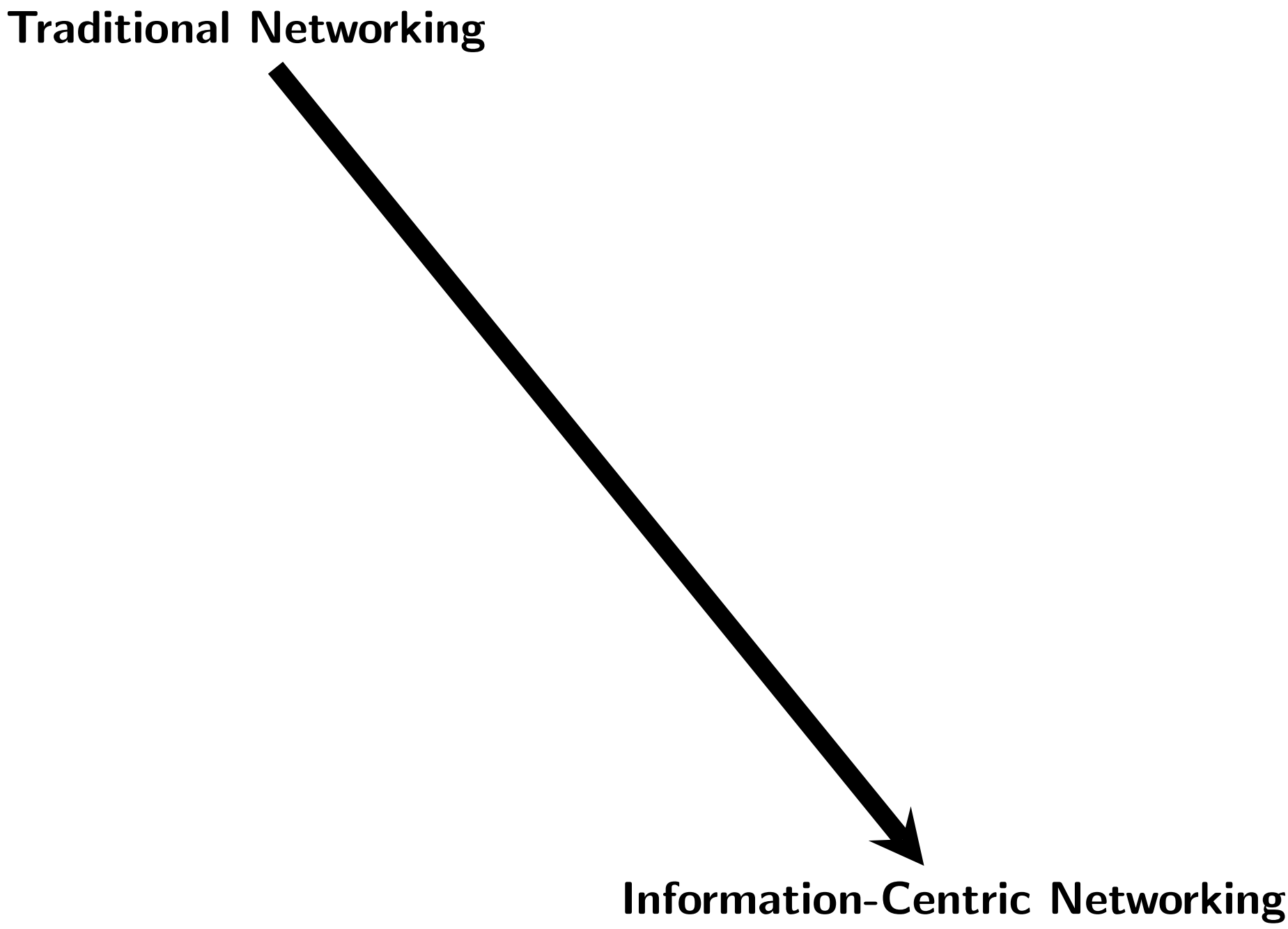
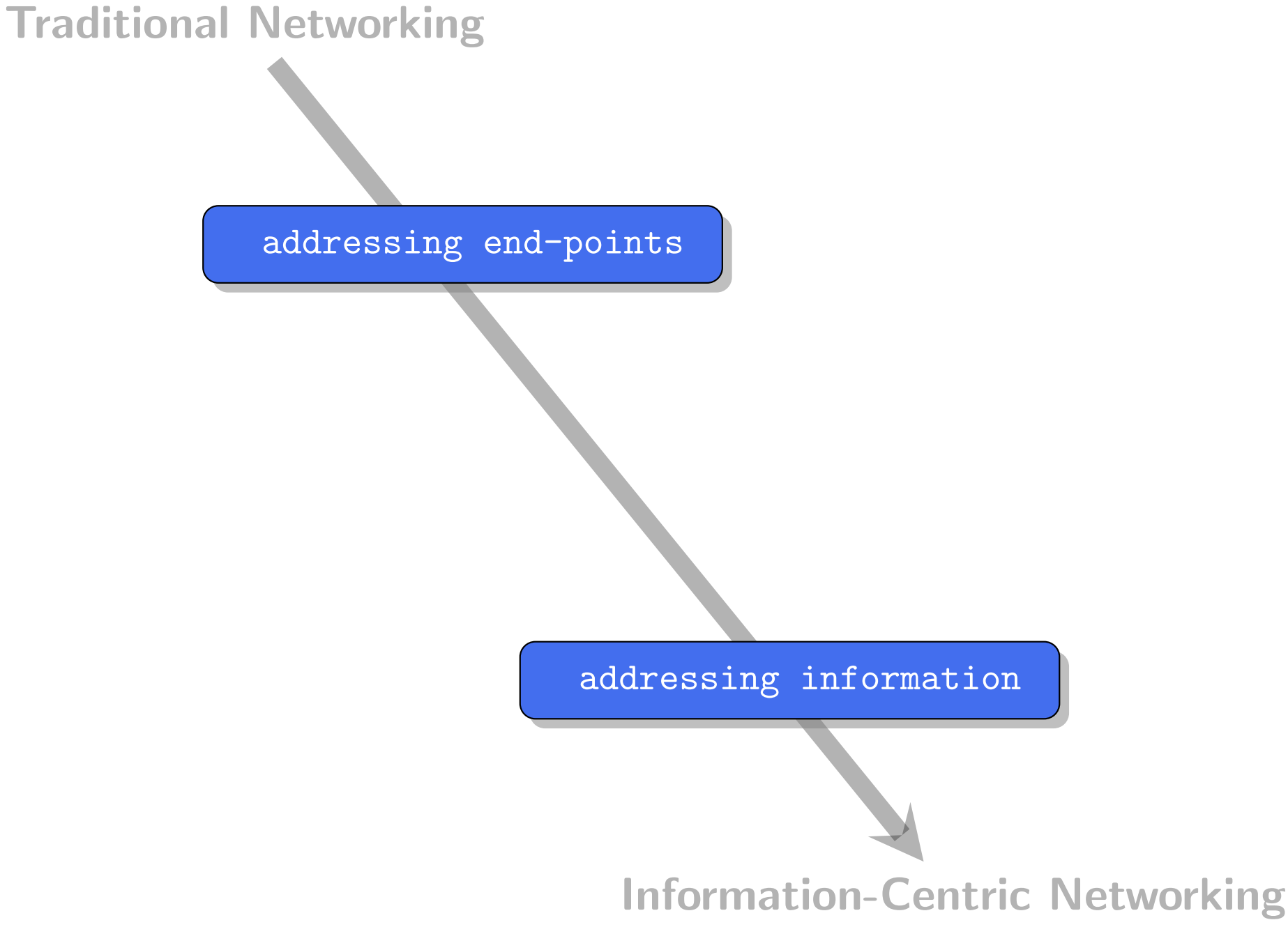
Is Information-Centric Multi-Tree Routing Feasible?

ICN workshop 2013

Michele Papalini (University of Lugano)

joint work with:
Antonio Carzaniga (University of Lugano)
Koorosh Khazaei (University of Lugano)
Alexander L. Wolf (Imperial College London)


Traditional Networking
Information-Centric Networking


Traditional Networking
addressing end-points
addressing information
Information-Centric Networking

Traditional Networking

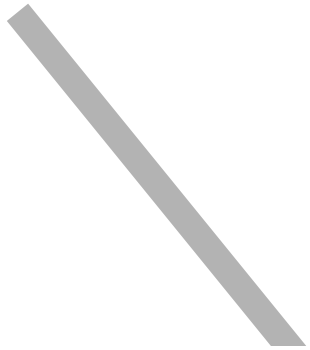
How do we address information?

How do we obtain information?

(Architectural questions)

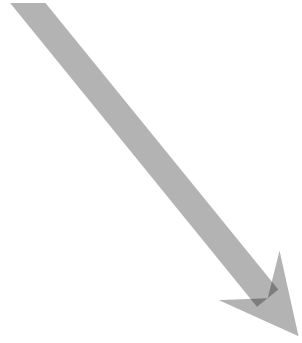
Information-Centric Networking

Traditional Networking

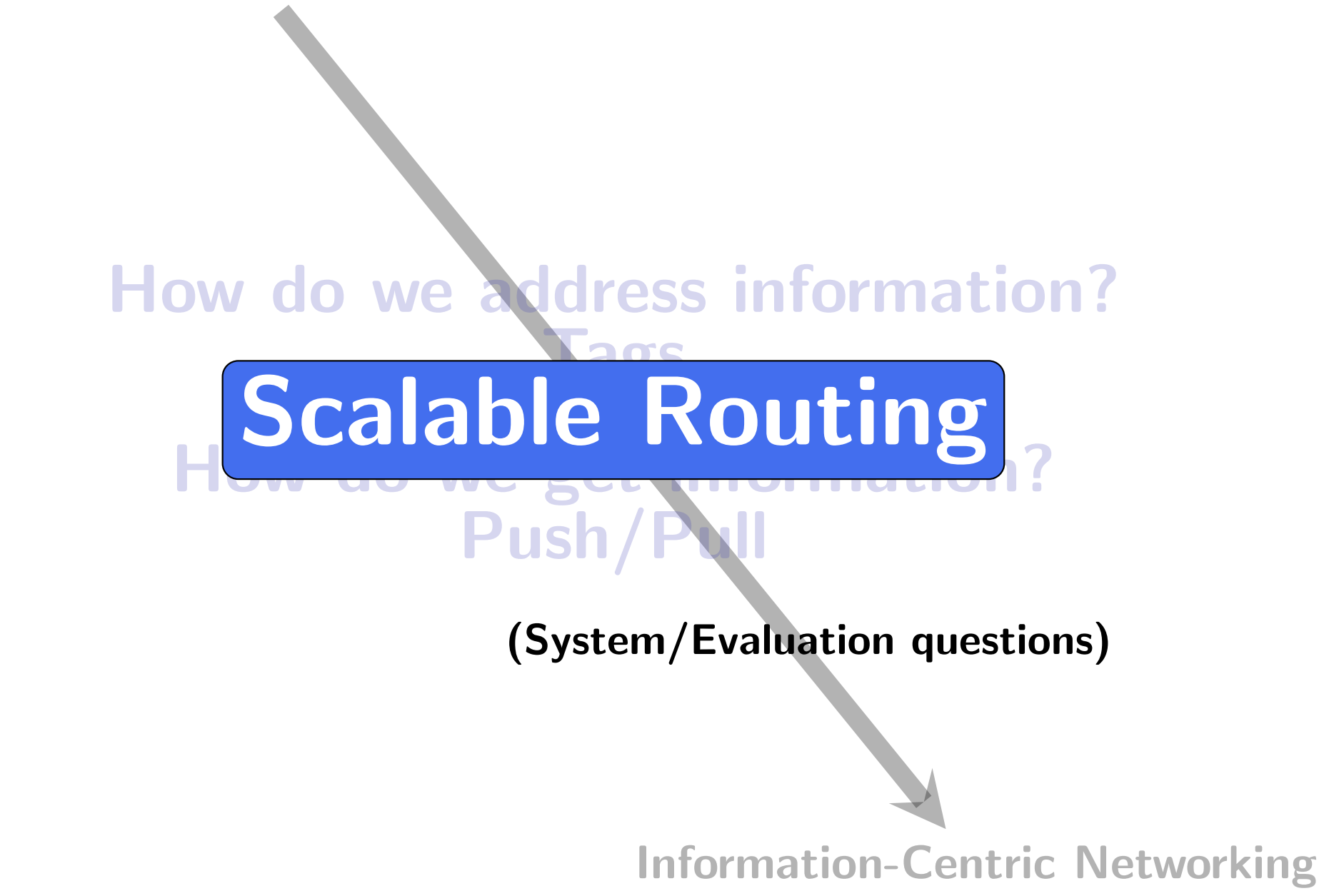


How do we address information?
Tags

How do we obtain information?
Push/Pull



Information-Centric Networking


Traditional Networking
How do we address information?
Tags
Scalable Routing
How do we get information?
Push/Pull
(System/Evaluation questions)
Information-Centric Networking

How do we address information?

Solution I: **name** the data

Solution I: **name** the data

- *flat*, not human readable identifiers

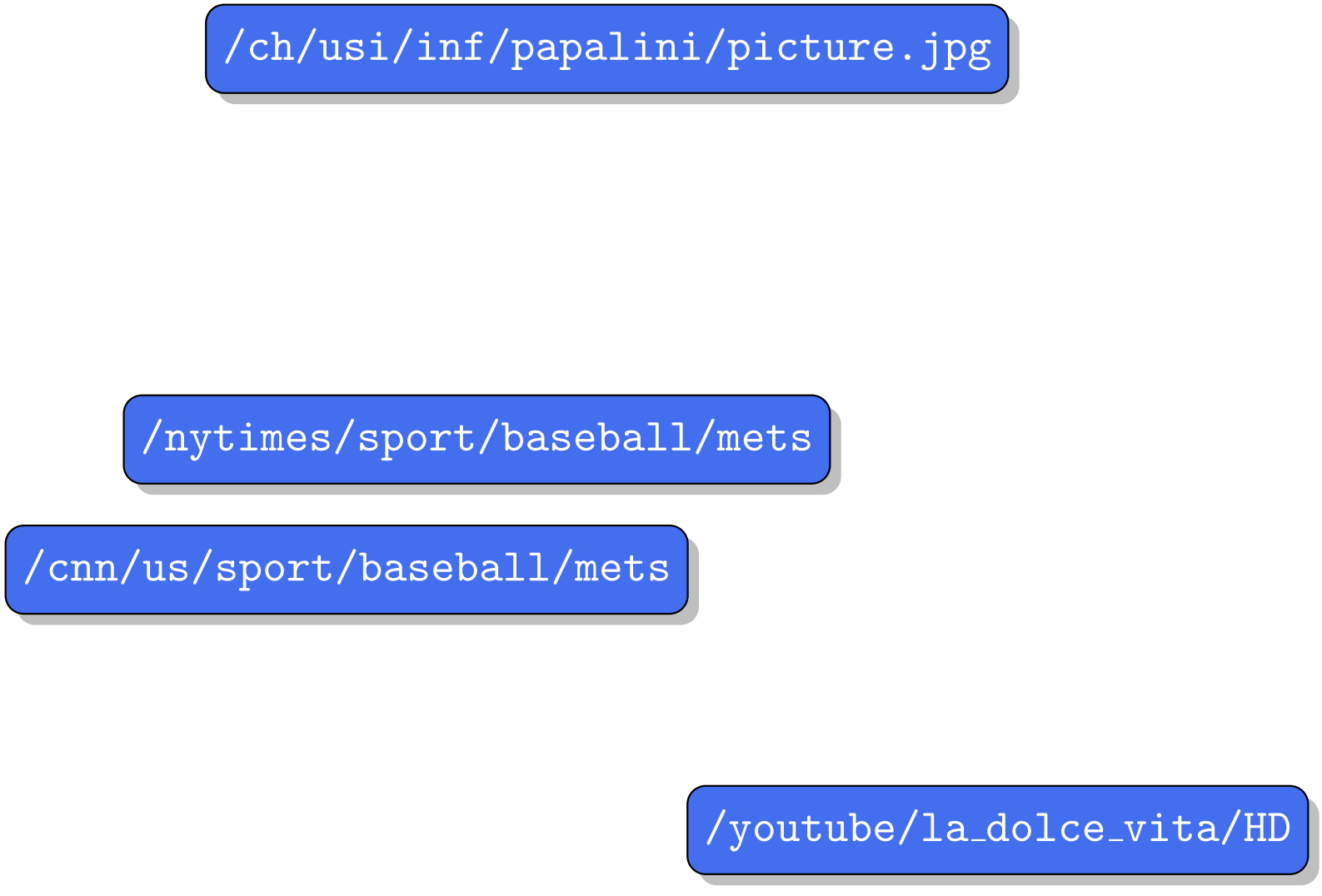
```
1DB76EB8DFD6B0B92A293AADC8421830BDE73CB6
```

- *hierarchical*, meaningful structured names

```
/ch/usi/inf/papalini/picture.jpg
```


/ch/usi/inf/papalini/picture.jpg


/ch/usi/inf/papalini/picture.jpg
/nytimes/sport/baseball/mets
/cnn/us/sport/baseball/mets


/ch/usi/inf/papalini/picture.jpg
/nytimes/sport/baseball/mets
/cnn/us/sport/baseball/mets
/youtube/la_dolce_vita/HD

baseball scores for NY Mets

baseball scores for NY Mets

la dolce vita in HD with english subtitles

Solution II: **describe** the data

Solution II: **describe** the data

- with *set* of **tags**

`baseball, new york, mets`

`la dolce vita, HD, en-sub`

Tags

Tags

- more expressivity

```
/ch/usi/inf/papalini/picture.jpg
```

```
1#ch, 2#usi, 3#inf, 4#papalini, 5#picture.jpg
```

Tags

- more expressivity

`/ch/usi/inf/papalini/picture.jpg`

`1#ch, 2#usi, 3#inf, 4#papalini, 5#picture.jpg`

- more aggregation

`sport, new, football`

`Lugano, sport, activities`

`sport`

Tags

- more expressivity

`/ch/usi/inf/papalini/picture.jpg`

`1#ch, 2#usi, 3#inf, 4#papalini, 5#picture.jpg`

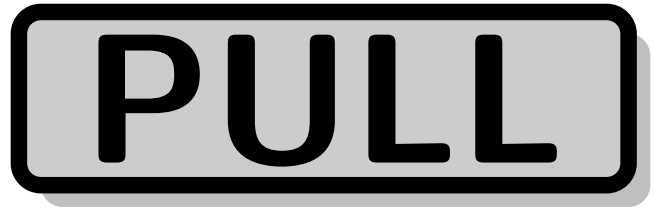
- more aggregation

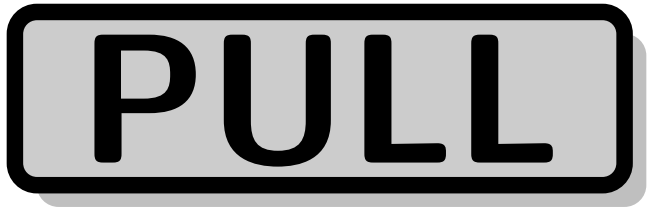
`sport, new, football`

`Lugano, sport, activities` → `sport`

`sport`

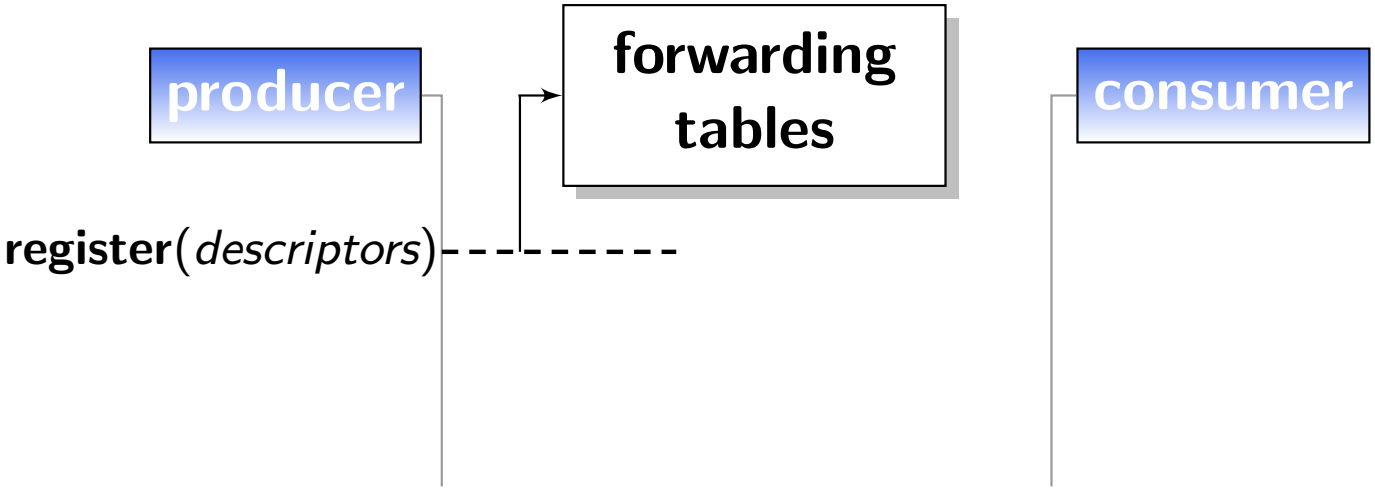
How do we obtain information?


