ICN Publish/Subscribe Networking

Junxiao Shi
2012-04-26

Paper

Antonio Carzaniga, Michele Papalini, and Alexander L. Wolf. 2011. Content-based publish/subscribe networking and information-centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking* (ICN '11). ACM, New York, NY, USA, 56-61.
DOI=10.1145/2018584.2018599
http://doi.acm.org/10.1145/2018584.2018599

Table of Contents

Publish/Subscribe Event Notification

NDN is Receiver-Driven

- On-demand content delivery
  - Consumer initiate requests
  - Data transmitted in replies
- Good for persistent information

Publish/Subscribe Event Notification

- Producer-driven, push to subscribers
- Tell me when @yoursunny posts a new tweet
    - Text `ON yoursunny` to 40404
- Alert me when it rains in Tucson
    - #ifttt
- Sound the alarm when a sensor detects an intrusion
    - IP multicast

Pub/Sub on top of On-Demand

- Implementing Publish/Subscribe Event Notification on top of On-Demand Content Delivery
    - Conceptually feasible
    - Lots of problems, not the best technical solution

Polling the Producer

- The consumer continually issues interests at regular intervals, and the producer replies with a “null” packet or an event notification.
- Problems
  - States overhead, for only a few effective transmissions
  - Caching cannot be used

Producer-initiated Transmission

- The producer sends an interest that is not intended to return any data, but carries a callback prefix or the notification itself
- Problems
  - States overhead
  - Overloaded use of interests as notifications

Long-Lived Interests

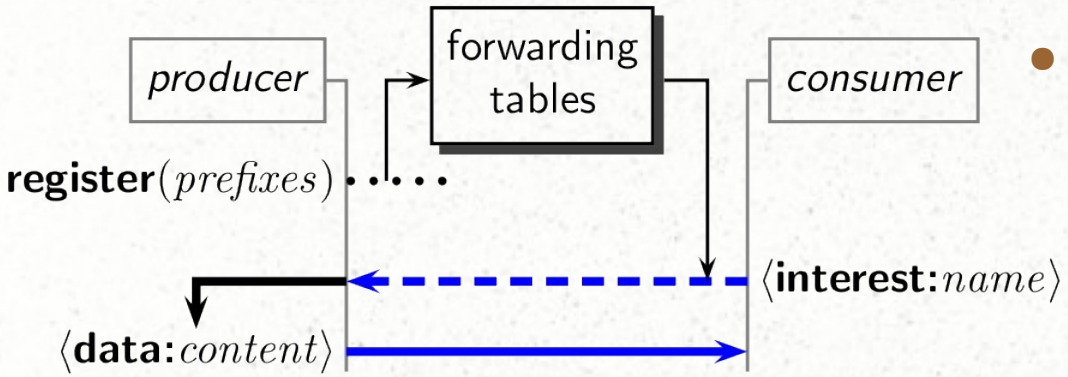
- Keep interests in producer for a long time, reply when there is a notification
    - Long HTTP connection in WebIM
- Problems
    - Lock valuable PIT entries for a long time
    - Events between last reply and new interest are lost

They are Different Enough

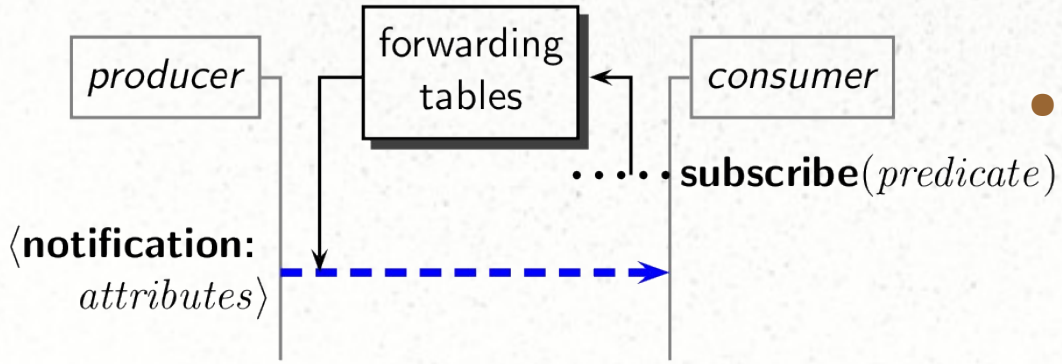
- It makes little sense to implement one on top of the other
- Each requires some level of specialized support in an underlying network fabric

