The Root of the Matter: A Discussion of DNS Security Part 2

Mark Allman

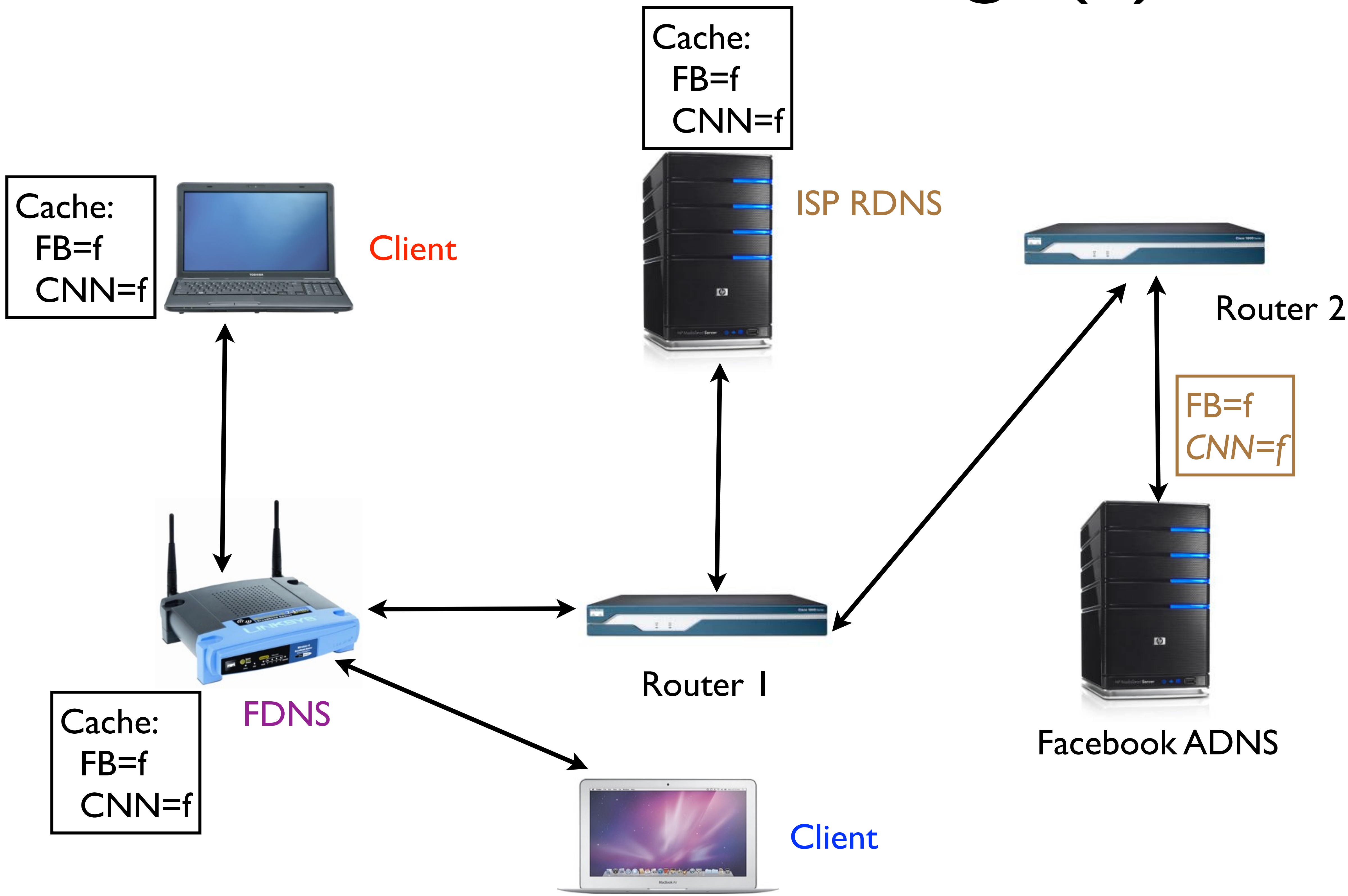
International Computer Science Institute

EECS 325 / 425

November 2018

"Like a preacher stealin' hearts in a travelin' show …"