PULL

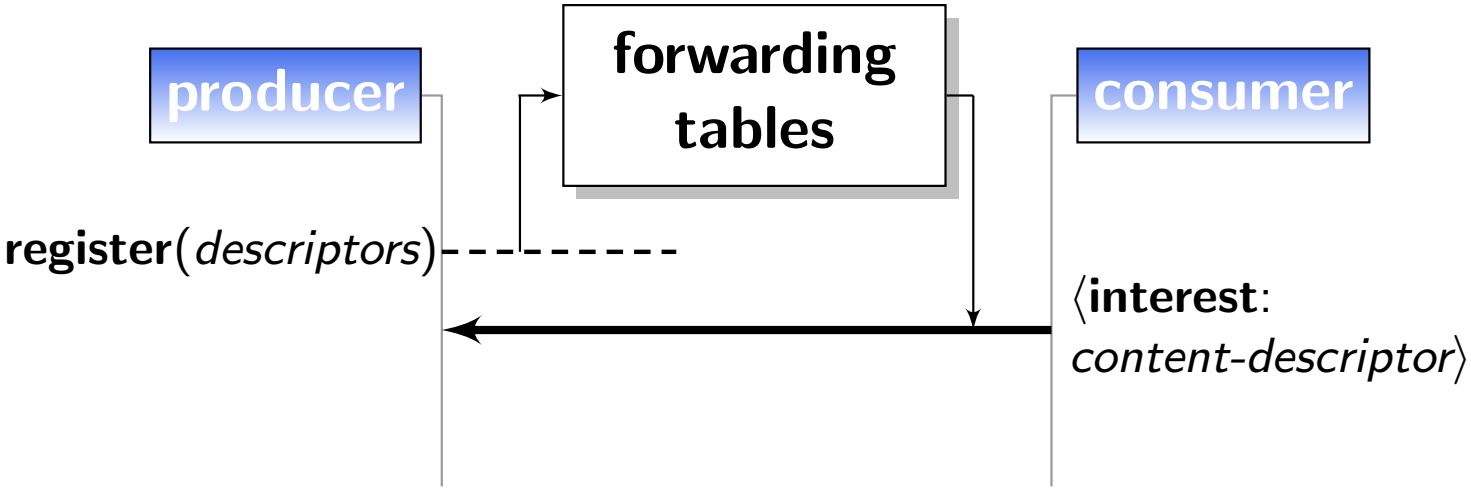

PULL


producer
consumer

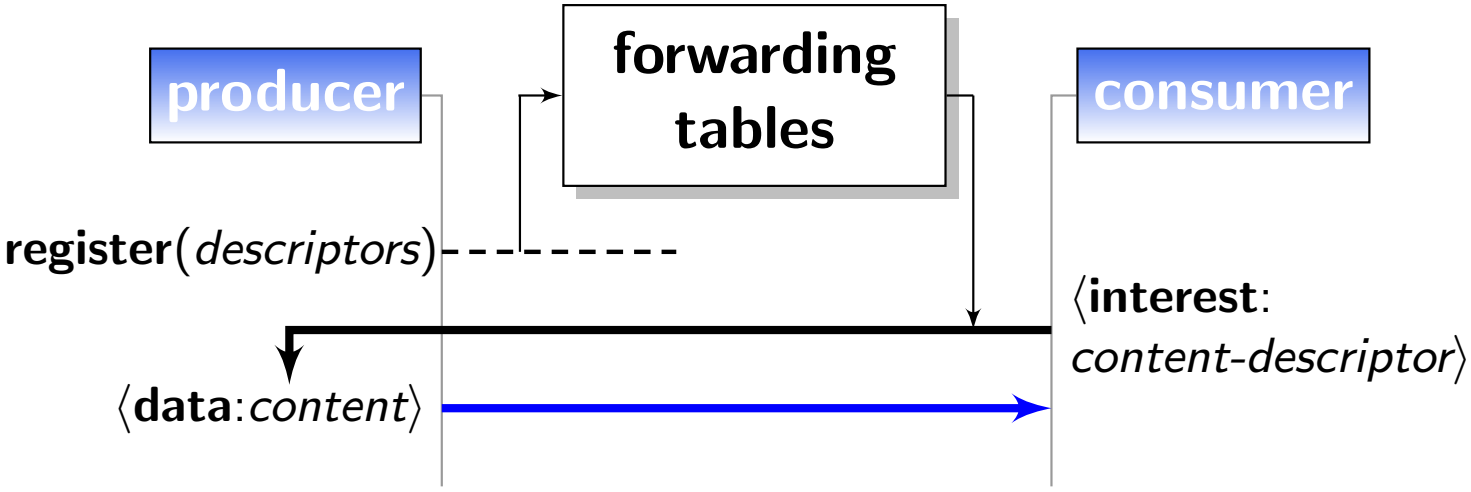
PULL

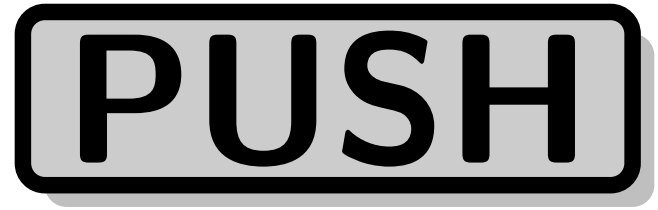

producer
forwarding tables
consumer
register(descriptors)

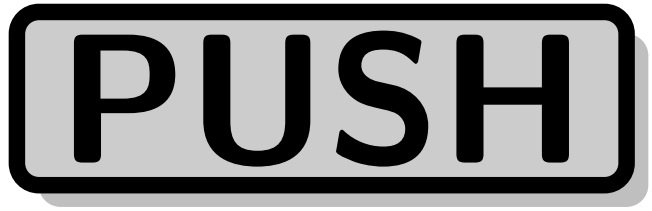
PULL

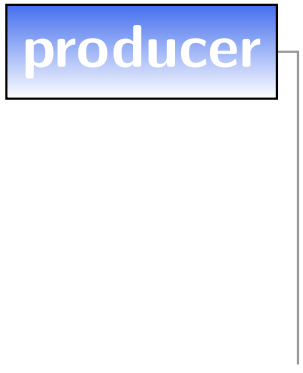

producer
forwarding tables
consumer
register(descriptors)
⟨interest:
content-descriptor⟩

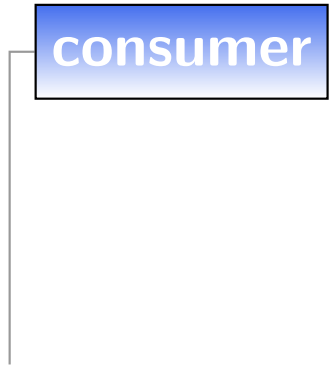
PULL


producer
forwarding tables
consumer
register(descriptors)
⟨interest: content-descriptor⟩
⟨data:content⟩

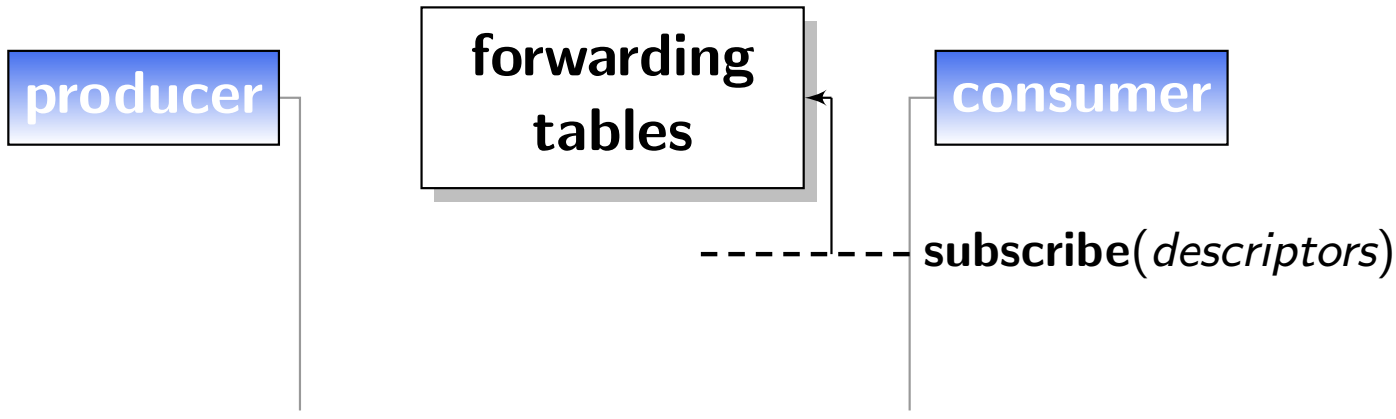

PUSH


PUSH

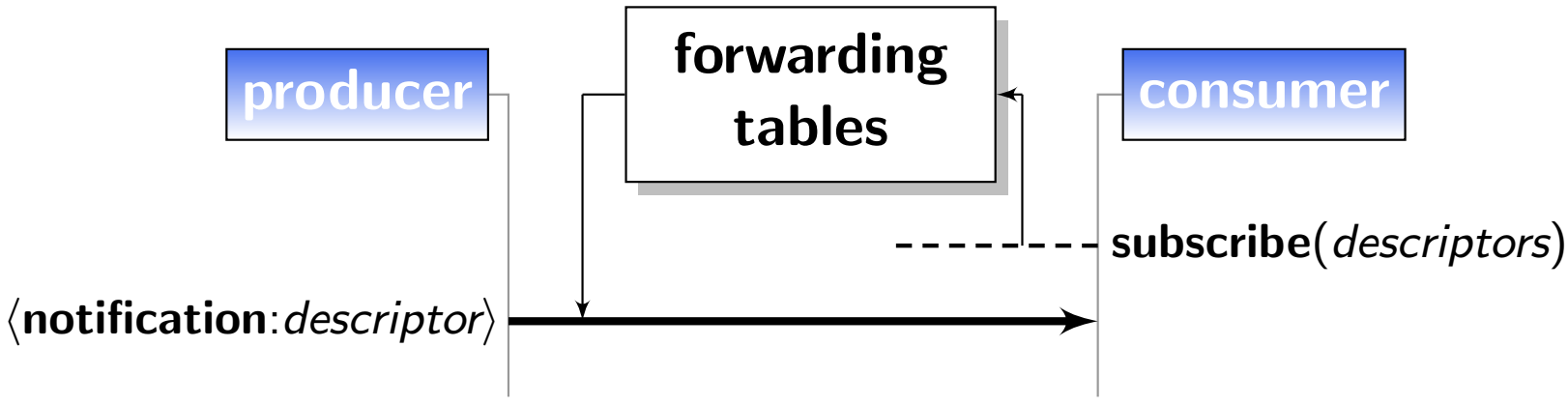

producer

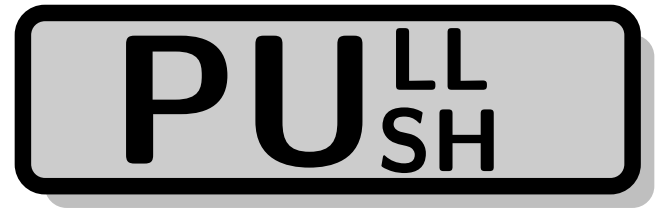

consumer

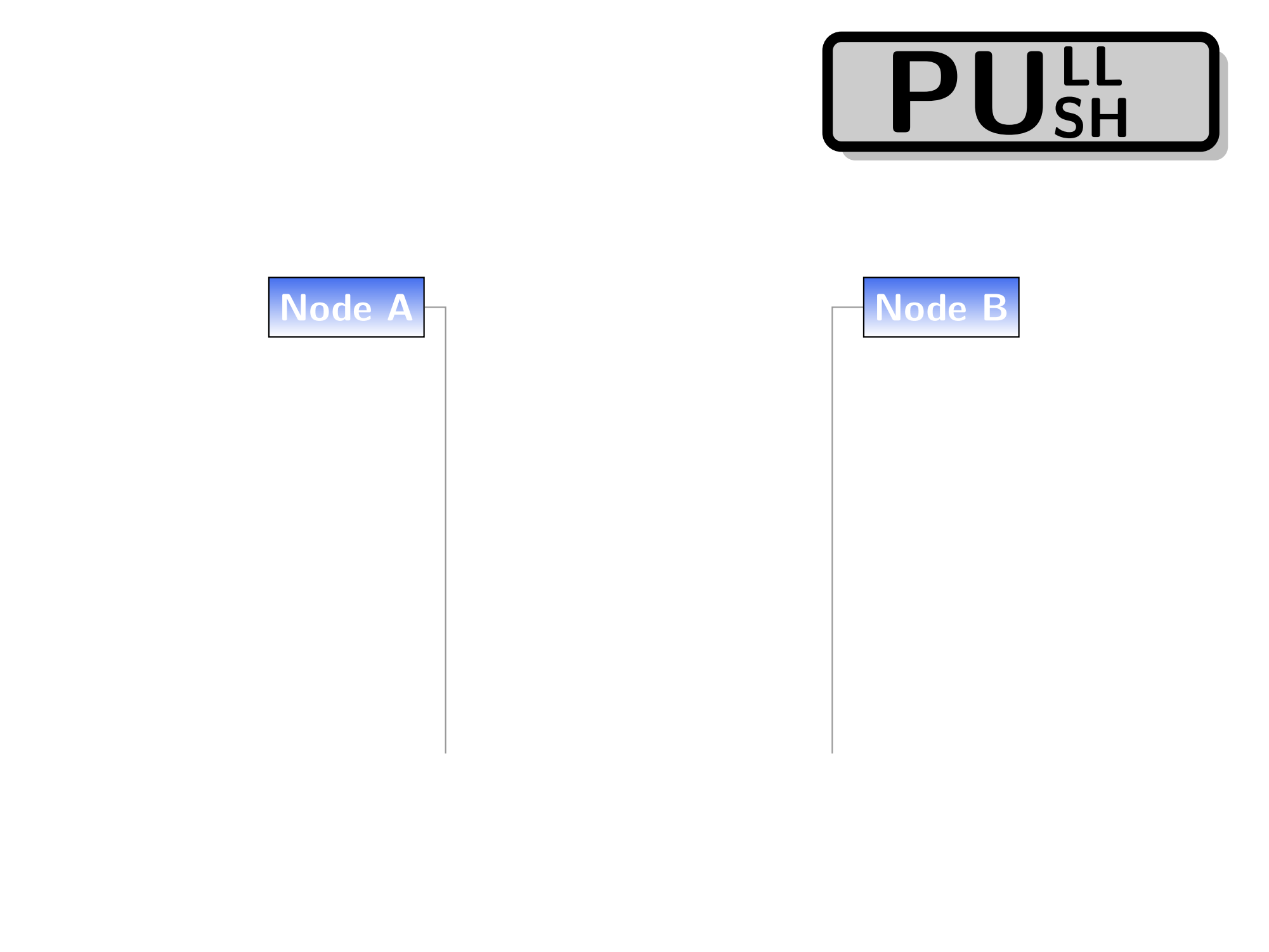
PUSH


producer
forwarding tables
consumer
subscribe(descriptors)

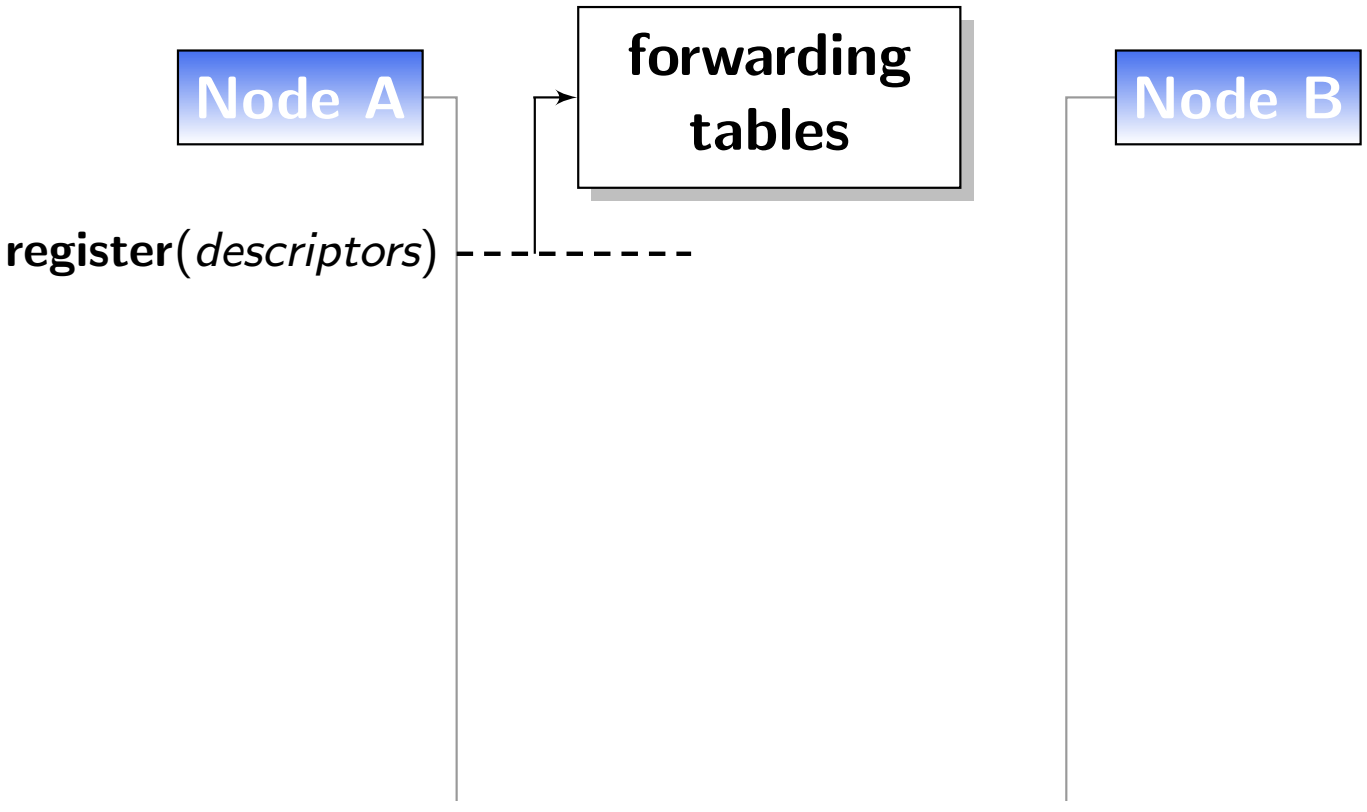
PUSH


producer
forwarding tables
consumer
subscribe(descriptors)
⟨notification:descriptor⟩