Unified Content-based Network Layer

They have commonality

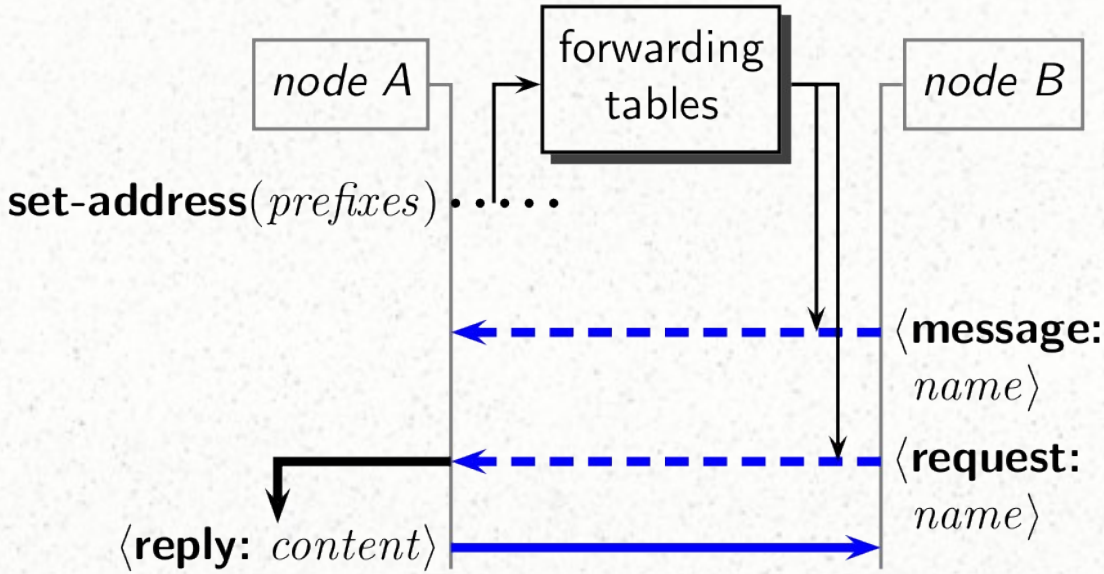


- Interest: goes to prefix of producer



- Event notification: goes to predicate of consumer

Unified Content-based Network Layer



	Interests	Event Notifications
Source of routing information	producers	consumers
Expecting replies	yes	no
Caching semantics	can be satisfied by cached content	must be forwarded to all consumers

Node Interface and Packet Formats

address advertisement
node
prefixes ...

message
forwarding information ...
name
opaque content ...

request
forwarding information ...
request ID
name
source/fork node
segment/byte-range

reply
request ID
name
destination/fork node
segment/byte-range
data ...

Forwarding Messages and Requests

- Forwarding is controlled by prefixes + policies
- Forwarding strategy
    - Compare names against prefixes at each hop
    - Source routing
- Both messages and requests can be forwarded using exactly the same scheme

