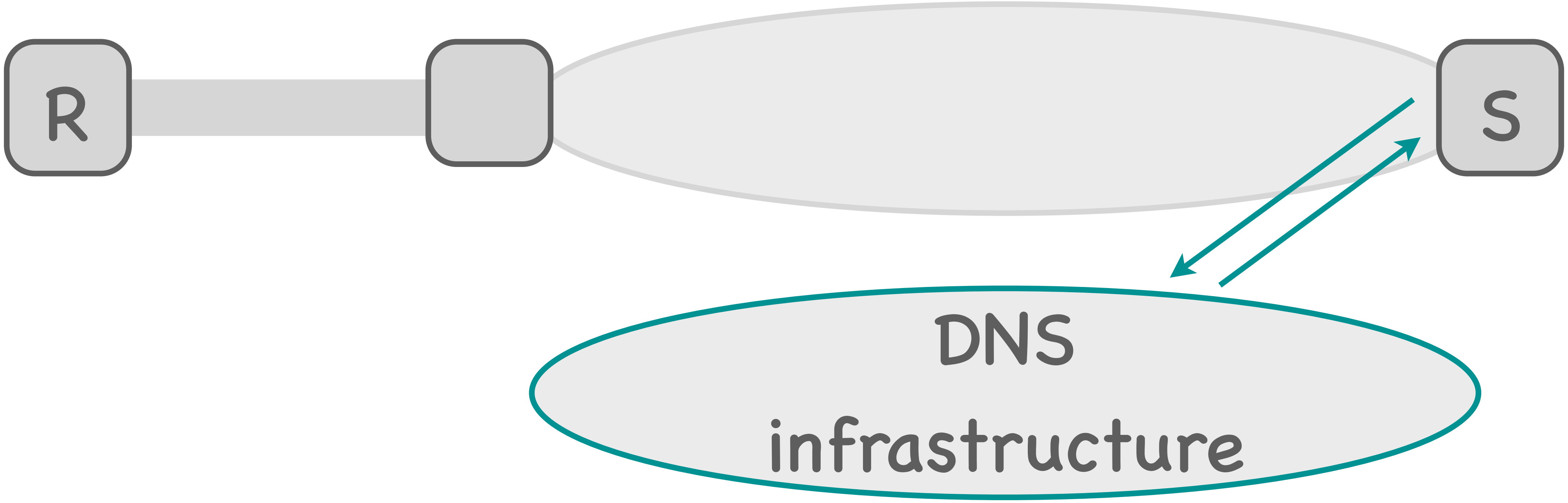
Where: senders + network

Managed: ticket distribution + filtering propagation

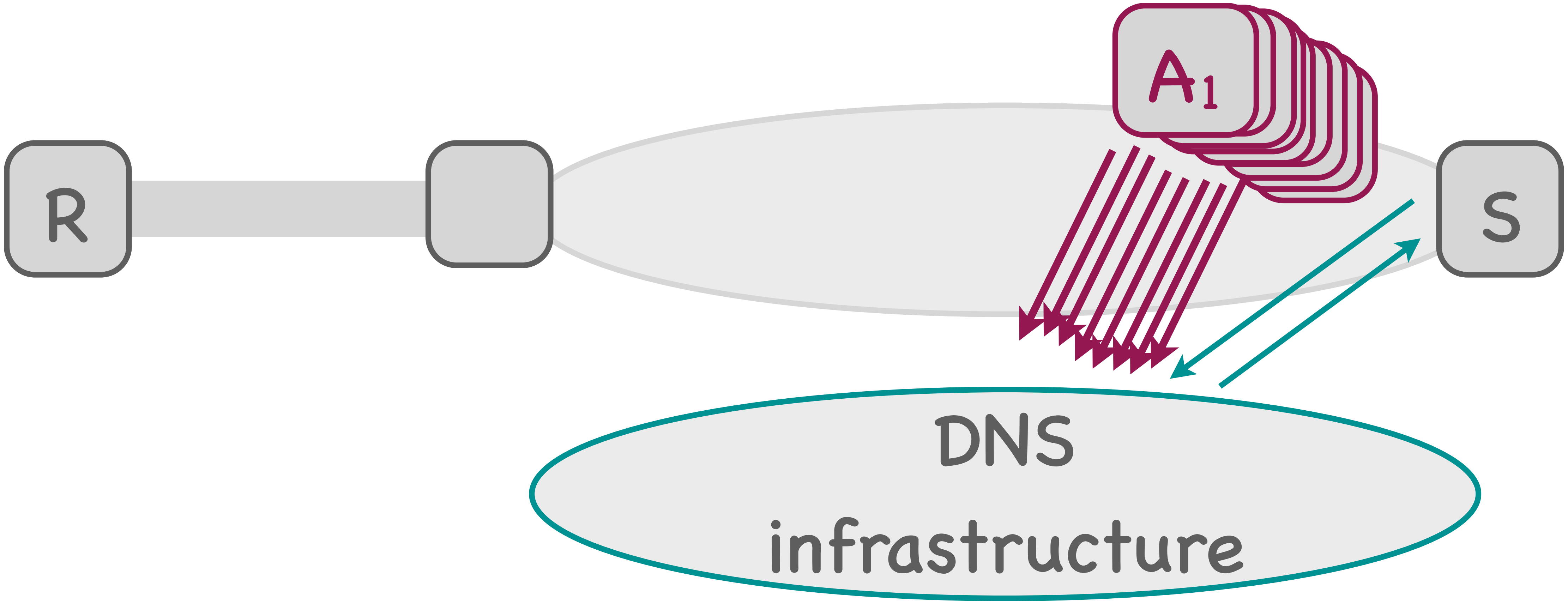
Outsource ticket distribution



Outsource ticket distribution



Outsource ticket distribution



Target: the DNS infrastructure

Fair-share the Internet

Fixed number of connections per sender

Reduces filtering state

Changes the nature of the Internet