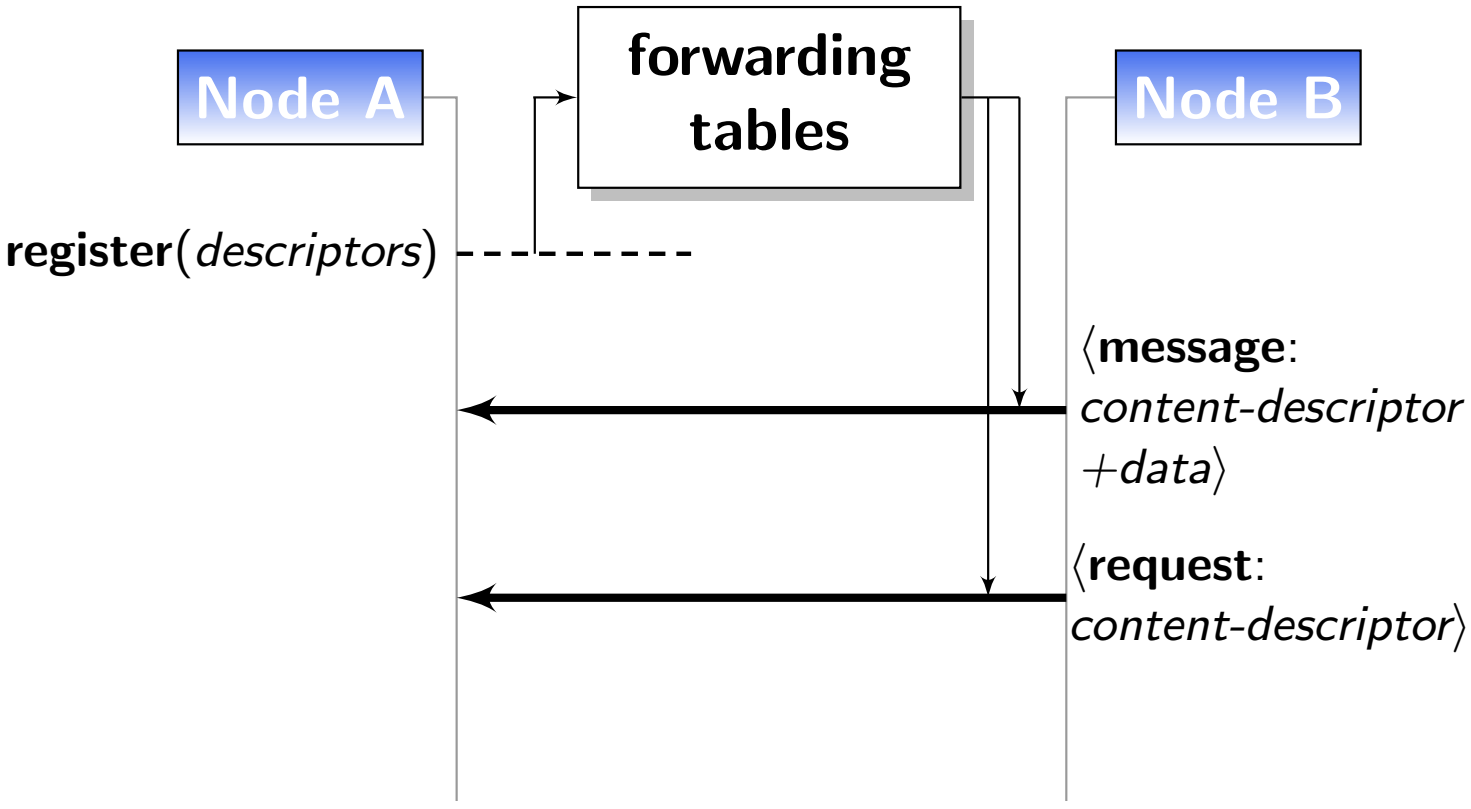

PULL
PUSH


PULL
PUSH
Node A
Node B

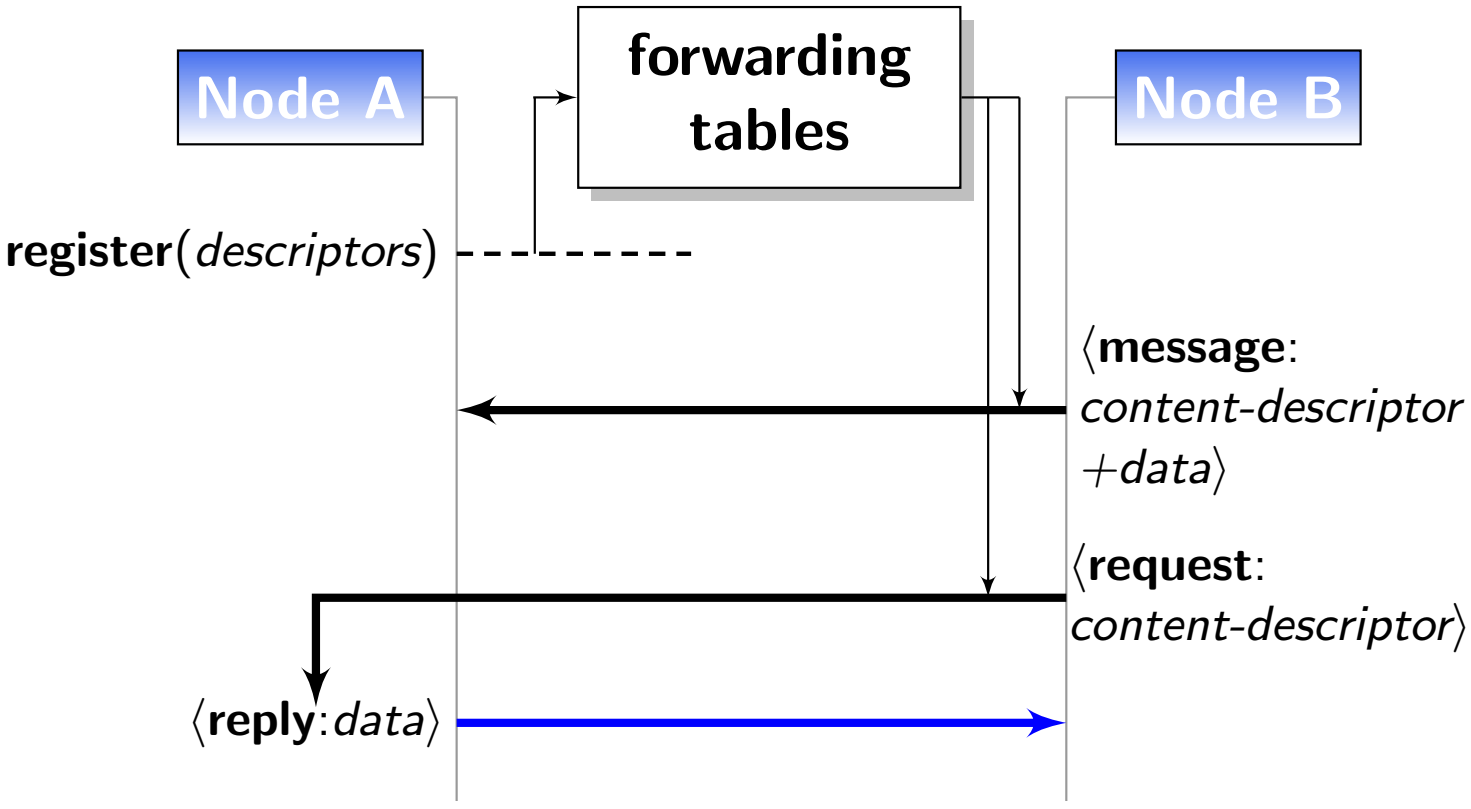
PULL SH

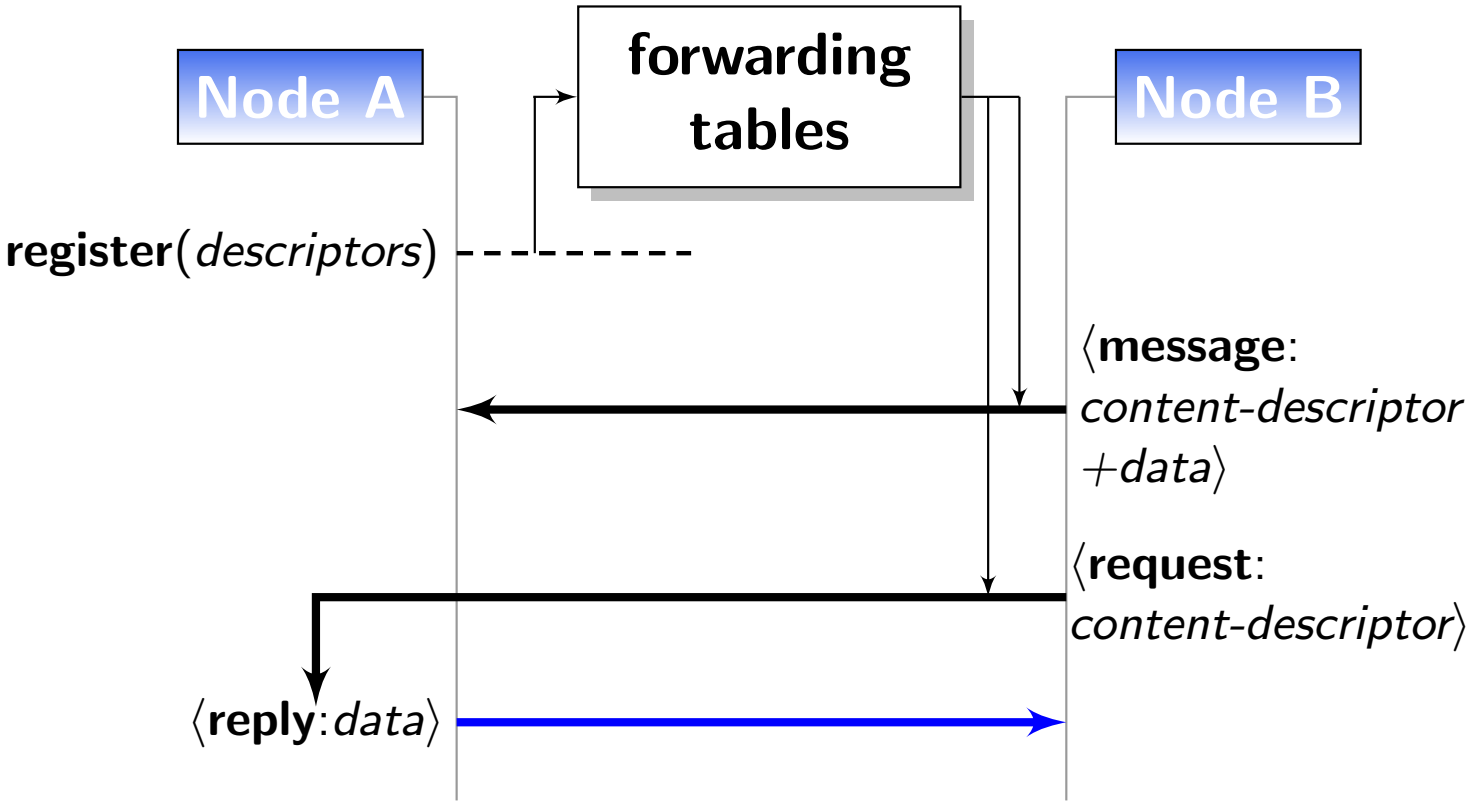


PULL SH

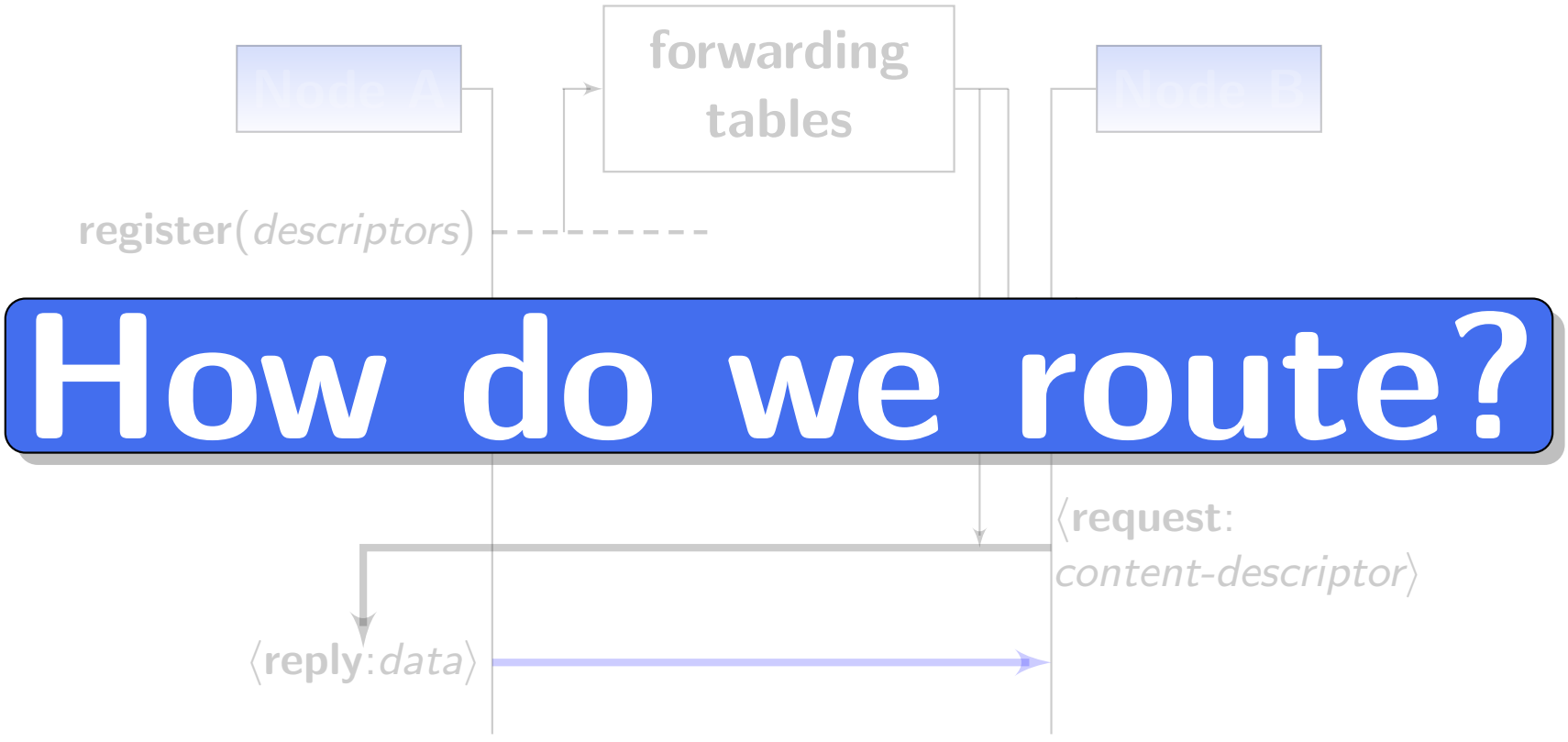

Node A
forwarding tables
Node B
register(descriptors)
⟨message: content-descriptor +data⟩
⟨request: content-descriptor⟩

PULL SH


Node A
forwarding tables
Node B
register(descriptors)
⟨message: content-descriptor +data⟩
⟨request: content-descriptor⟩
⟨reply:data⟩



ICN 2011:
"Content-Based Publish/Subscribe Networking and Information-Centric Networking"


Node A
forwarding tables
Node B
register(descriptors)
How do we route?
⟨request: content-descriptor⟩
⟨reply:data⟩

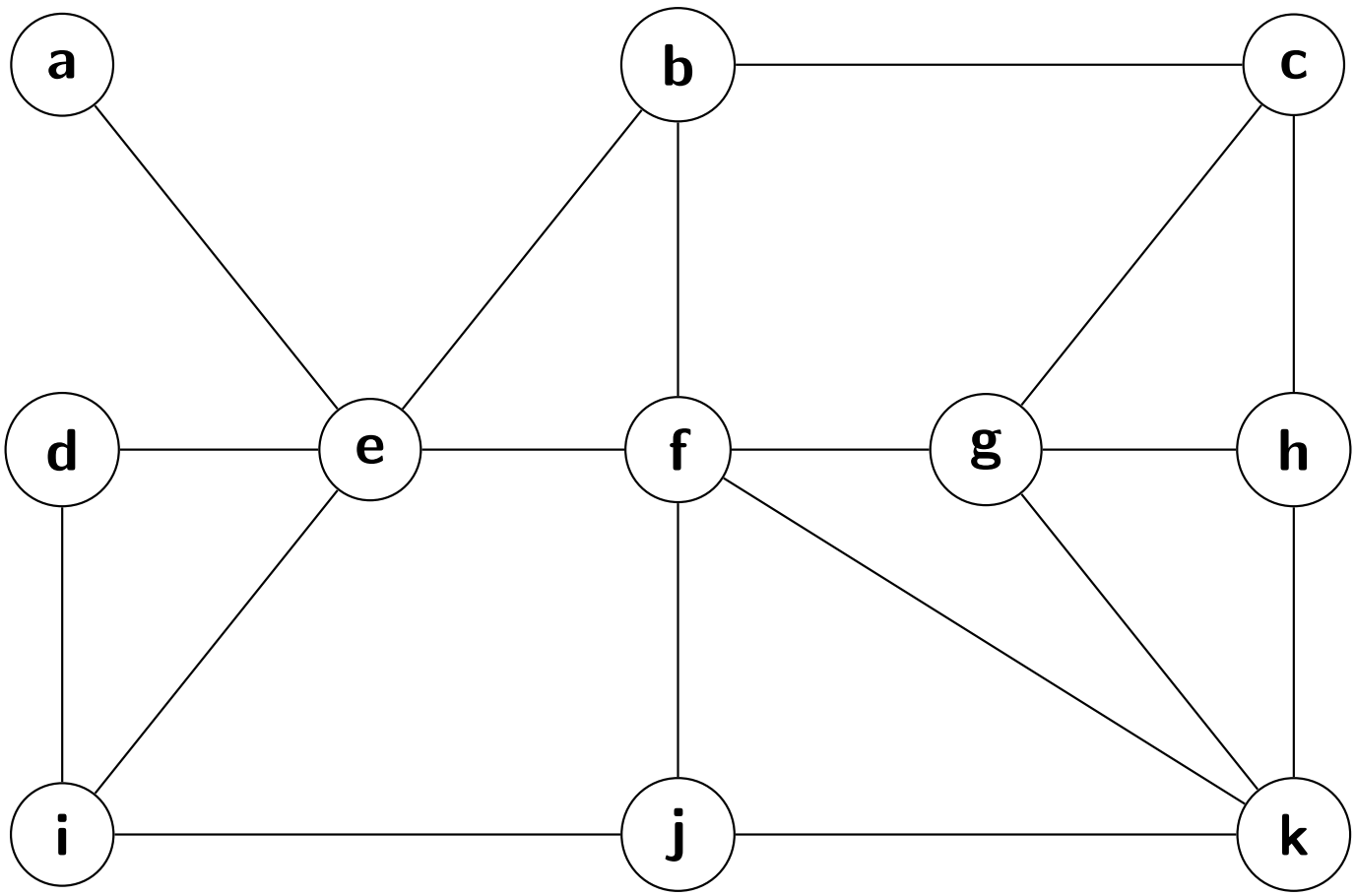
Main contribution

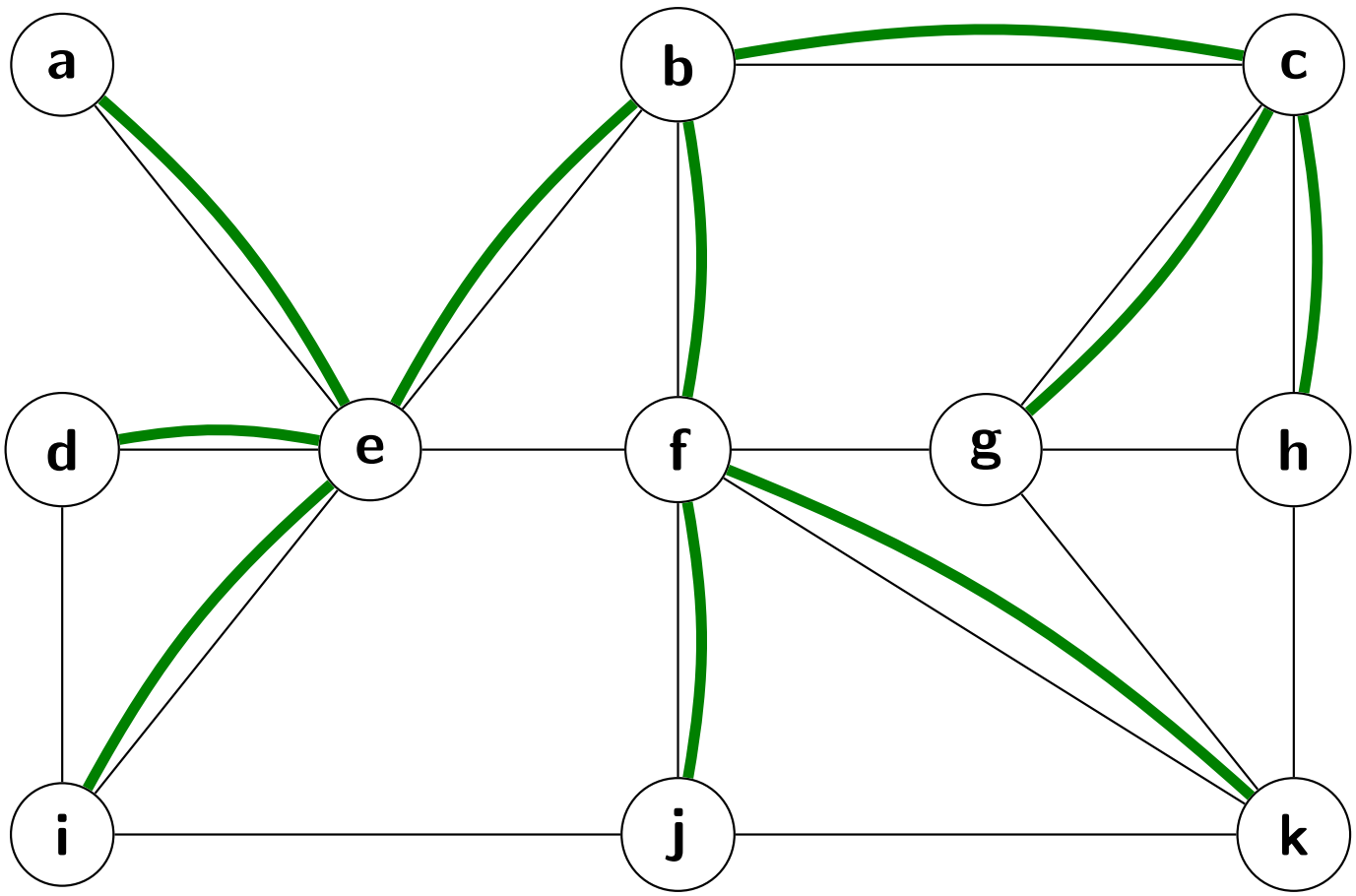
- Routing schema based on multiple trees

Main contribution

- Routing schema based on multiple trees
- Scalability analysis
  - Additional cost (hops/state) using multiple trees
  - FIBs size using a realistic workload