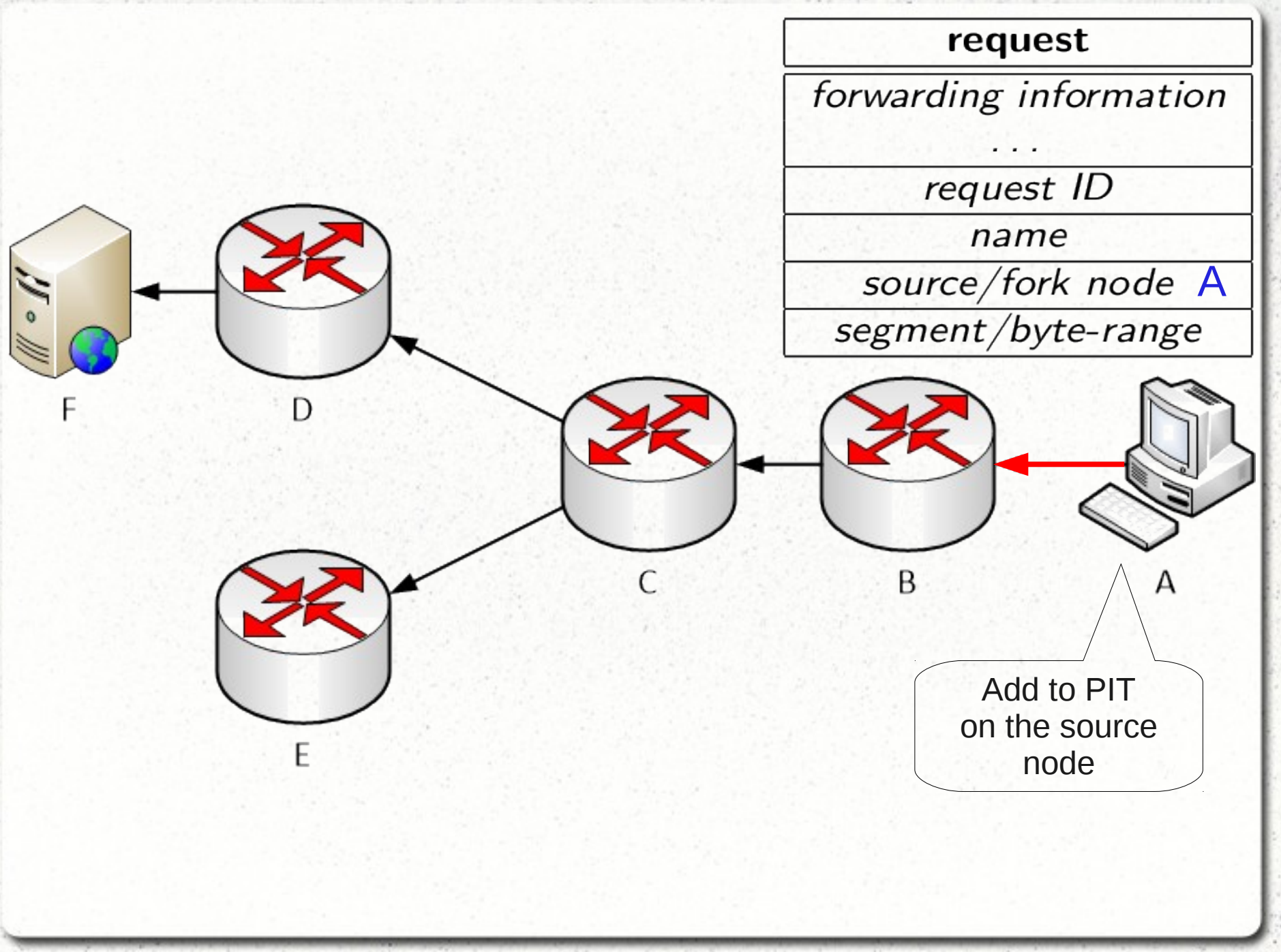
Handling Replies

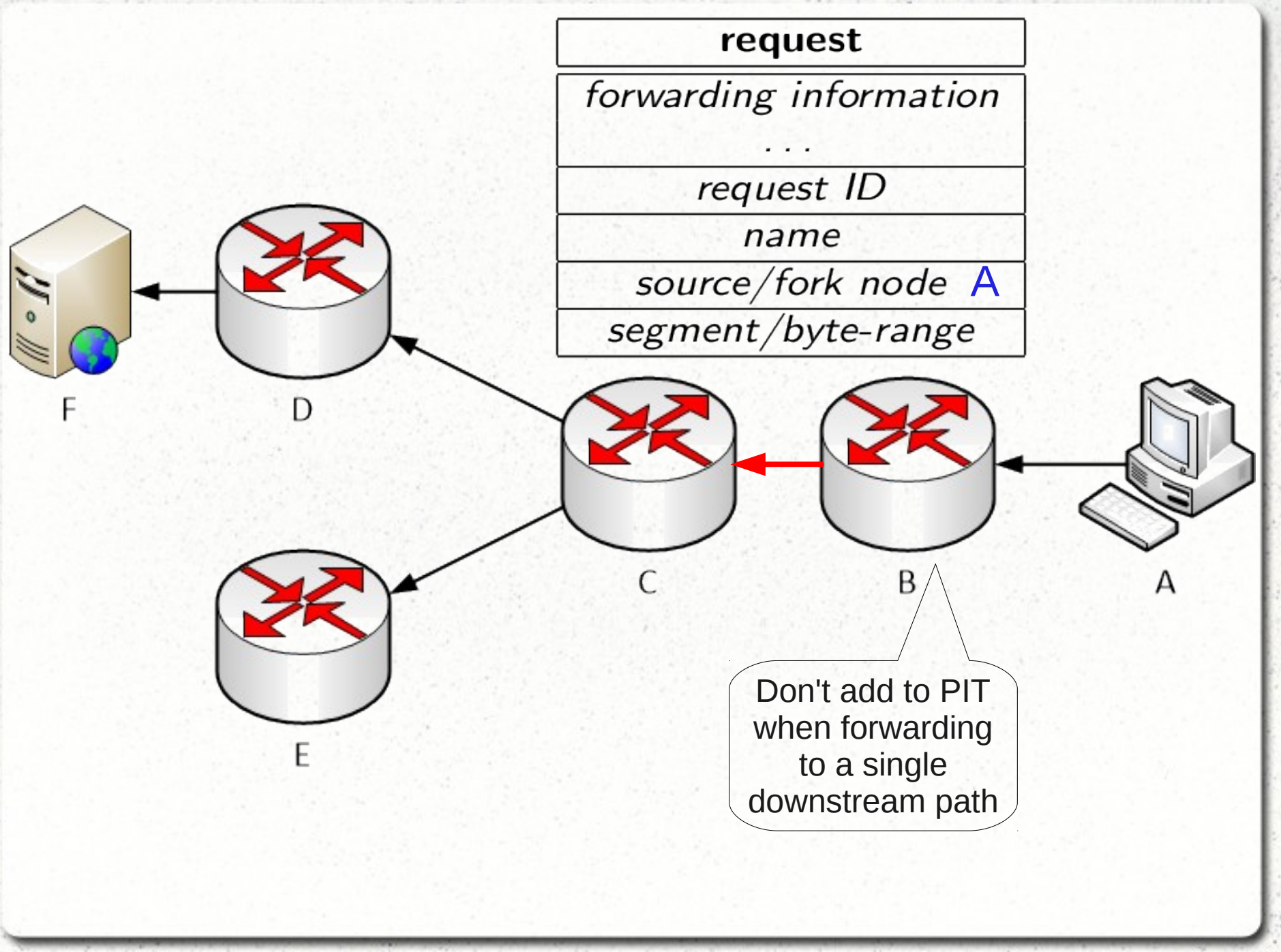
- Flow backward toward consumers
    - Soft state (PIT) is still needed
- Negative Replies
    - “No such data exist on this path”
- How to reduce the space overhead of PIT?