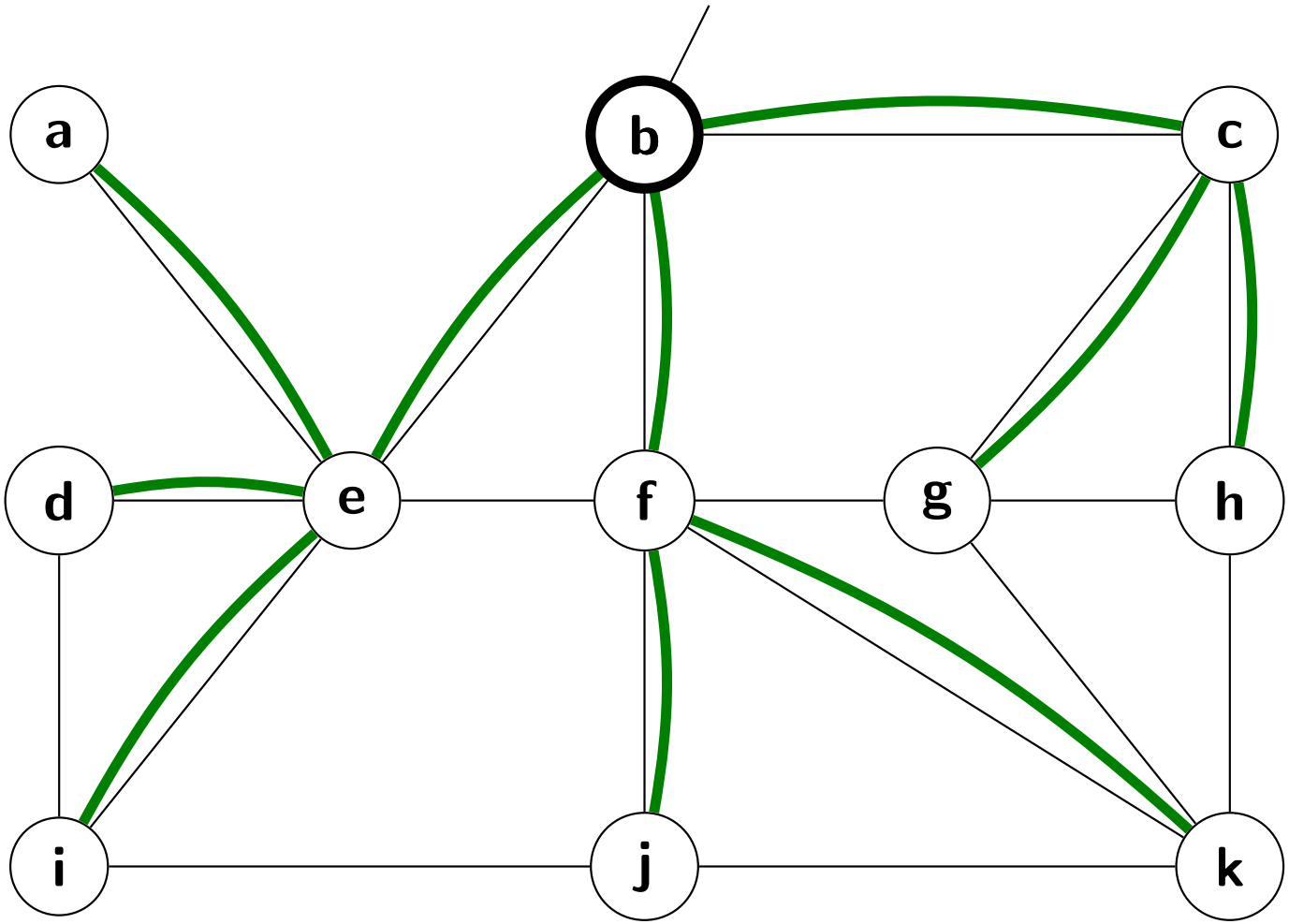
Routing


a
b
c
d
e
f
g
h
i
j
k


a
b
c
d
e
f
g
h
i
j
k

router b: (FIB)

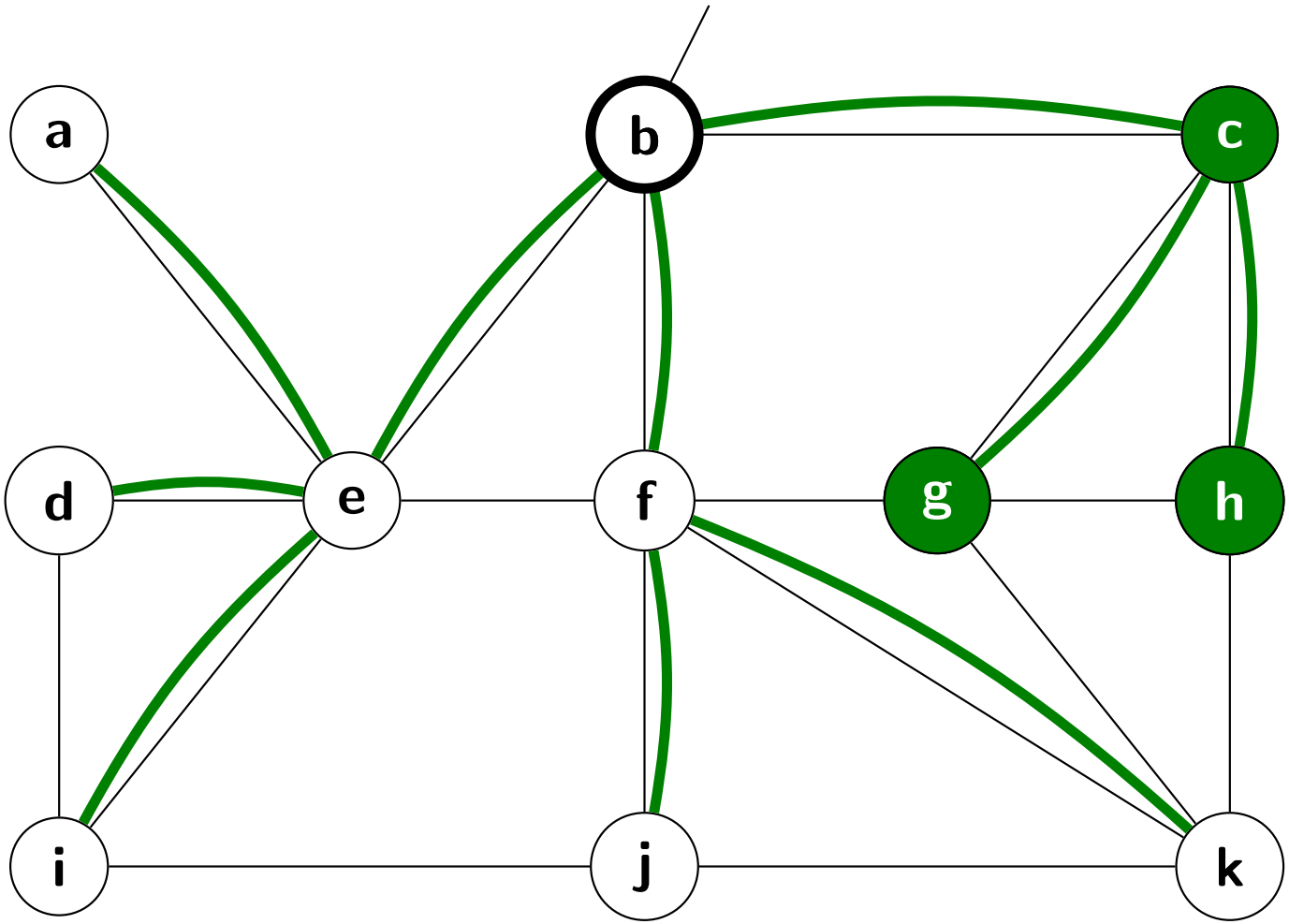
next-hop $w \rightarrow$	**predicate** P_w
$c \mapsto$	$p_c \vee p_g \vee p_h$
$f \mapsto$	$p_f \vee p_j \vee p_k$
$e \mapsto$	$p_a \vee p_d \vee p_e \vee p_i$

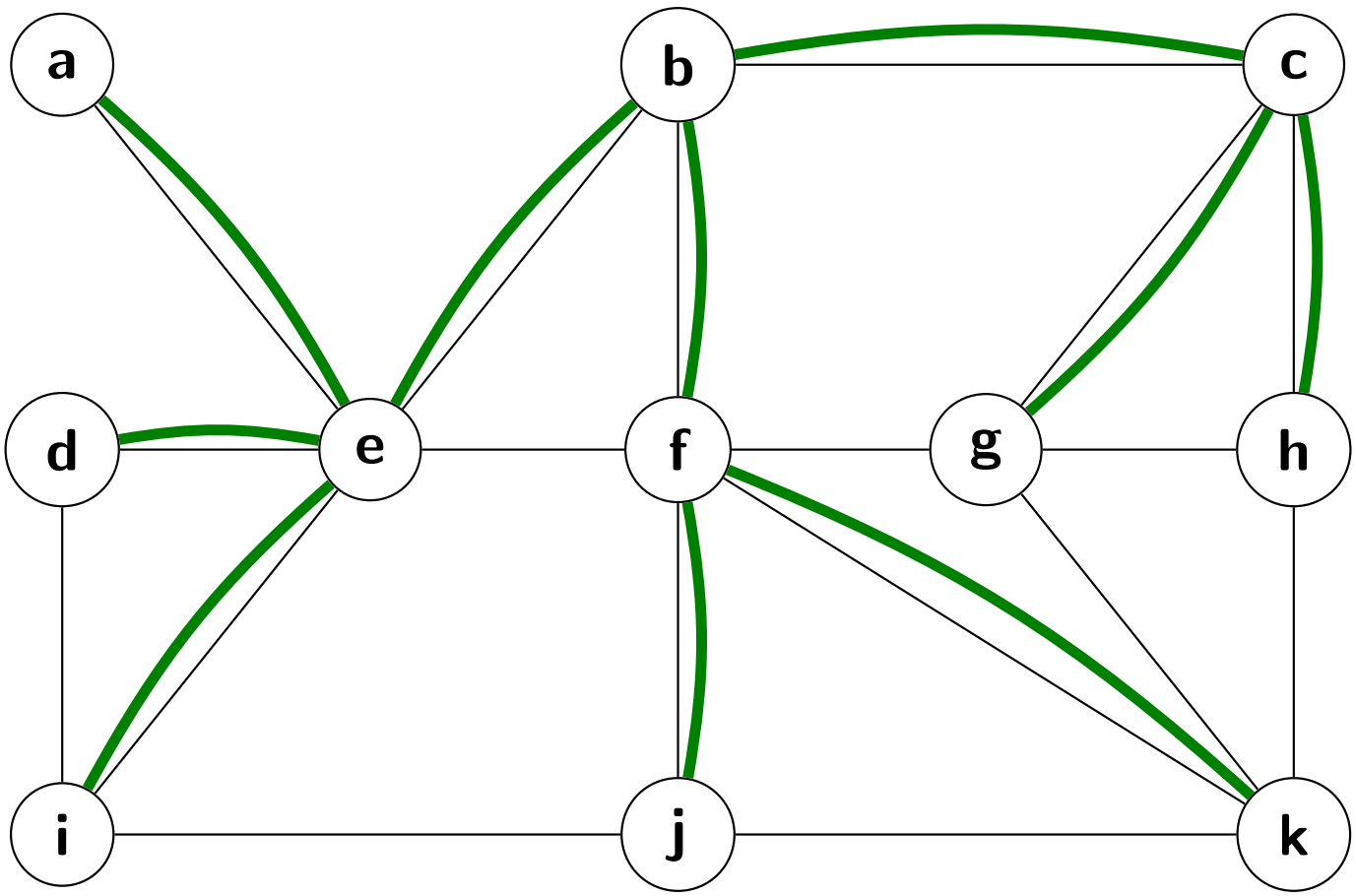


router b: **next-hop** $w \rightarrow$ **predicate** P_w

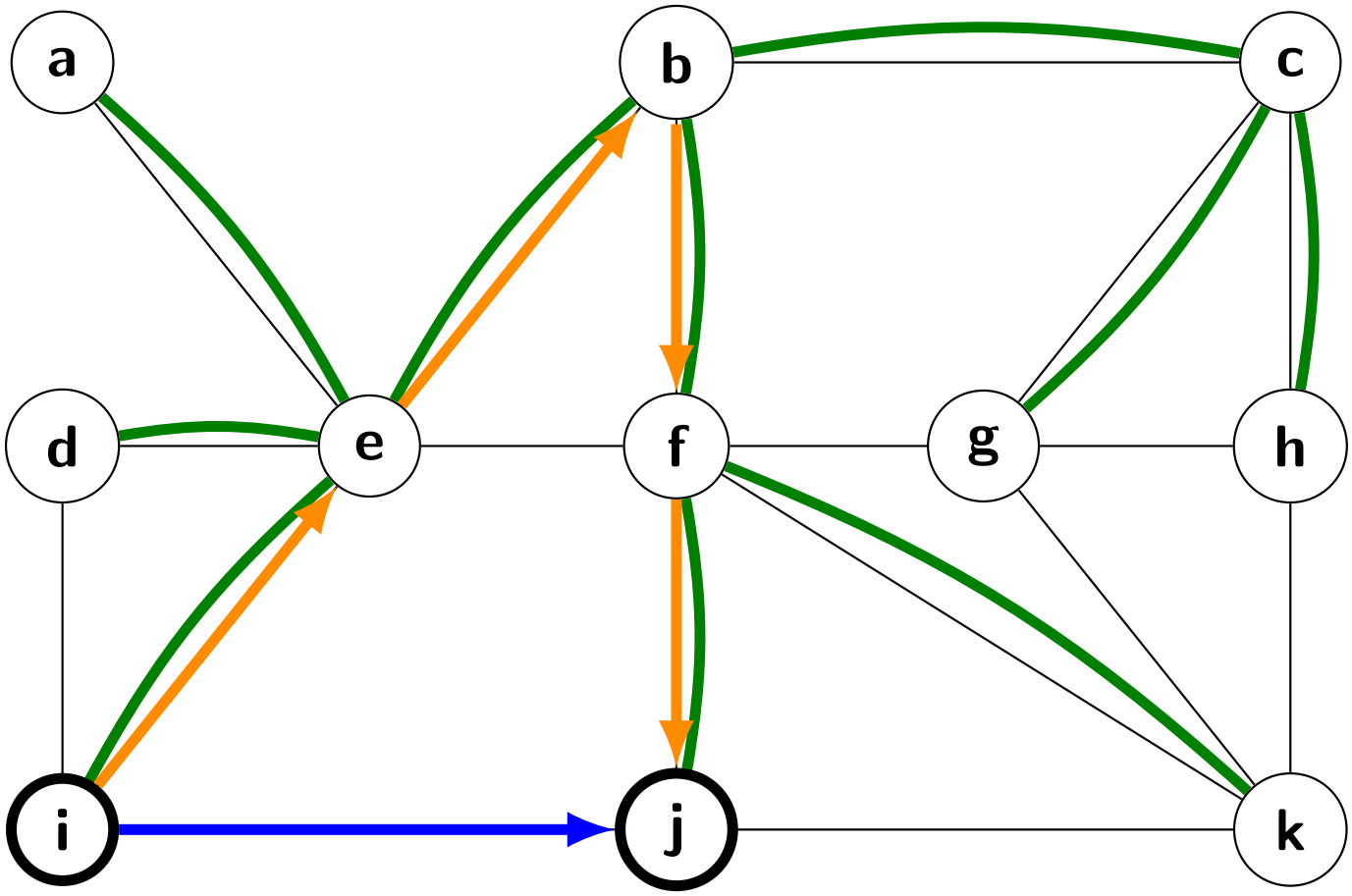
(FIB)

$c \mapsto p_c \vee p_g \vee p_h$
$f \mapsto p_f \vee p_j \vee p_k$
$e \mapsto p_a \vee p_d \vee p_e \vee p_i$

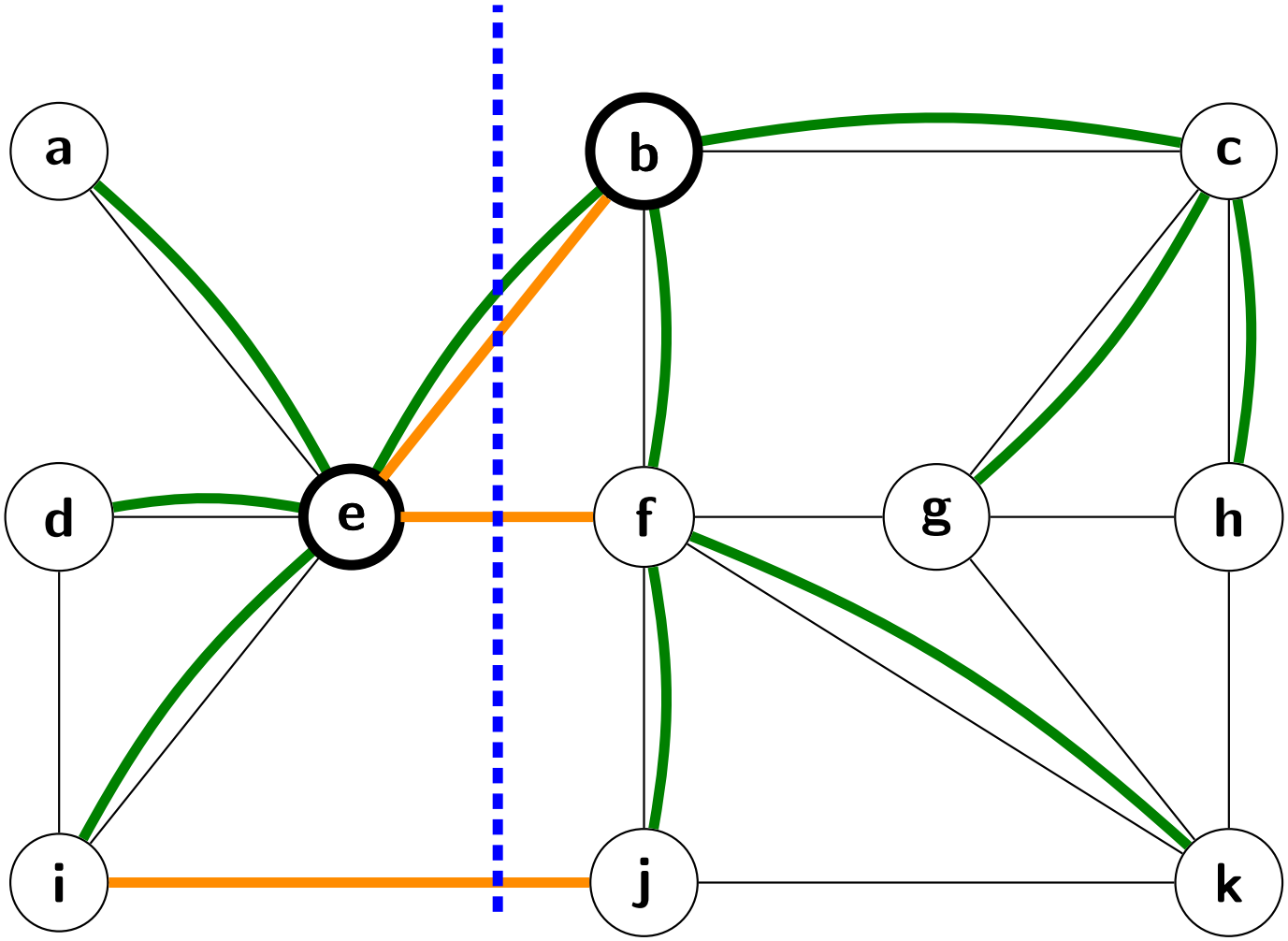



a
b
c
d
e
f
g
h
i
j
k

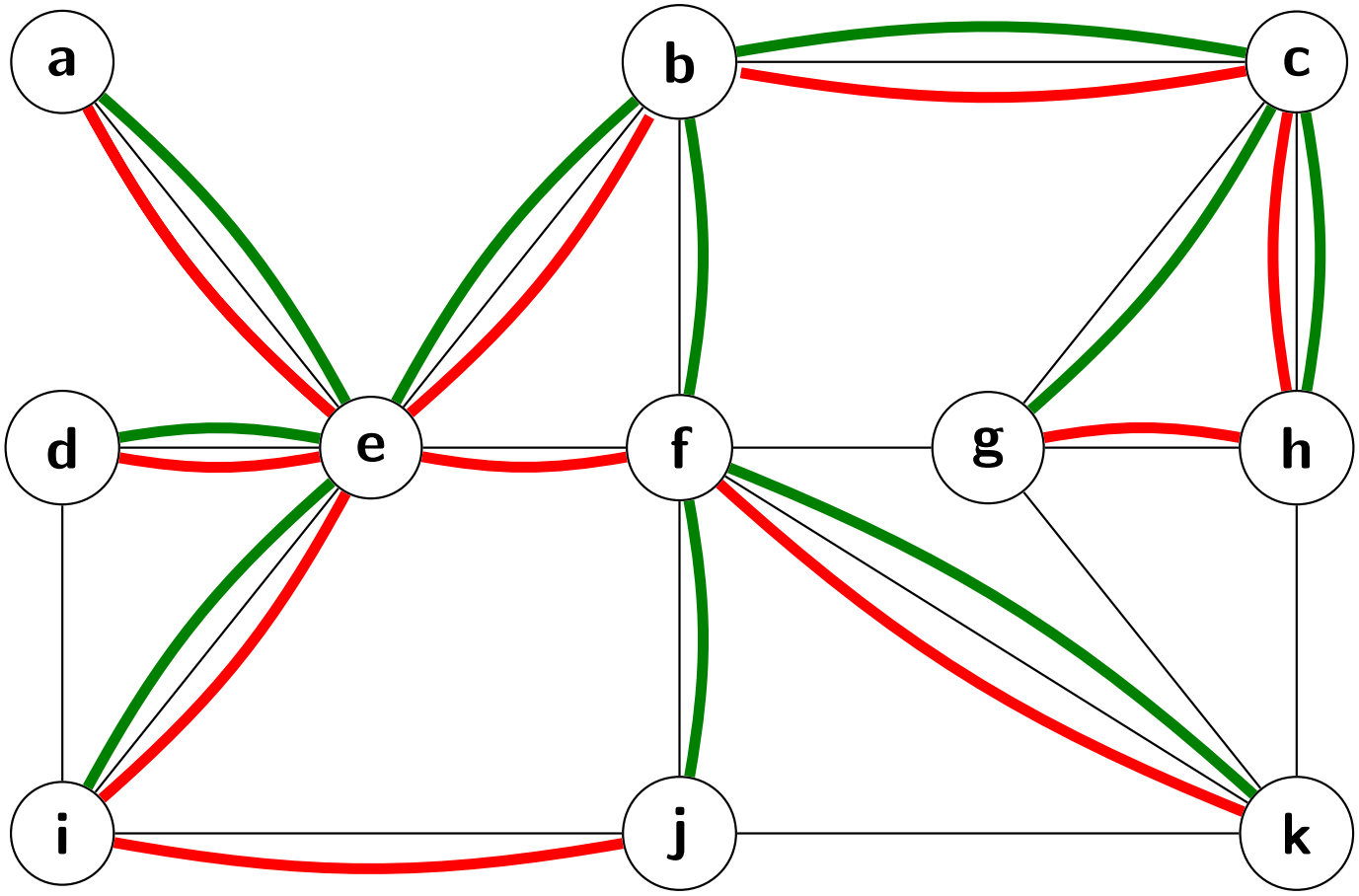
Stretched Paths



Load



Multiple Trees



router b: **tree** T**,next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

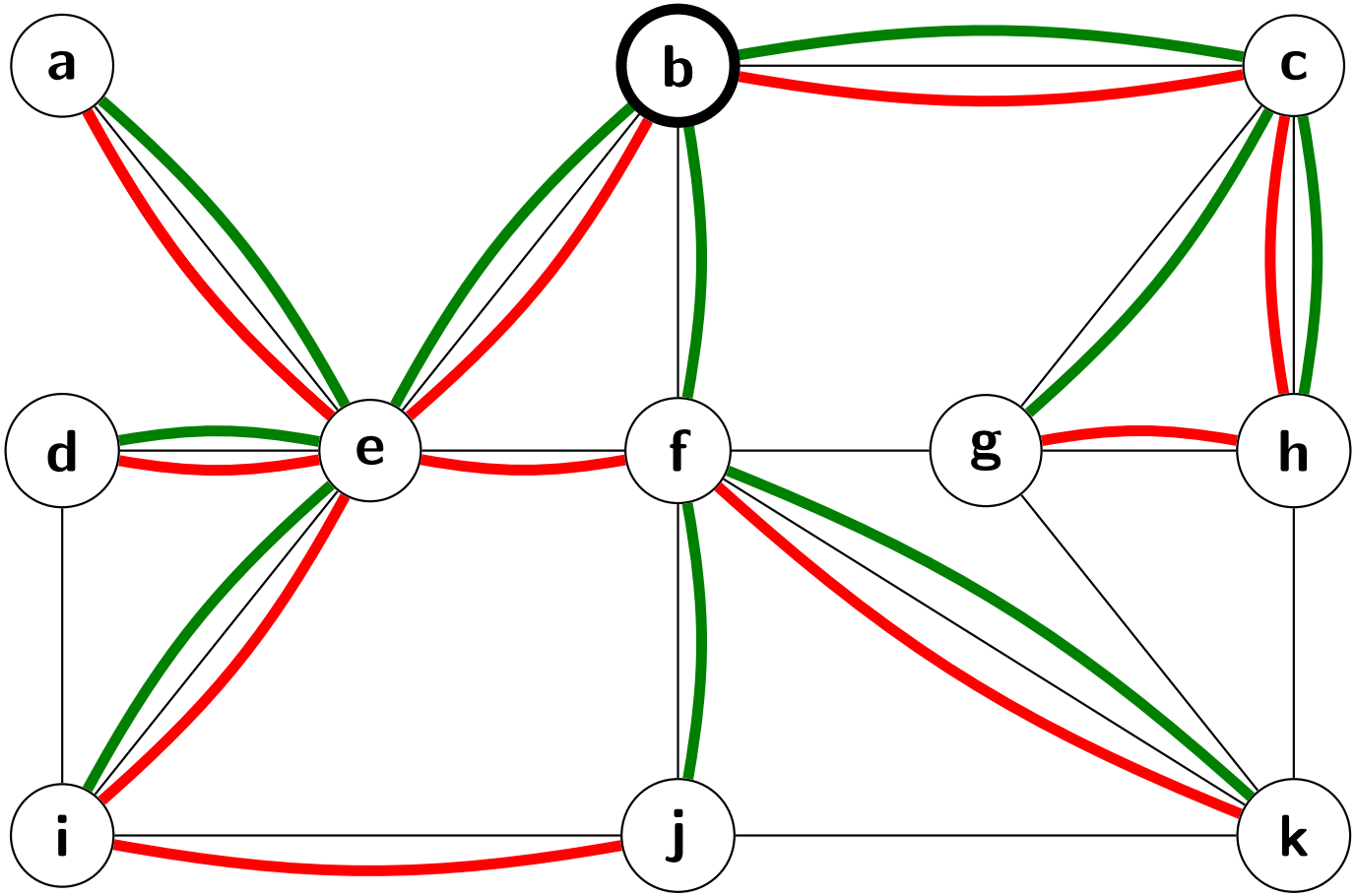
$\mathbf{T_1}, c \mapsto p_c \vee p_g \vee p_h$

$\mathbf{T_1}, f \mapsto p_f \vee p_j \vee p_k$

$\mathbf{T_1}, e \mapsto p_a \vee p_d \vee p_e \vee p_i$

$\mathbf{T_2}, c \mapsto p_c \vee p_h \vee p_g$

$\mathbf{T_2}, e \mapsto p_a \vee p_d \vee p_e \vee p_f \vee p_i \vee p_j \vee p_k$



router b: tree T,next-hop $w \rightarrow$ predicate $P_{T,w}$

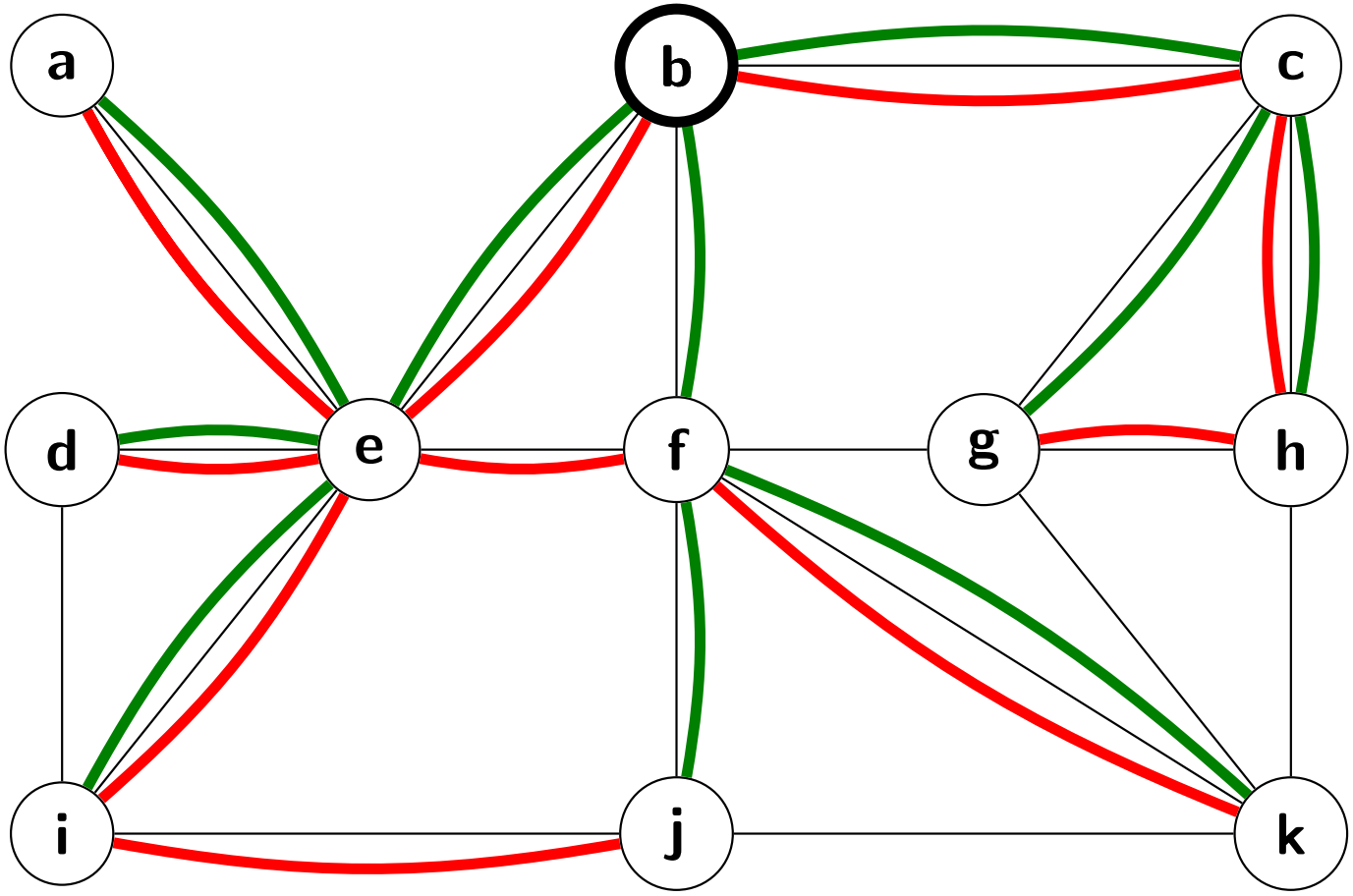
(FIB)

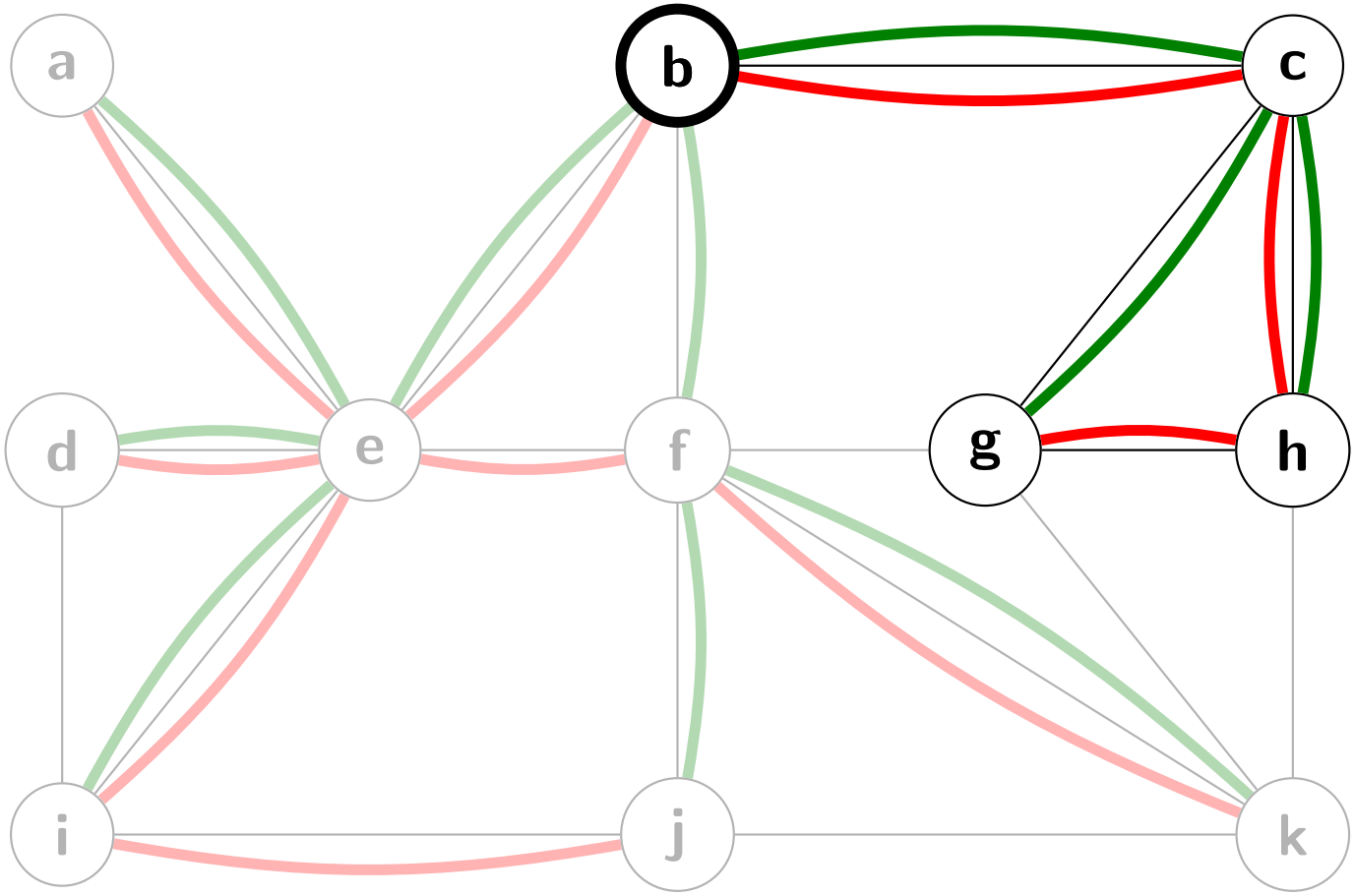
$T_1, c \mapsto p_c \vee p_g \vee p_h$

Memory Complexity Problem

$T_2, c \mapsto p_c \vee p_g \vee p_h$

$T_2, e \mapsto p_a \vee p_d \vee p_e \vee p_f \vee p_i \vee p_j \vee p_k$




a
b
c
d
e
f
g
h
i
j
k

router b: **tree** T**,next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

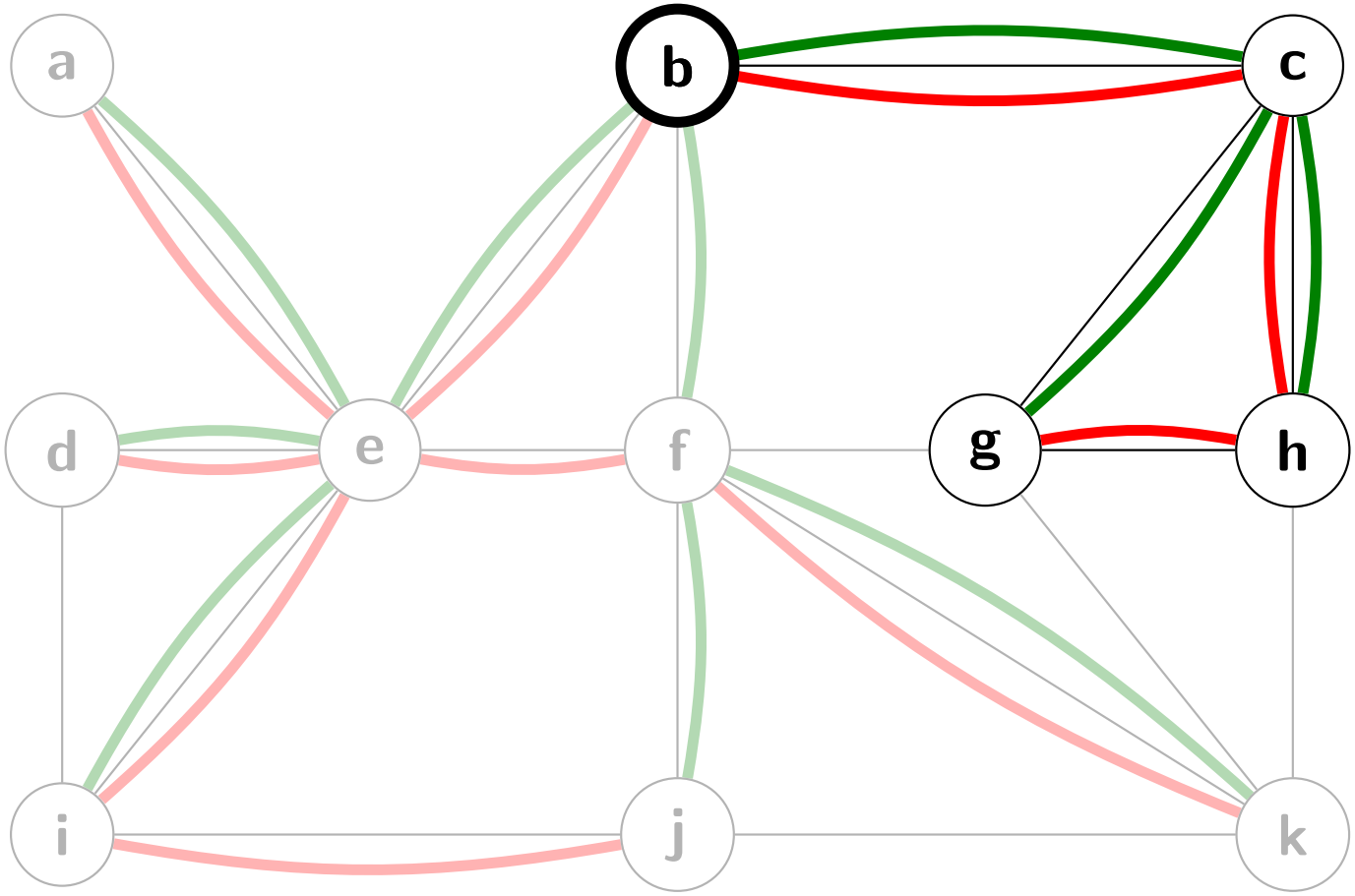
$\mathbf{T_1}, c \mapsto p_c \vee p_g \vee p_h$

$\mathbf{T_1}, f \mapsto p_f \vee p_j \vee p_k$

$\mathbf{T_1}, e \mapsto p_a \vee p_d \vee p_e \vee p_i$

$\mathbf{T_2}, c \mapsto p_c \vee p_h \vee p_g$

$\mathbf{T_2}, e \mapsto p_a \vee p_d \vee p_e \vee p_f \vee p_i \vee p_j \vee p_k$



router b: **tree** T**,next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

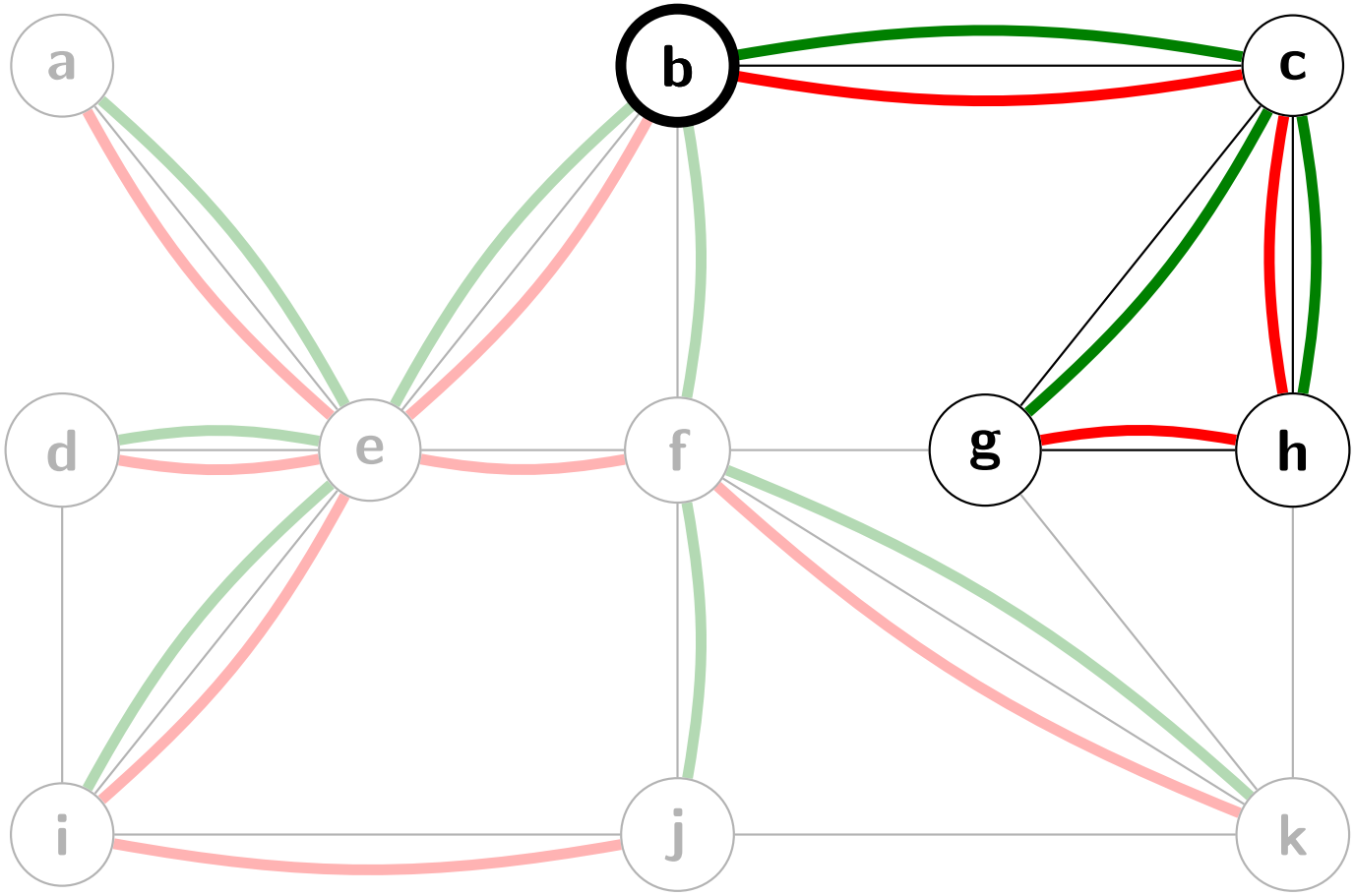
~~$\mathbf{T_1}, c \mapsto p_c \vee p_g \vee p_h$~~