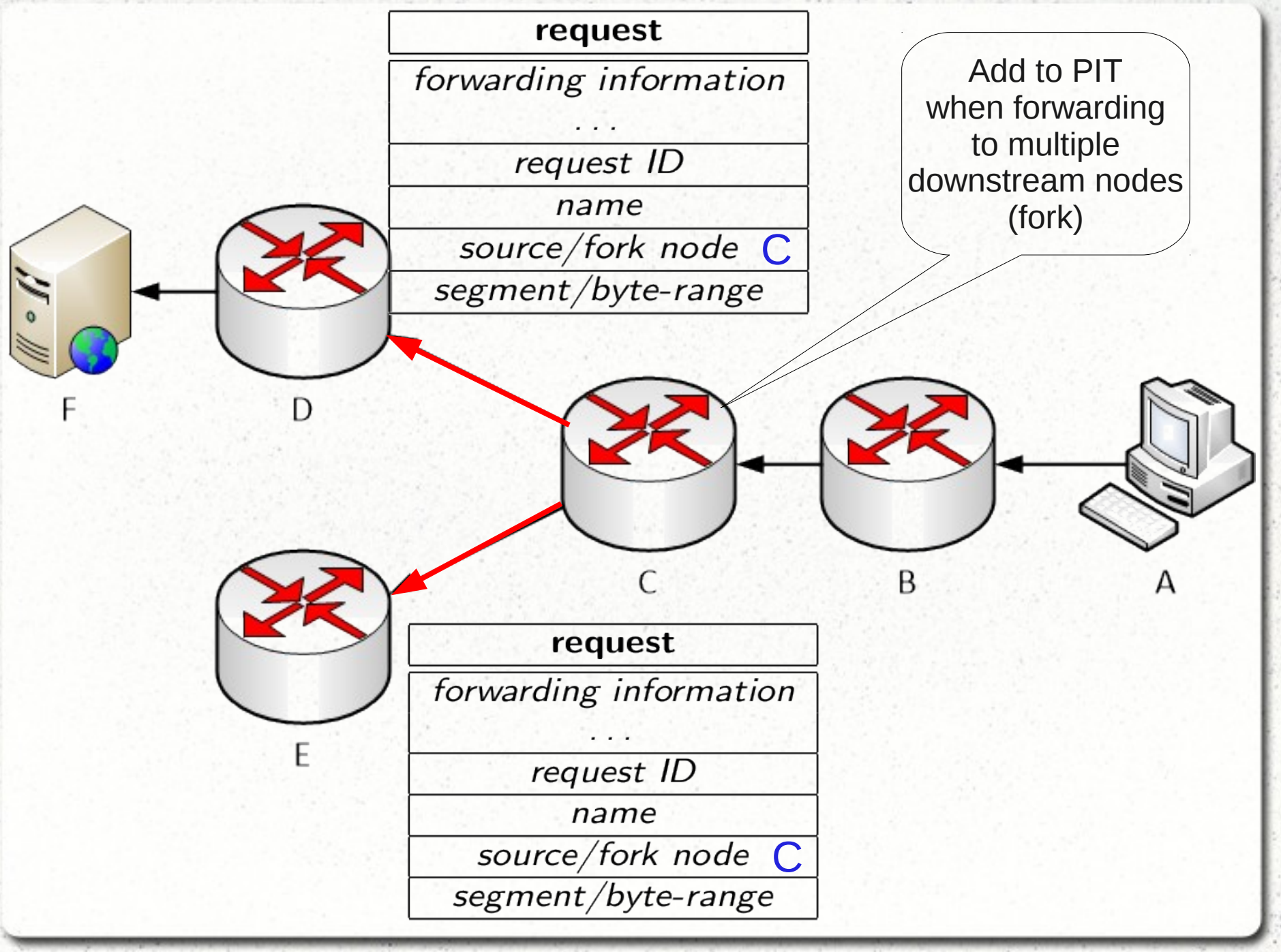
The New Node Model

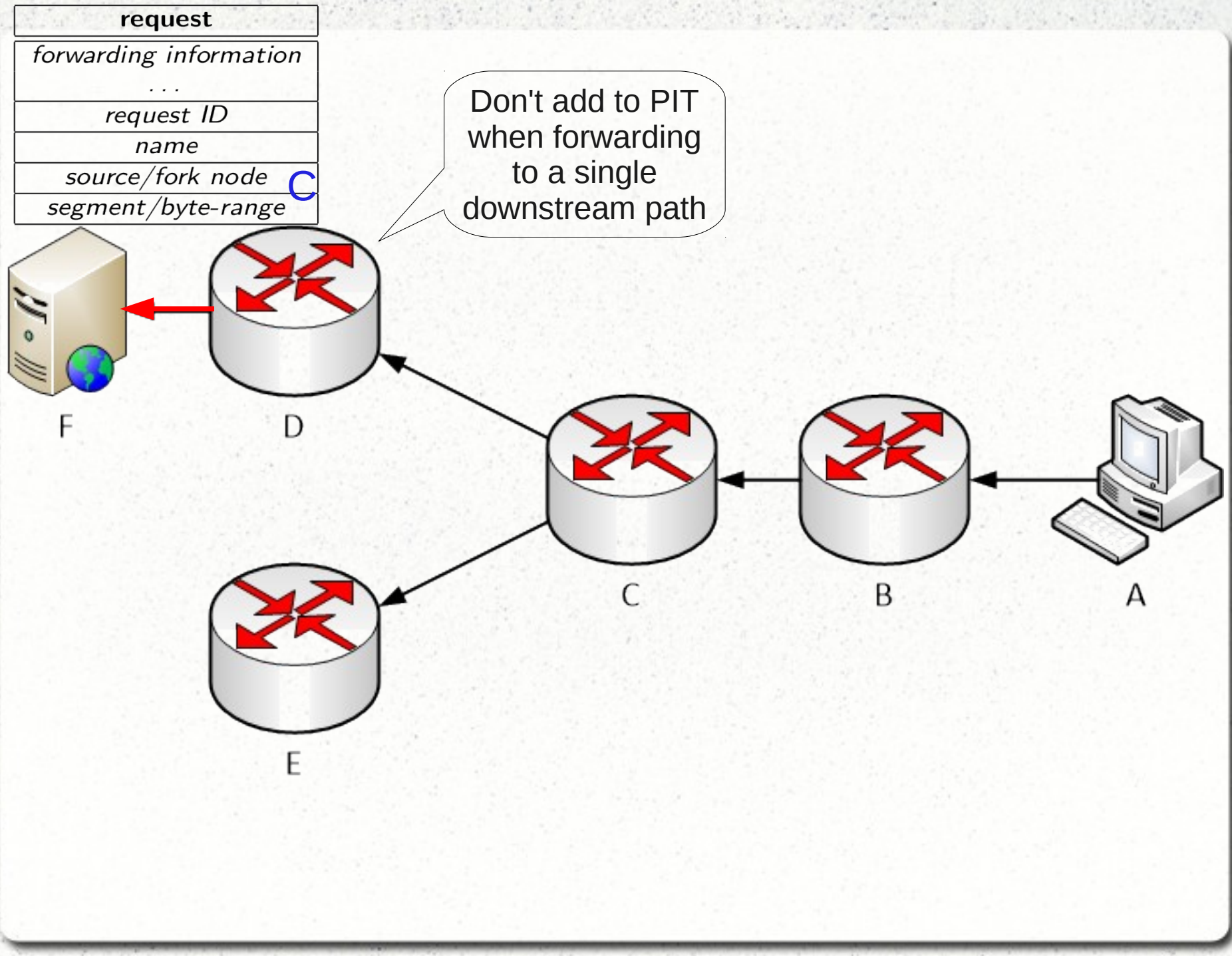
The New Node Model