$\mathbf{T_1}, f \mapsto p_f \vee p_j \vee p_k$

$\mathbf{T_1}, e \mapsto p_a \vee p_d \vee p_e \vee p_i$

~~$\mathbf{T_2}, c \mapsto p_c \vee p_h \vee p_g$~~

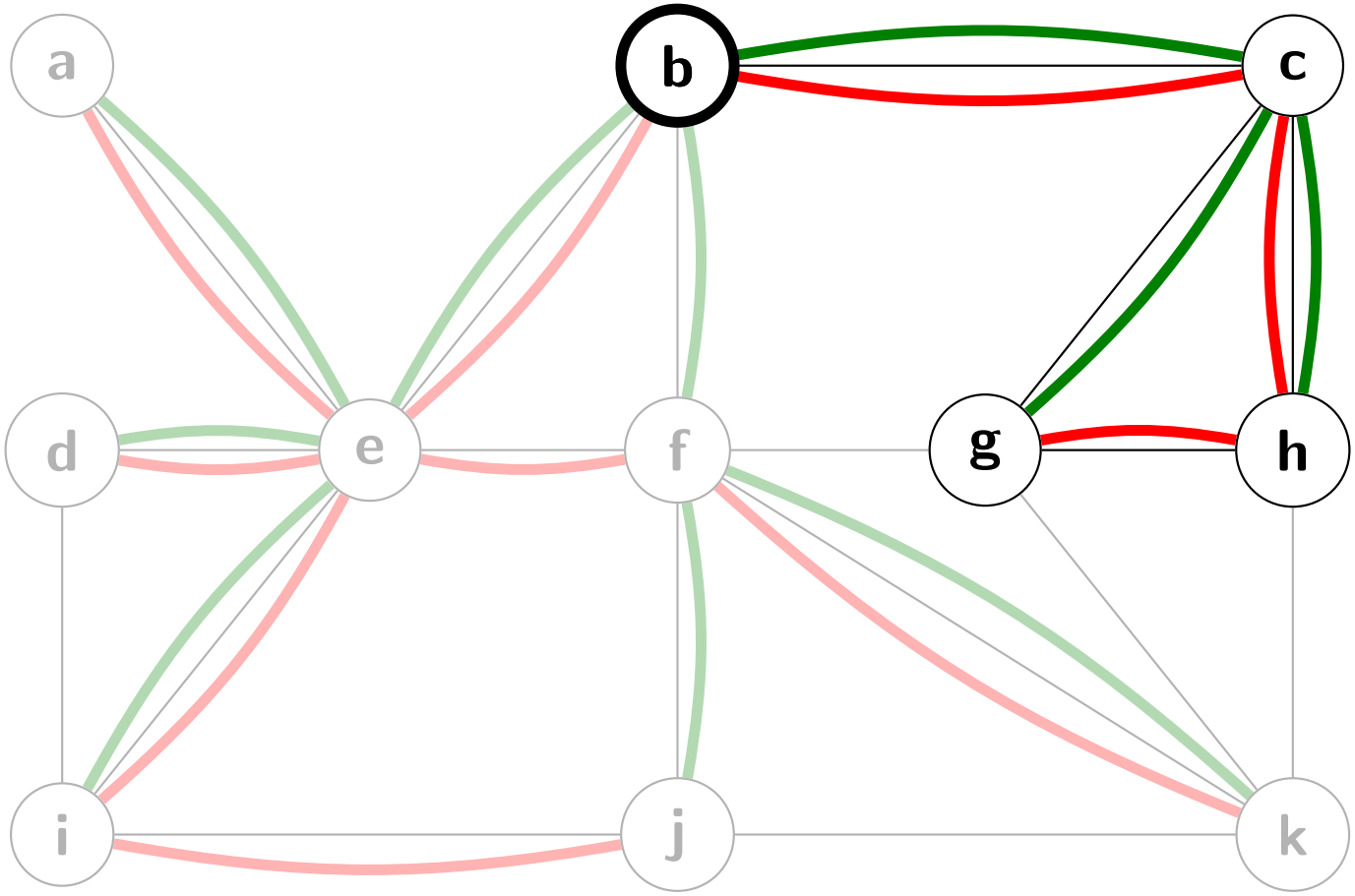
$\mathbf{T_2}, e \mapsto p_a \vee p_d \vee p_e \vee p_f \vee p_i \vee p_j \vee p_k$

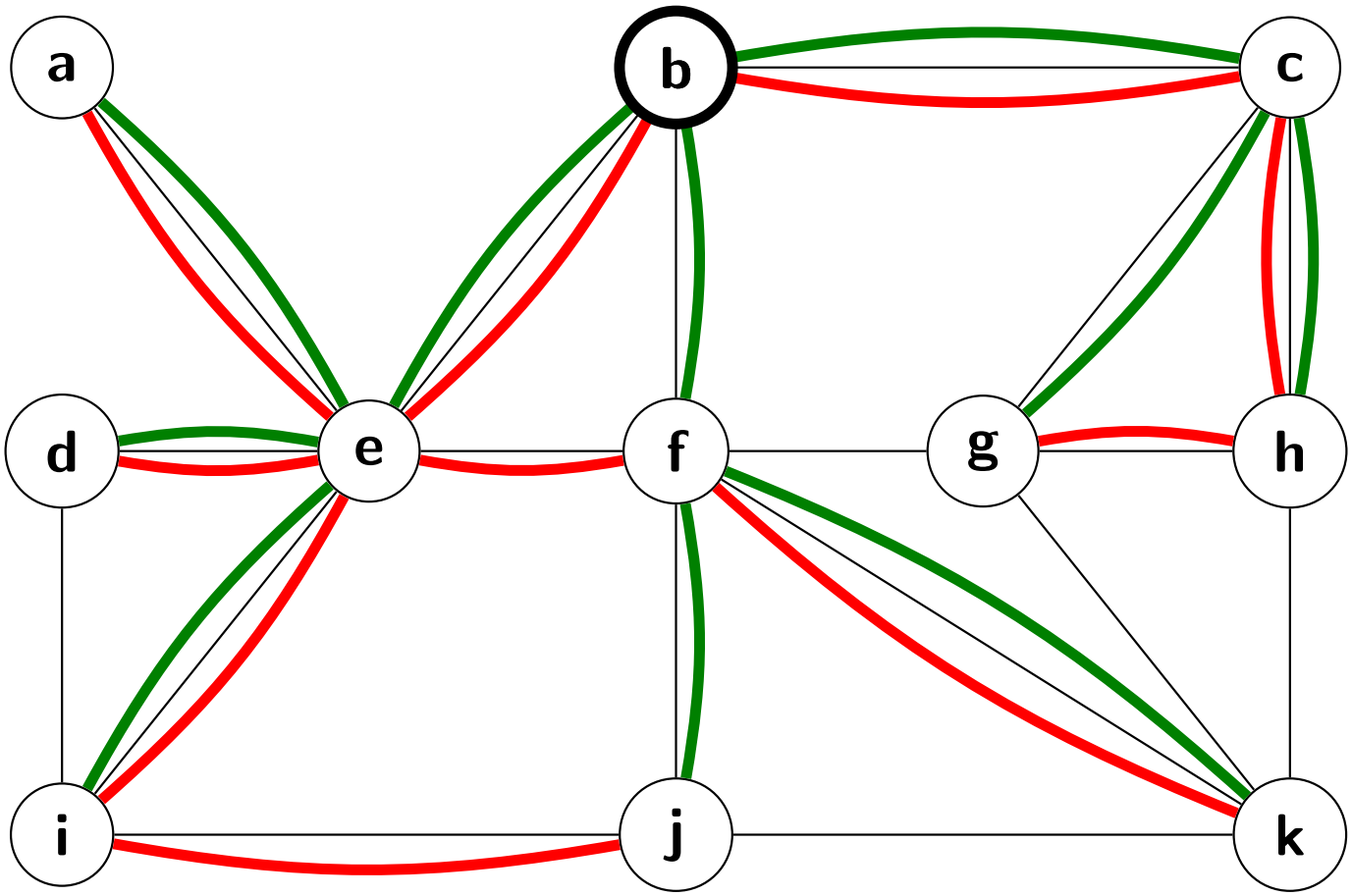


router b: **tree** T**,next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

c	$\mapsto p_c \vee p_h \vee p_g$
$\mathbf{T_1}, f$	$\mapsto p_f \vee p_j \vee p_k$
$\mathbf{T_1}, e$	$\mapsto p_a \vee p_d \vee p_e \vee p_i$
$\mathbf{T_2}, e$	$\mapsto p_a \vee p_d \vee p_e \vee p_f \vee p_i \vee p_j \vee p_k$

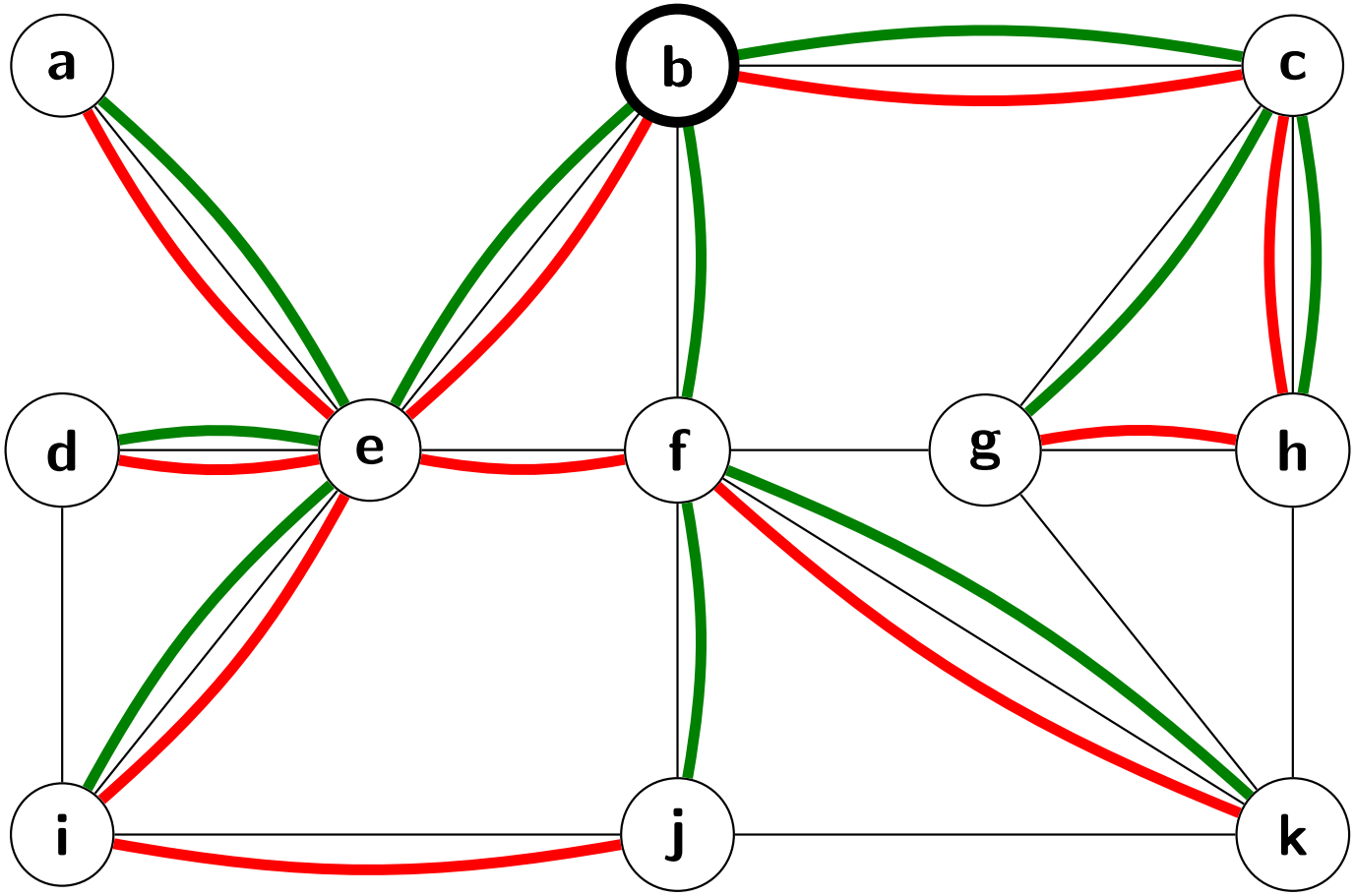



a
b
c
d
e
f
g
h
i
j
k

router b: **next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

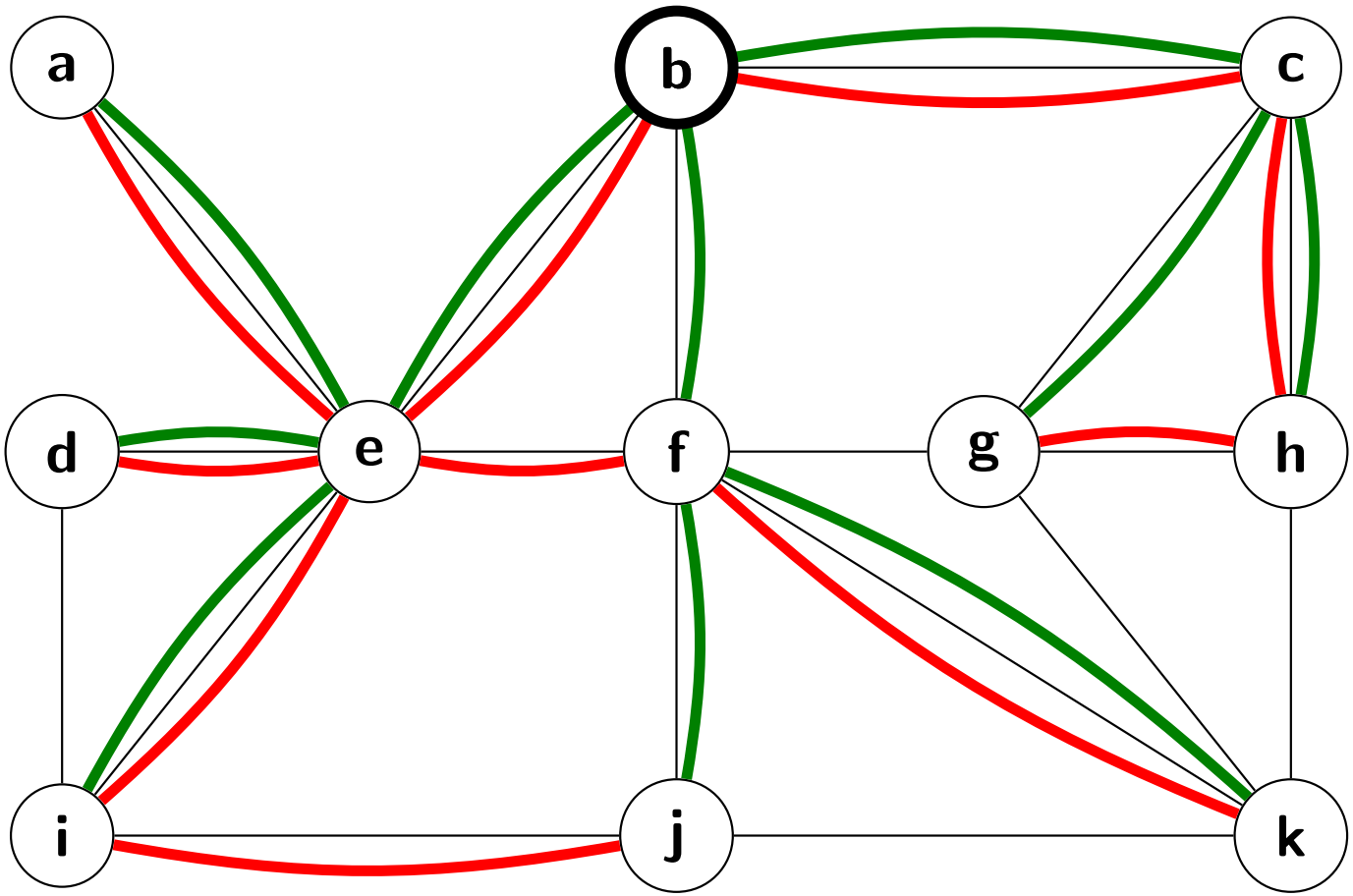
c	$\mapsto p_c \vee p_h \vee p_g$
$\mathbf{T_1}, f$	$\mapsto p_f \vee p_j \vee p_k$
$\mathbf{T_1}, e$	$\mapsto p_a \vee p_d \vee p_e \vee p_i$
$\mathbf{T_2}, e$	$\mapsto p_a \vee p_d \vee p_e \vee p_f \vee p_i \vee p_j \vee p_k$



router b: **next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

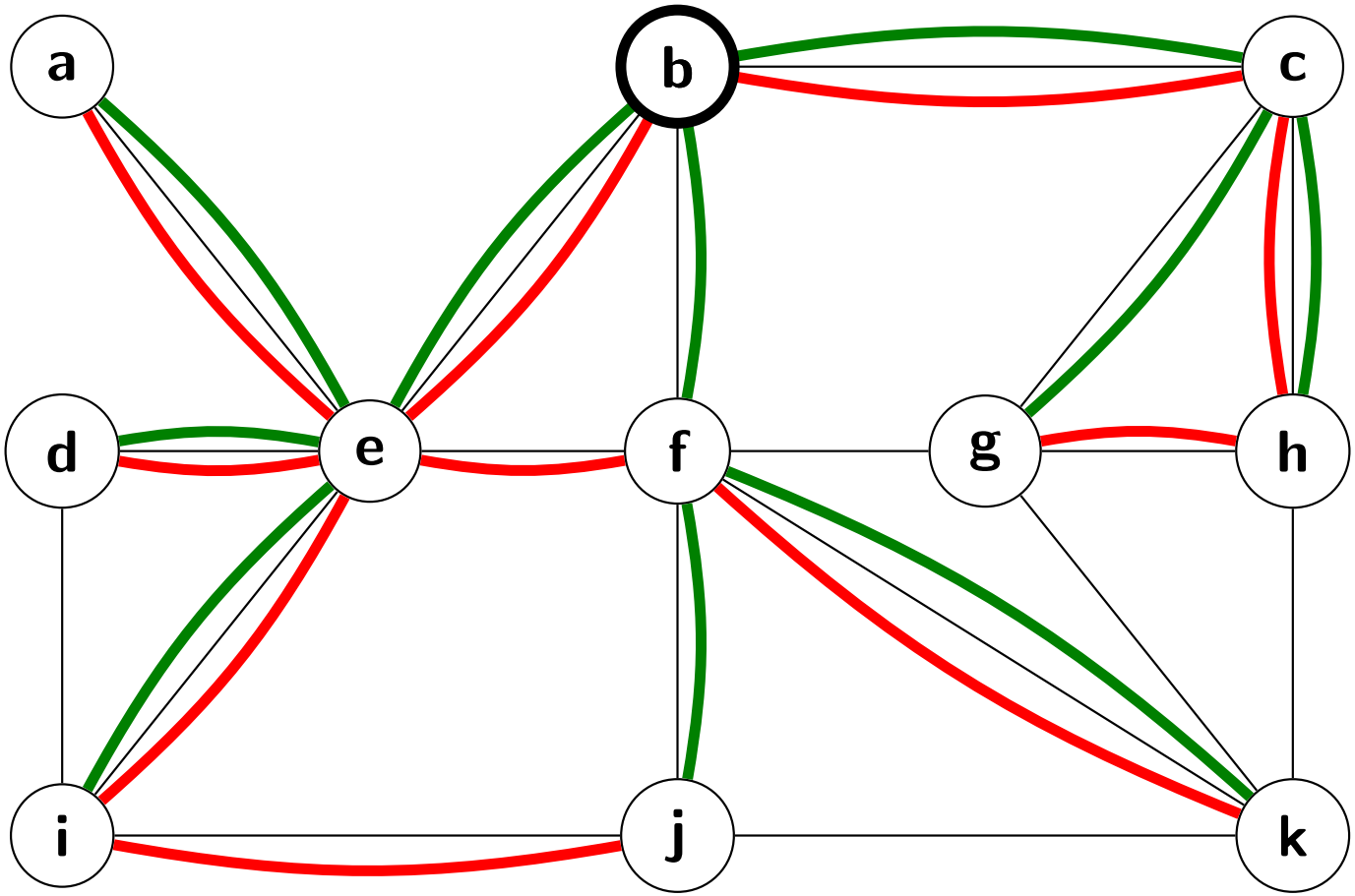
$$
\begin{aligned}
c &\mapsto p_c \vee p_h \vee p_g \\
f &\mapsto (\mathbf{T_1} \wedge p_f) \vee (\mathbf{T_1} \wedge p_j) \vee (\mathbf{T_1} \wedge p_k) \\
e &\mapsto (\mathbf{T_1} \wedge p_a) \vee (\mathbf{T_1} \wedge p_d) \vee (\mathbf{T_1} \wedge p_e) \vee (\mathbf{T_1} \wedge p_i) \vee \\
&\quad\ (\mathbf{T_2} \wedge p_a) \vee (\mathbf{T_2} \wedge p_d) \vee (\mathbf{T_2} \wedge p_e) \vee (\mathbf{T_2} \wedge p_i) \vee \\
&\quad\ (\mathbf{T_2} \wedge p_f) \vee (\mathbf{T_2} \wedge p_j) \vee (\mathbf{T_2} \wedge p_k)
\end{aligned}
$$



router b: next-hop $w \rightarrow$ predicate $P_{T,w}$

(FIB)