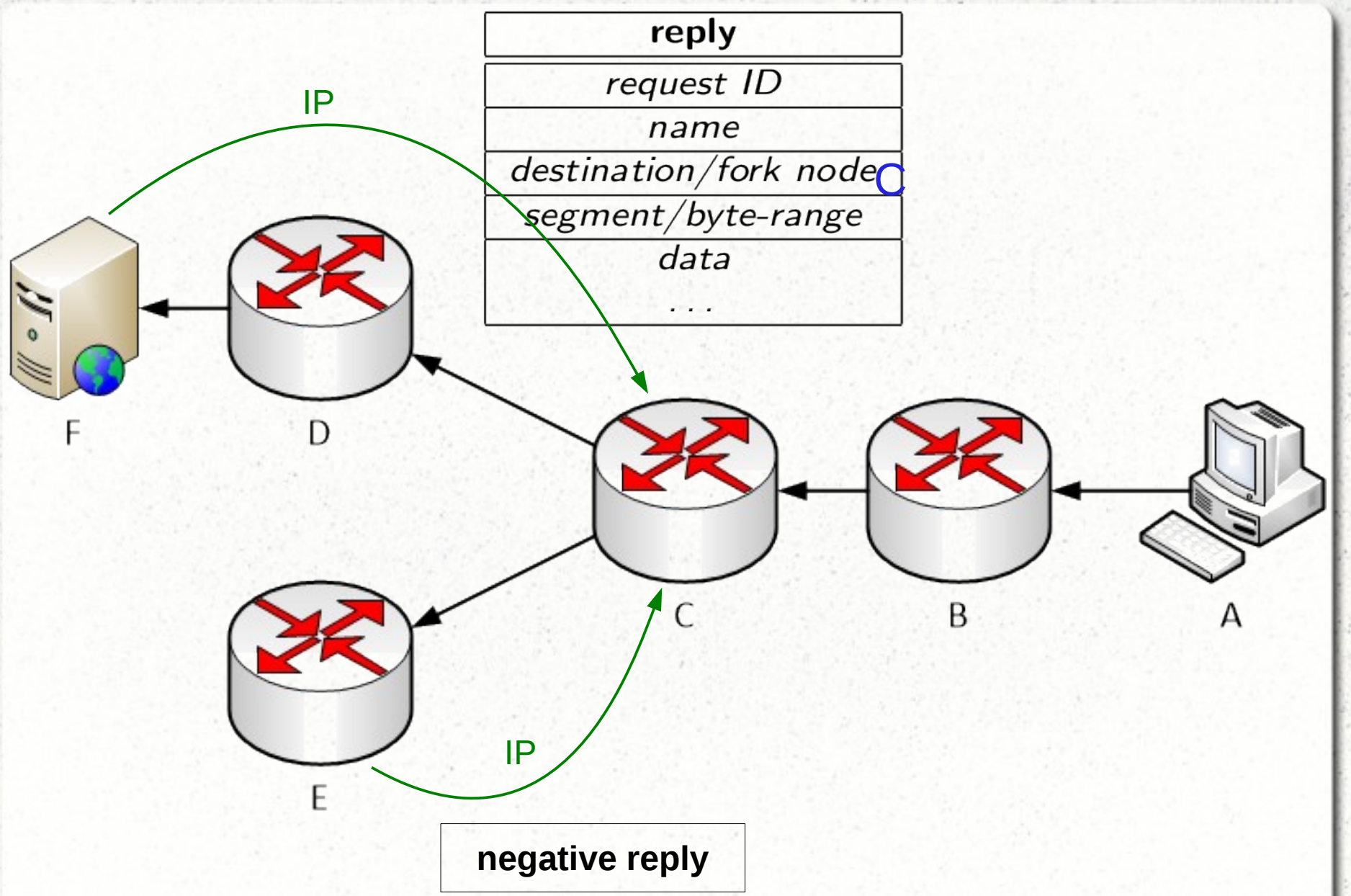
- Create a PIT entry only at the source node of the request, and wherever a request is duplicated over two or more downstream paths (fork)
- Send replies upstream using standard IP forwarding

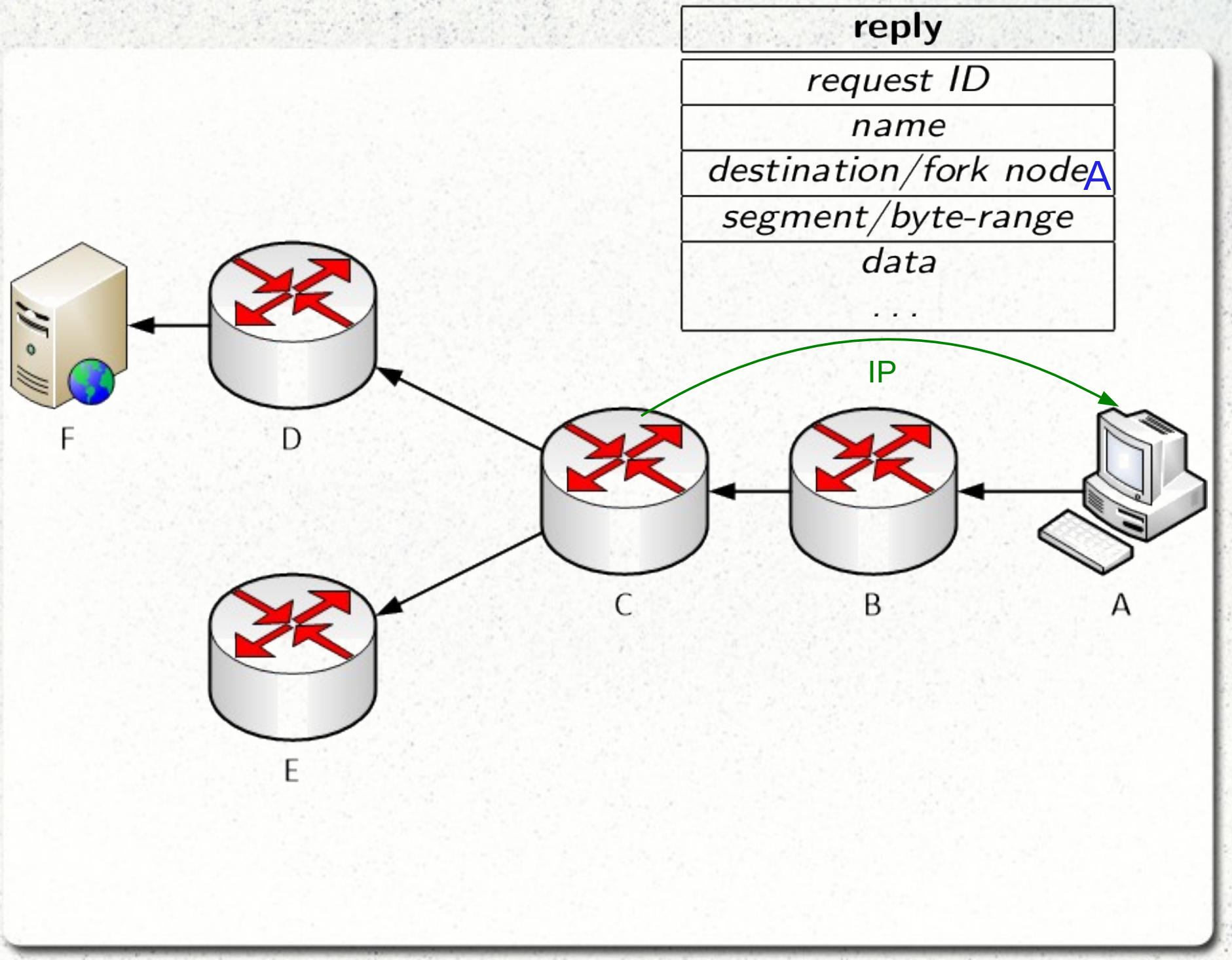

request
forwarding information
. . .
request ID
name
source/fork node A
segment/byte-range
F
D
C
B
A
E
Add to PIT
on the source
node


request
forwarding information
. . .
request ID
name
source/fork node A
segment/byte-range
F
D
C
B
A
E
Don't add to PIT when forwarding to a single downstream path


request
forwarding information
. . .
request ID
name
source/fork node C
segment/byte-range
Add to PIT
when forwarding
to multiple
downstream nodes
(fork)
F
D
C
B
A
E
request
forwarding information
. . .
request ID
name
source/fork node C
segment/byte-range


request
forwarding information
...
request ID
name
source/fork node
segment/byte-range
C
Don't add to PIT when forwarding to a single downstream path
F
D
C
B
A
E


reply
request ID
name
destination/fork node
segment/byte-range
data
. . .
IP
F
D
C
B
A
E
IP
negative reply


reply
request ID
name
destination/fork node A
segment/byte-range
data
. . .
IP
F
D
C
B
A
E

Is it Evil?

- ICN relies on IP
- Same content may traverse a link multiple times
- Forwarding strategy is limited