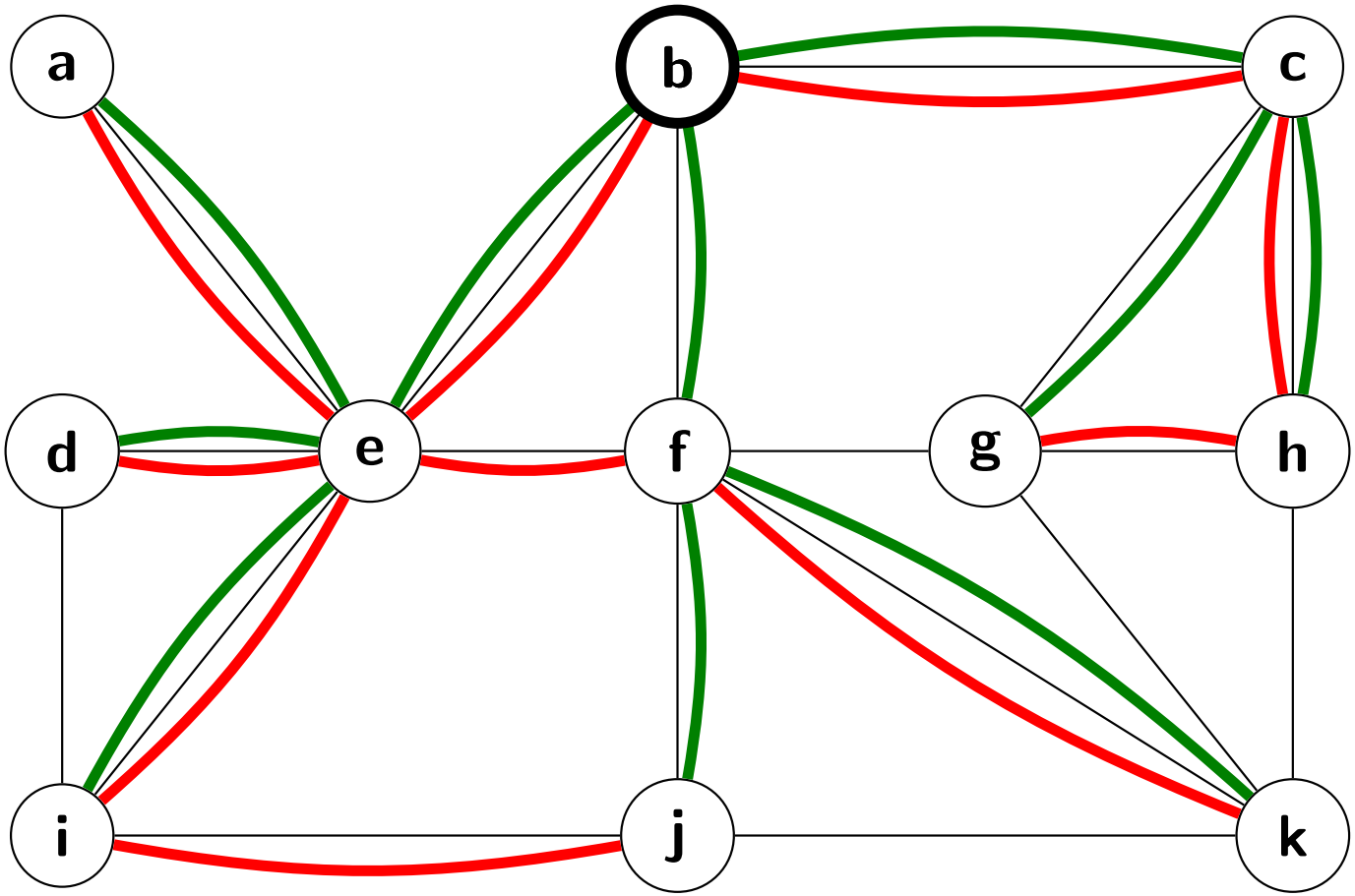
$$
\begin{aligned}
c &\mapsto p_c \vee p_h \vee p_g \\
f &\mapsto (\mathbf{T_1} \wedge p_f) \vee (\mathbf{T_1} \wedge p_j) \vee (\mathbf{T_1} \wedge p_k) \\
e &\mapsto \boxed{(\mathbf{T_1} \wedge p_a)} \vee \boxed{(\mathbf{T_1} \wedge p_d)} \vee \boxed{(\mathbf{T_1} \wedge p_e)} \vee \boxed{(\mathbf{T_1} \wedge p_i)} \vee \\
&\quad \boxed{(\mathbf{T_2} \wedge p_a)} \vee \boxed{(\mathbf{T_2} \wedge p_d)} \vee \boxed{(\mathbf{T_2} \wedge p_e)} \vee \boxed{(\mathbf{T_2} \wedge p_i)} \vee \\
&\quad (\mathbf{T_2} \wedge p_f) \vee (\mathbf{T_2} \wedge p_j) \vee (\mathbf{T_2} \wedge p_k)
\end{aligned}
$$



router b: **next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

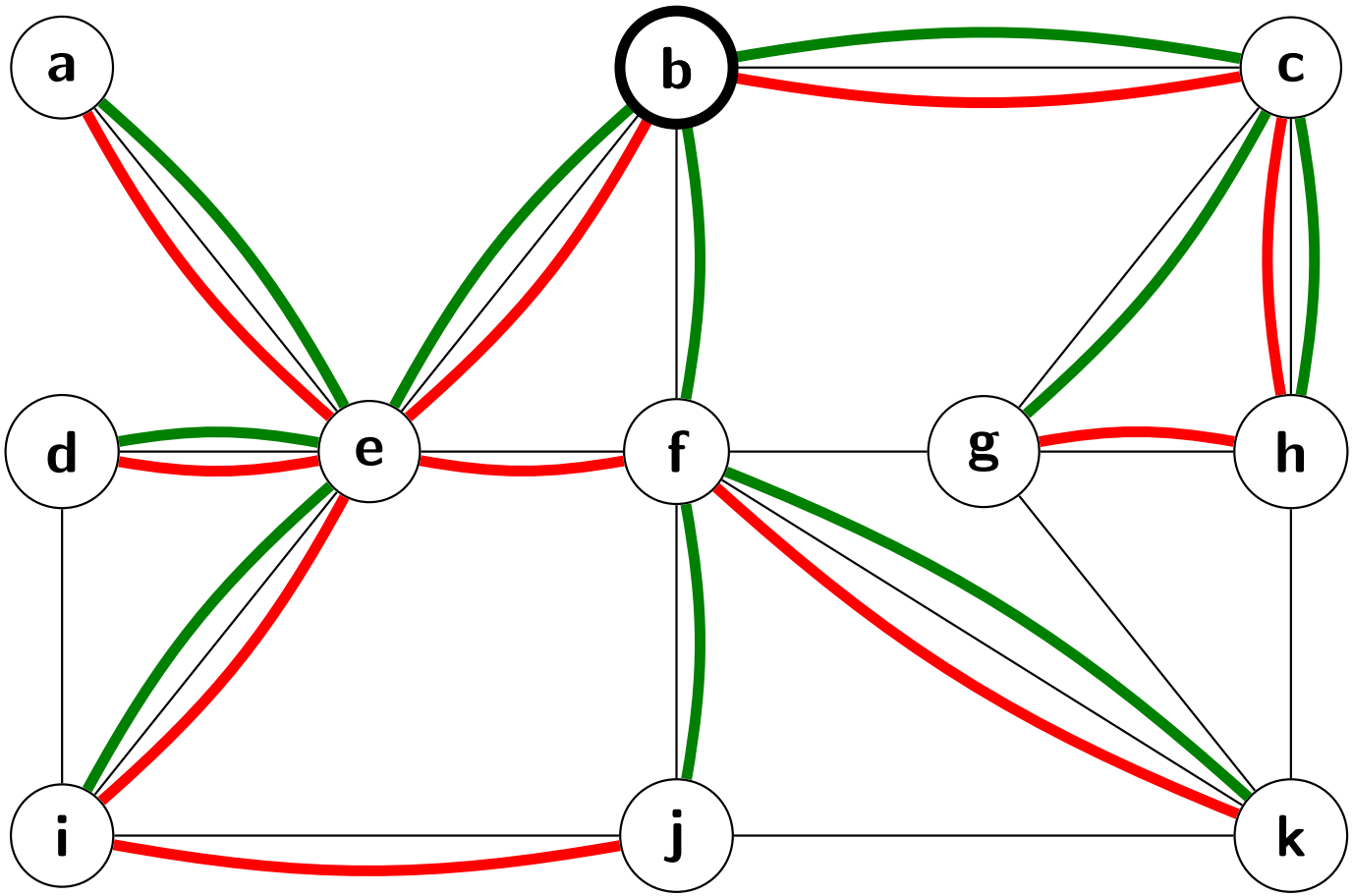
$$
\begin{array}{l}
c \mapsto p_c \vee p_h \vee p_g \\
f \mapsto (\mathbf{T_1} \wedge p_f) \vee (\mathbf{T_1} \wedge p_j) \vee (\mathbf{T_1} \wedge p_k) \\
e \mapsto p_a \vee p_d \vee p_e \vee p_i \vee \\
\qquad (\mathbf{T_2} \wedge p_f) \vee (\mathbf{T_2} \wedge p_j) \vee (\mathbf{T_2} \wedge p_k)
\end{array}
$$



router b: **next-hop** $w \rightarrow$ **predicate** $P_{T,w}$

(FIB)

$$
\begin{array}{l}
c \mapsto p_c \vee p_h \vee p_g \\
f \mapsto (\mathbf{T_1} \wedge p_f) \vee (\mathbf{T_1} \wedge p_j) \vee (\mathbf{T_1} \wedge p_k) \\
e \mapsto p_a \vee p_d \vee p_e \vee p_i \vee \\
\qquad (\mathbf{T_2} \wedge p_f) \vee (\mathbf{T_2} \wedge p_j) \vee (\mathbf{T_2} \wedge p_k)
\end{array}
$$

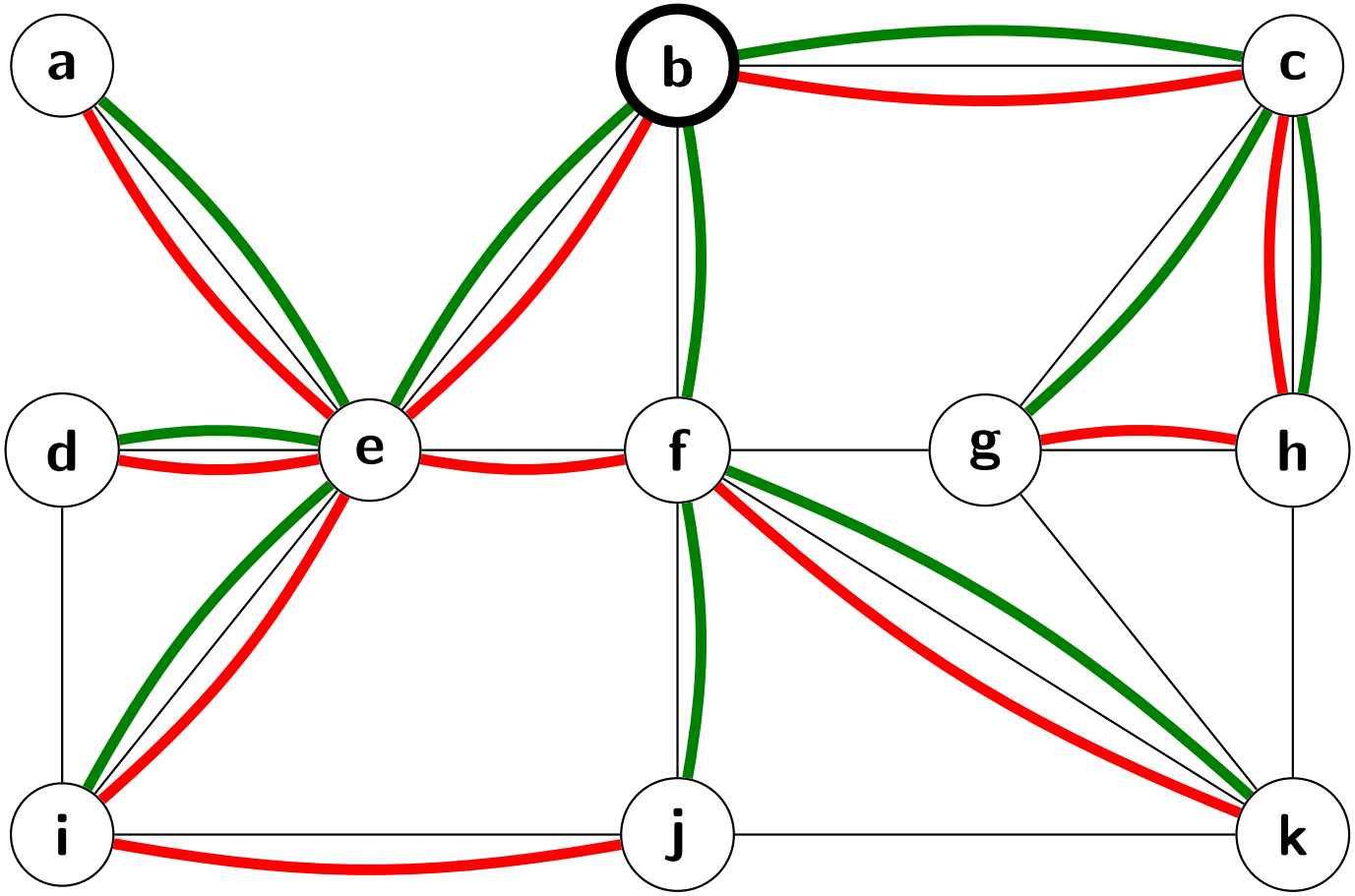


router b: next-hop $w \rightarrow$ predicate $P_{T,w}$

(FIB) $c \mapsto p_c \vee p_h \vee p_g$

coming soon: new data structure

$e \mapsto p_a \vee p_d \vee p_e \vee p_i \vee$
$(\mathsf{T}_2 \wedge p_f) \vee (\mathsf{T}_2 \wedge p_j) \vee (\mathsf{T}_2 \wedge p_k)$



Evaluation

Evaluation

- Q 1: Is it possible to use trees to route traffic over the Internet?

Evaluation

- Q 1: Is it possible to use trees to route traffic over the Internet?
- Q 2: Do user-defined descriptor-based addresses aggregate?

Evaluation

- Q 1: Is it possible to use trees to route traffic over the Internet?
- Q 2: Do user-defined descriptor-based addresses aggregate?

We need a workload

What do we need?

- Topology

What do we need?

- Topology
- Distribute users on the nodes

What do we need?

- Topology
- Distribute users on the nodes
- Assign applications to users

What do we need?

- Topology
- Distribute users on the nodes
- Assign applications to users
- Create the registrations

What do we need?

- Topology
  - AS-level Internet topology
- Distribute users on the nodes
- Assign applications to users
- Create the registrations

What do we need?

- Topology
- Distribute users on the nodes
  - assigned to each AS according to the estimated population
- Assign applications to users
- Create the registrations

What do we need?

- Topology
- Distribute users on the nodes
- Assign applications to users
  - selected according to the real number of users
- Create the registrations

What do we need?

- Topology
- Distribute users on the nodes
- Assign applications to users
- Create the registrations
  - ???

- Imagine the future Internet
  - study the users behavior on different applications
  - define registrations with actual tags used by users

- Imagine the future Internet
  - study the users behavior on different applications
  - define registrations with actual tags used by users

- Push content
  - web content and blog posts
  - short messages (tweets)

- Imagine the future Internet
  - study the users behavior on different applications
  - define registrations with actual tags used by users
- Push content
  - web content and blog posts
  - short messages (tweets)
- Pull content
  - videos

Web Content

Goal: Understand users interests

Web Content

Goal: Understand users interests

- Users Bookmarks (Delicious)
  - bookmarks = subscription

Application	**User**	**Registration**
Delicious	1M	124M
Blogs	60K	180K
Video	1K	10K
Twitter Graph	41M	1B
Twitter Messages	400K	500K

Data Amplification

Data Amplification

- Multiple languages
  - replicate the data for the 25 most spoken languages
  - language is chosen according to the popularity

Data Amplification

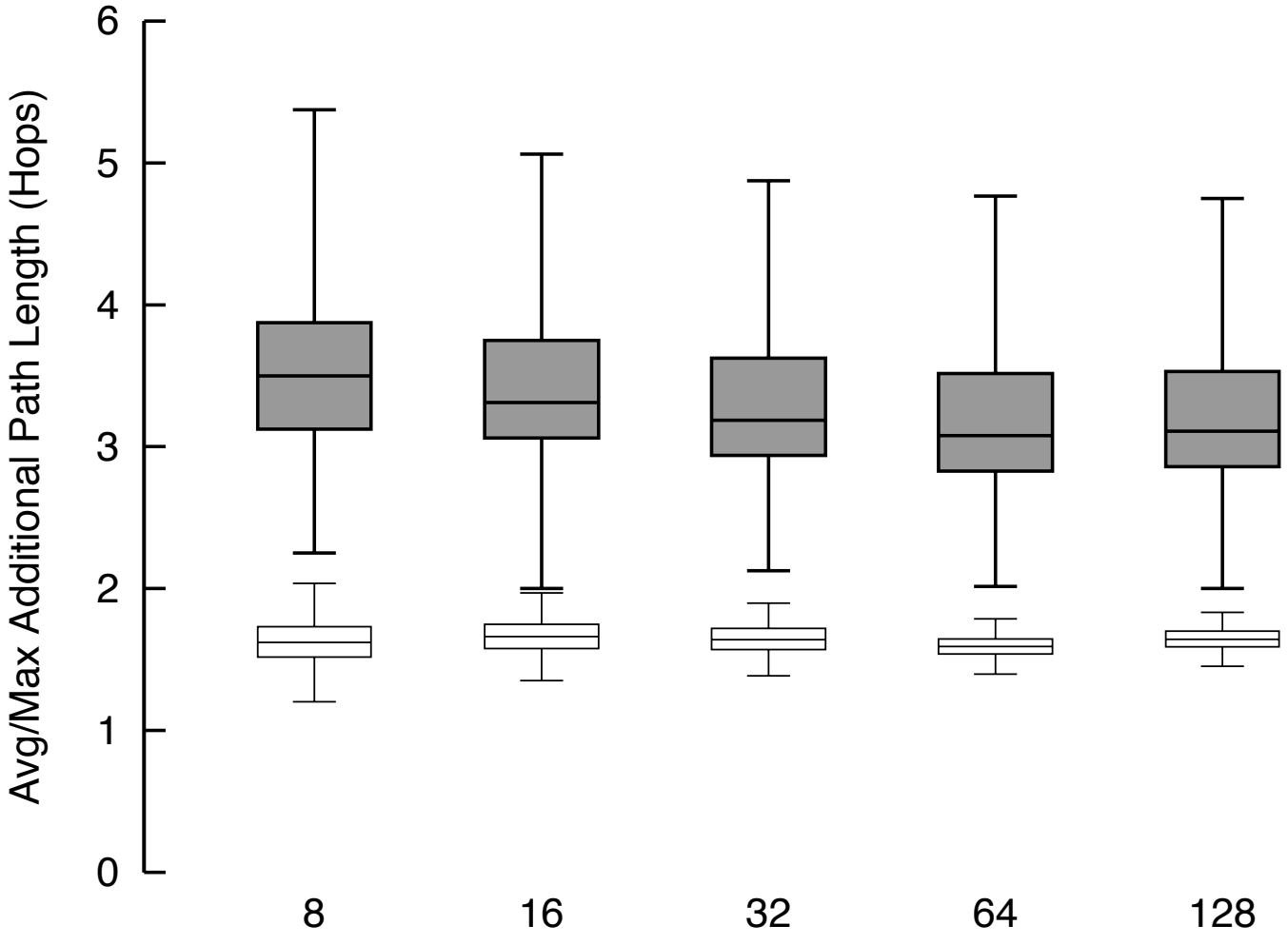
- Multiple languages
  - replicate the data for the 25 most spoken languages
  - language is chosen according to the popularity
- Synonyms
  - for each word we define synonyms
  - synonyms are randomly chosen

Evaluation

- Q 1: Is it possible to use trees to route traffic over the Internet?
- Q 2: Do user-defined descriptor-based addresses aggregate?

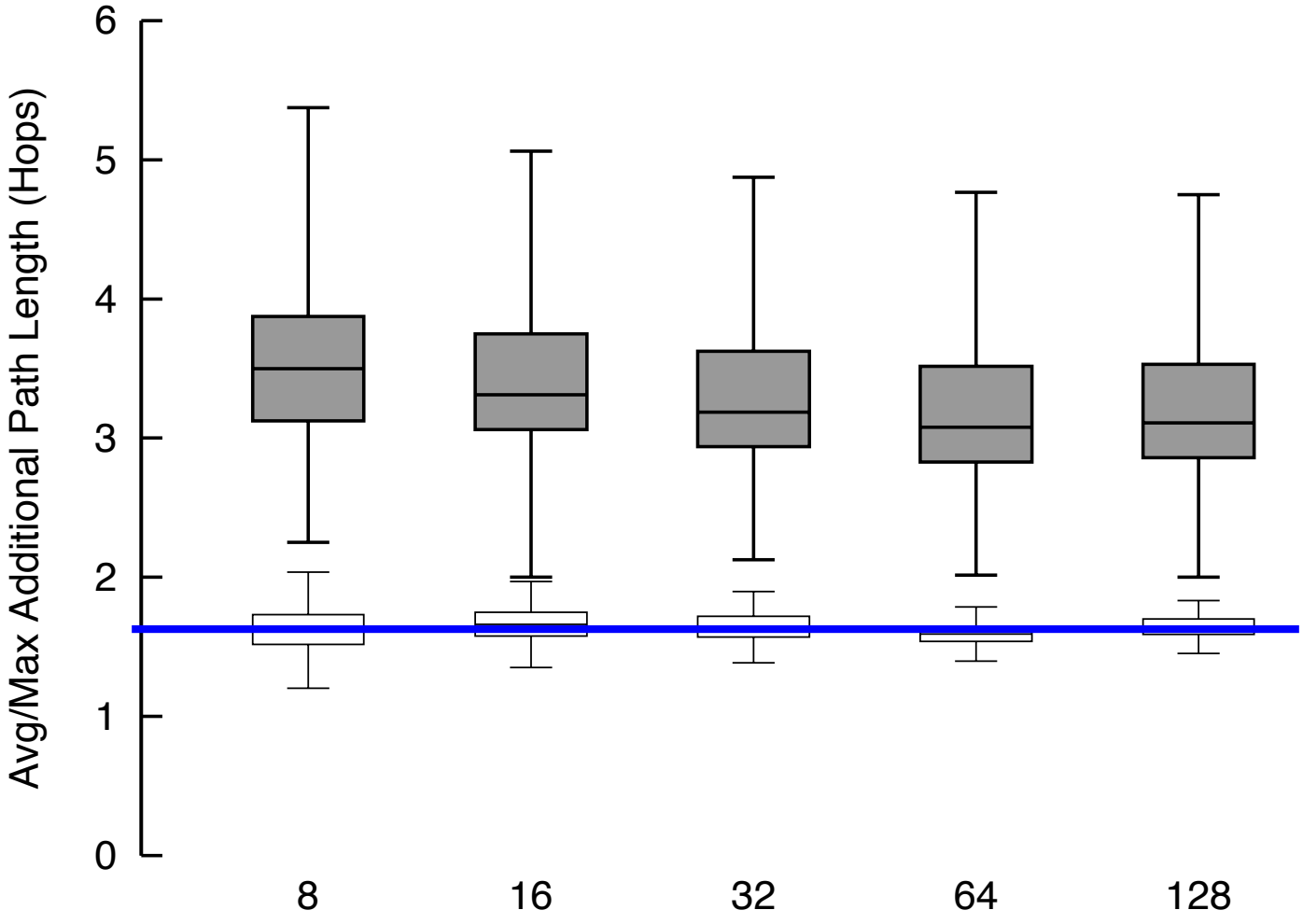
Additional cost in using k trees on the actual AS-level topology

with k = 8, 16, 32, 64, 128 trees



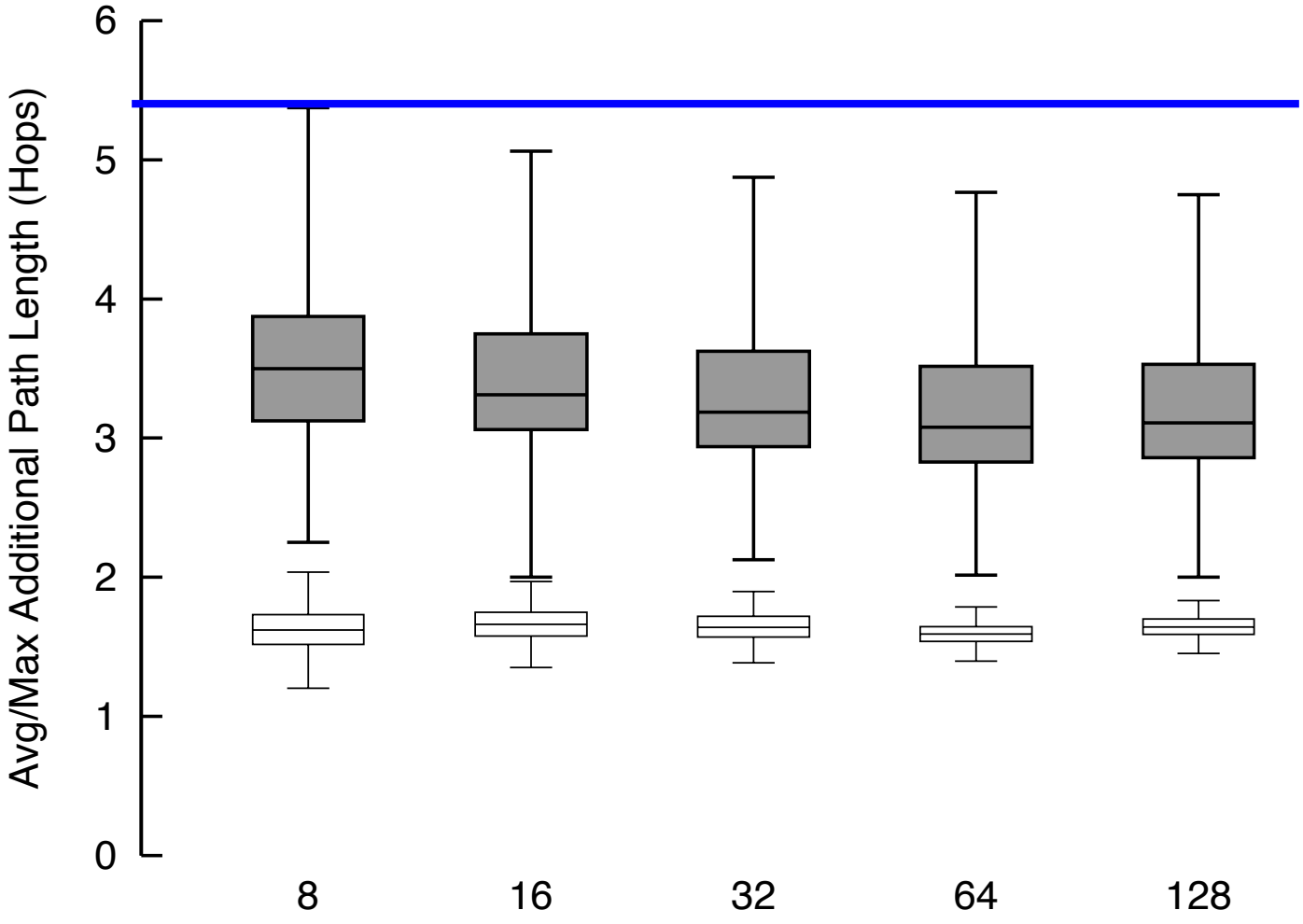
Additional cost in using k trees on the actual AS-level topology

with k = 8, 16, 32, 64, 128 trees



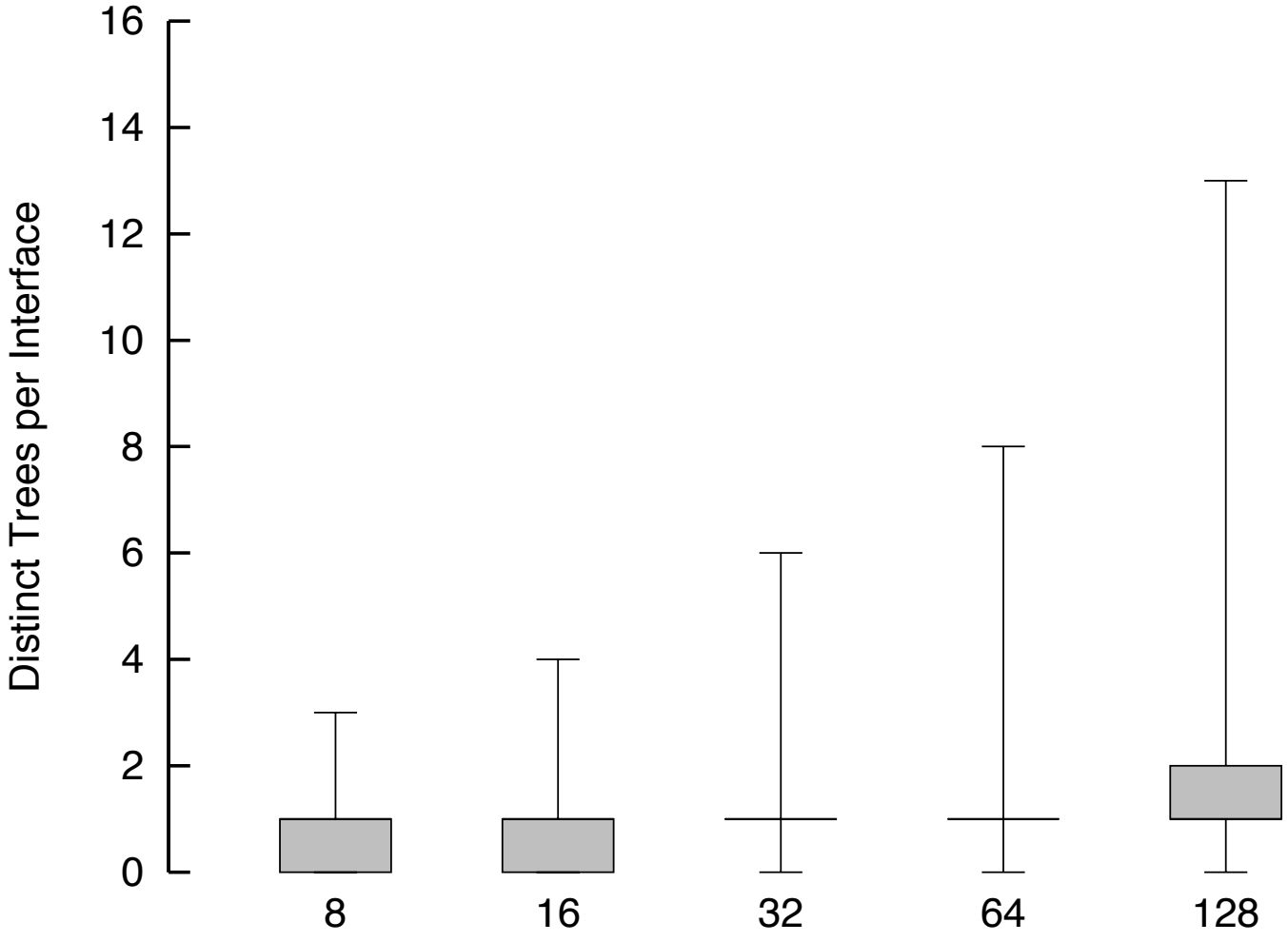
Additional cost in using k trees on the actual AS-level topology

with k = 8, 16, 32, 64, 128 trees



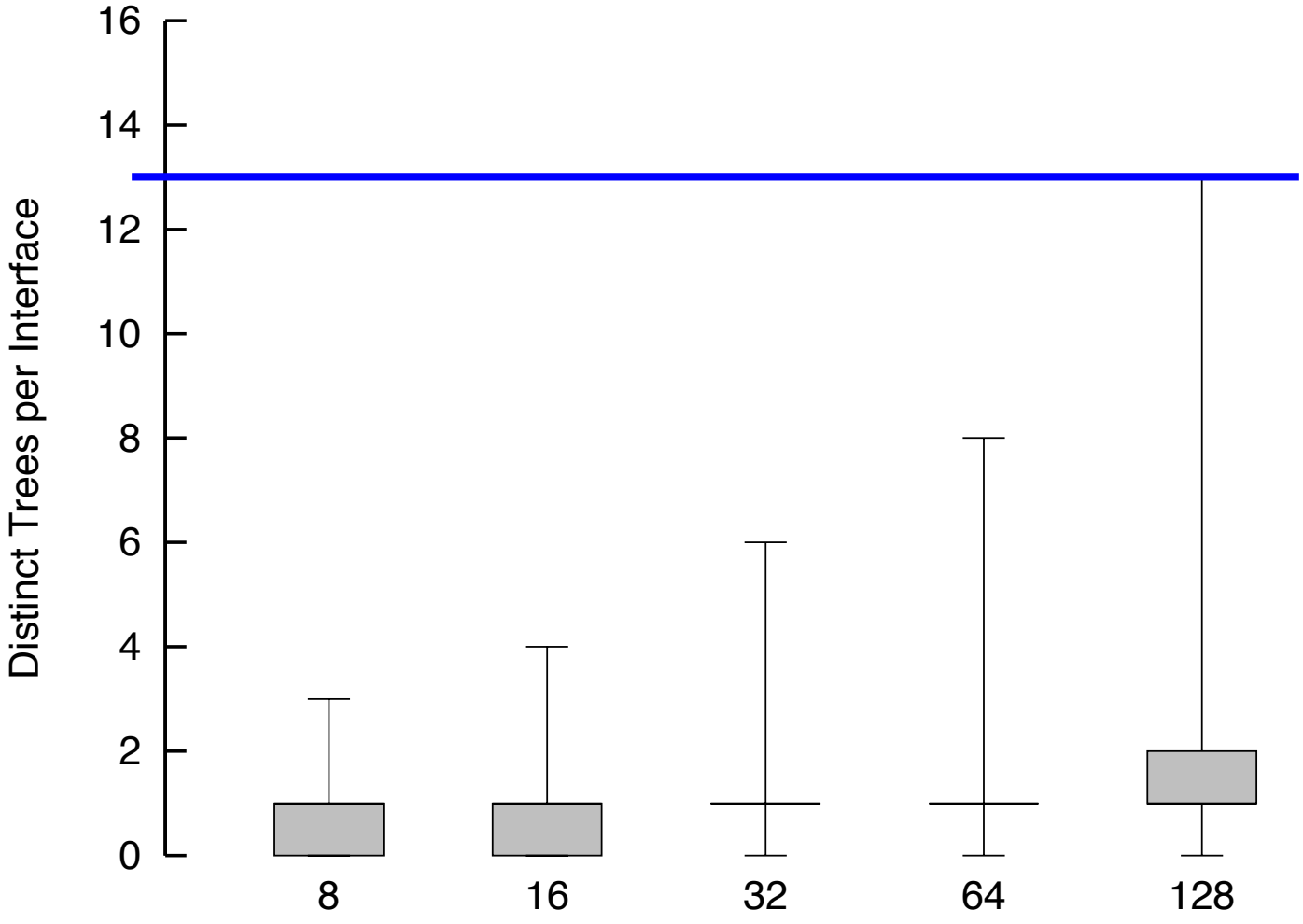
Tree aggregation in FIBs

with k = 8, 16, 32, 64, 128 trees



Tree aggregation in FIBs

with k = 8, 16, 32, 64, 128 trees



Aggregation of tag-based addresses in FIBs

memory requirements in a central node for a single tree

2.5M users

	All Interfaces	Largest
Interfaces	325	1
Destinations	42,112	6,559
Tags	276,501,173	35,814,399
Original Descriptors	85,504,514	10,727,593
Actual Descriptors	10,880,657	1,145,713
Size (MB)	518.83	54.63

Aggregation of tag-based addresses in FIBs

memory requirements in a central node for a single tree

2.5M users

	All Interfaces	Largest
Interfaces	325	1
Destinations	42,112	6,559
Tags	276,501,173	35,814,399
Original Descriptors	85,504,514	10,727,593
Actual Descriptors	10,880,657	1,145,713
Size (MB)	518.83	54.63

Aggregation of tag-based addresses in FIBs

memory requirements in a central node for a single tree

2.5M users

	All Interfaces	Largest
Interfaces	325	1
Destinations	42,112	6,559
Tags	276,501,173	35,814,399
Original Descriptors	85,504,514	10,727,593
Actual Descriptors	10,880,657	1,145,713
Size (MB)	518.83	54.63

Aggregation of tag-based addresses in FIBs

memory requirements in a central node for a single tree

2.5M users

	All Interfaces	Largest
Interfaces	325	1
Destinations	42,112	6,559
Tags	276,501,173	35,814,399
Original Descriptors	85,504,514	10,727,593
Actual Descriptors	10,880,657	1,145,713
Size (MB)	518.83	54.63

Aggregation of tag-based addresses in FIBs

memory requirements in a central node for a single tree

2.5M users

	All Interfaces	Largest
Interfaces	325	1
Destinations	42,112	6,559
Tags	276,501,173	35,814,399
Original Descriptors	85,504,514	10,727,593
Actual Descriptors	10,880,657	1,145,713
Size (MB)	518.83	54.63
Aggregation Factor	**7.85**	**9.36**

Aggregation of tag-based addresses in FIBs

memory requirements in a central node for a single tree

2.5M users

	All Interfaces	Largest
Interfaces	325	1
Destinations	42,112	6,559
Tags	276,501,173	35,814,399
Original Descriptors	85,504,514	10,727,593
Actual Descriptors	10,880,657	1,145,713
Size (MB)	518.83	54.63

Bloom Filter size = 400 bits

Current Work

- Workload: 25M users, 513M descriptors, 8 trees
- Total descriptors: 4.1 billion

Current Work

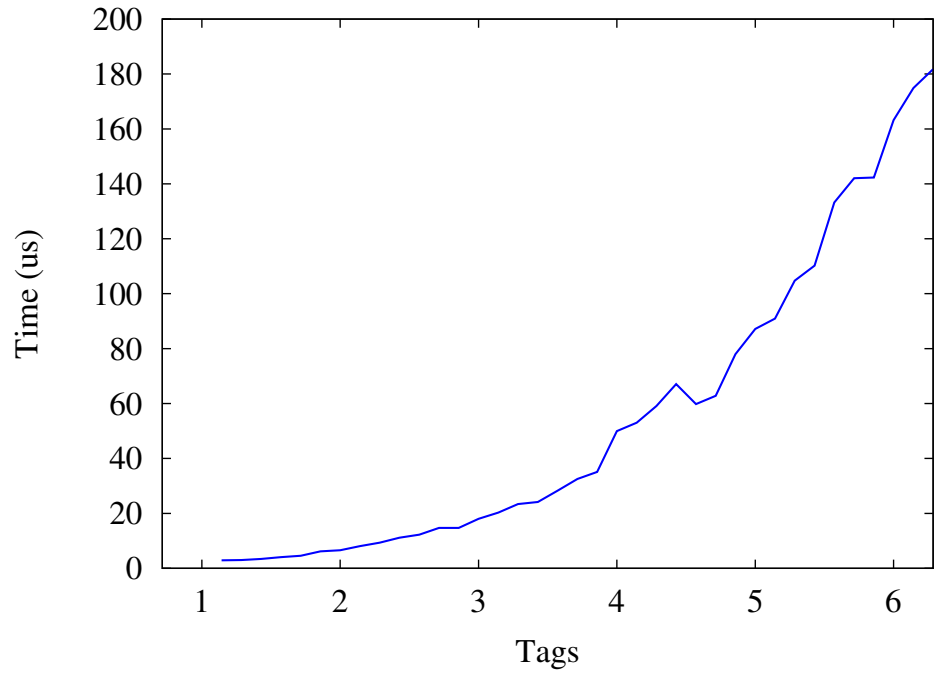
- Workload: 25M users, 513M descriptors, 8 trees
- Total descriptors: 4.1 billion
- Compressed Table: 300M descriptors

Current Work

- Workload: 25M users, 513M descriptors, 8 trees
- Total descriptors: 4.1 billion
- Compressed Table: 300M descriptors
- Update time (average): 6 μsec per descriptor

Current Work

- Workload: 25M users, 513M descriptors, 8 trees
- Total descriptors: 4.1 billion
- Compressed Table: 300M descriptors
- Update time (average): 6 μsec per descriptor
- Matching time



Conclusion

- Routing scheme:
  - tag-based address
  - push-pull communication

Conclusion

- Routing scheme:
  - tag-based address
  - push-pull communication
- Scalability:
  - AS-topology can be cover with few trees
  - FIB size is reasonable

Conclusion

- Routing scheme:
  - tag-based address
  - push-pull communication
- Scalability:
  - AS-topology can be cover with few trees
  - FIB size is reasonable
- Workload