What Can Go Wrong? (1)



Hostnames in Links

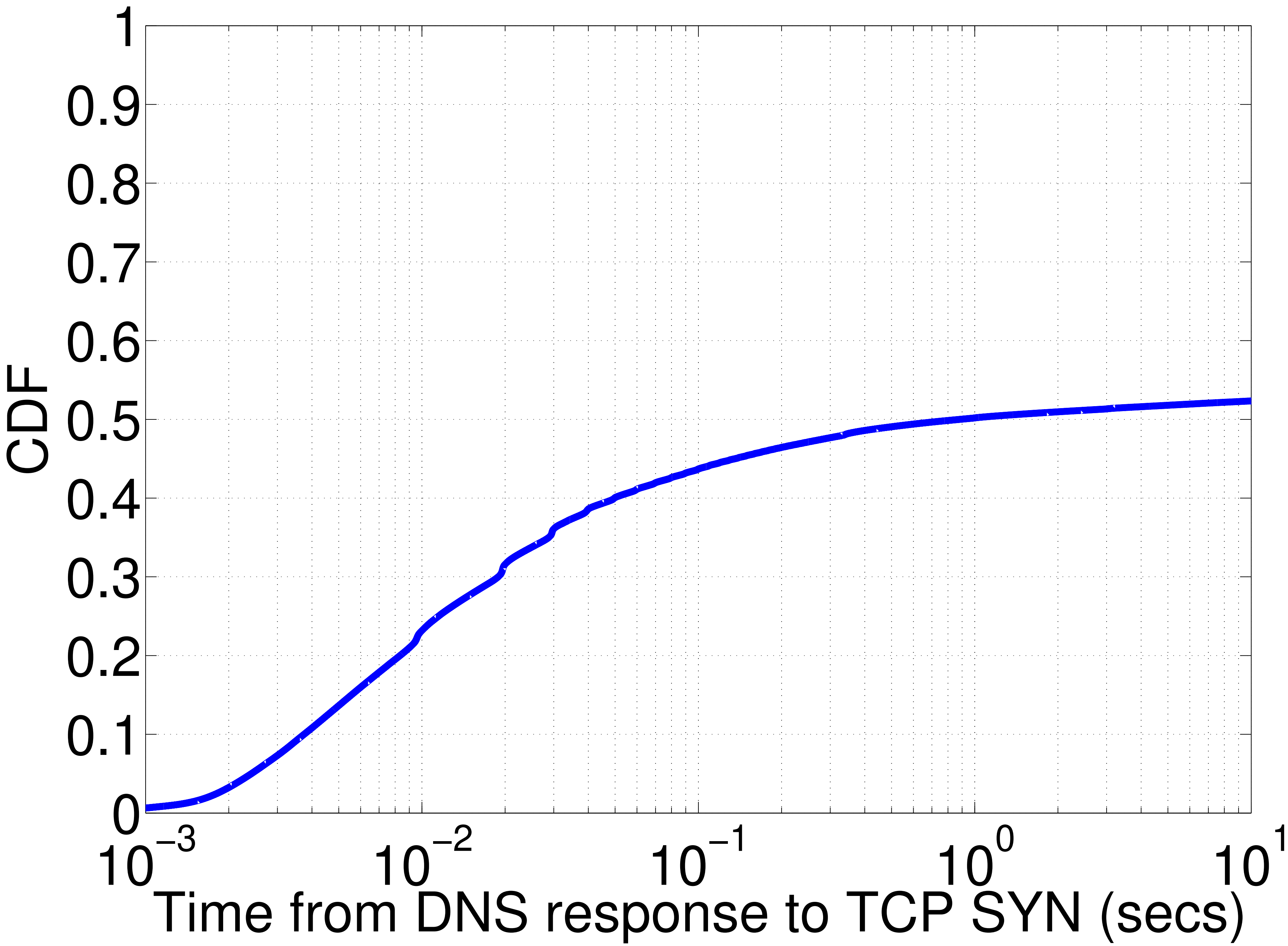
Hostnames in Links

```
lynx -dump -listonly http://www.cnn.com | […] |sort -u
```

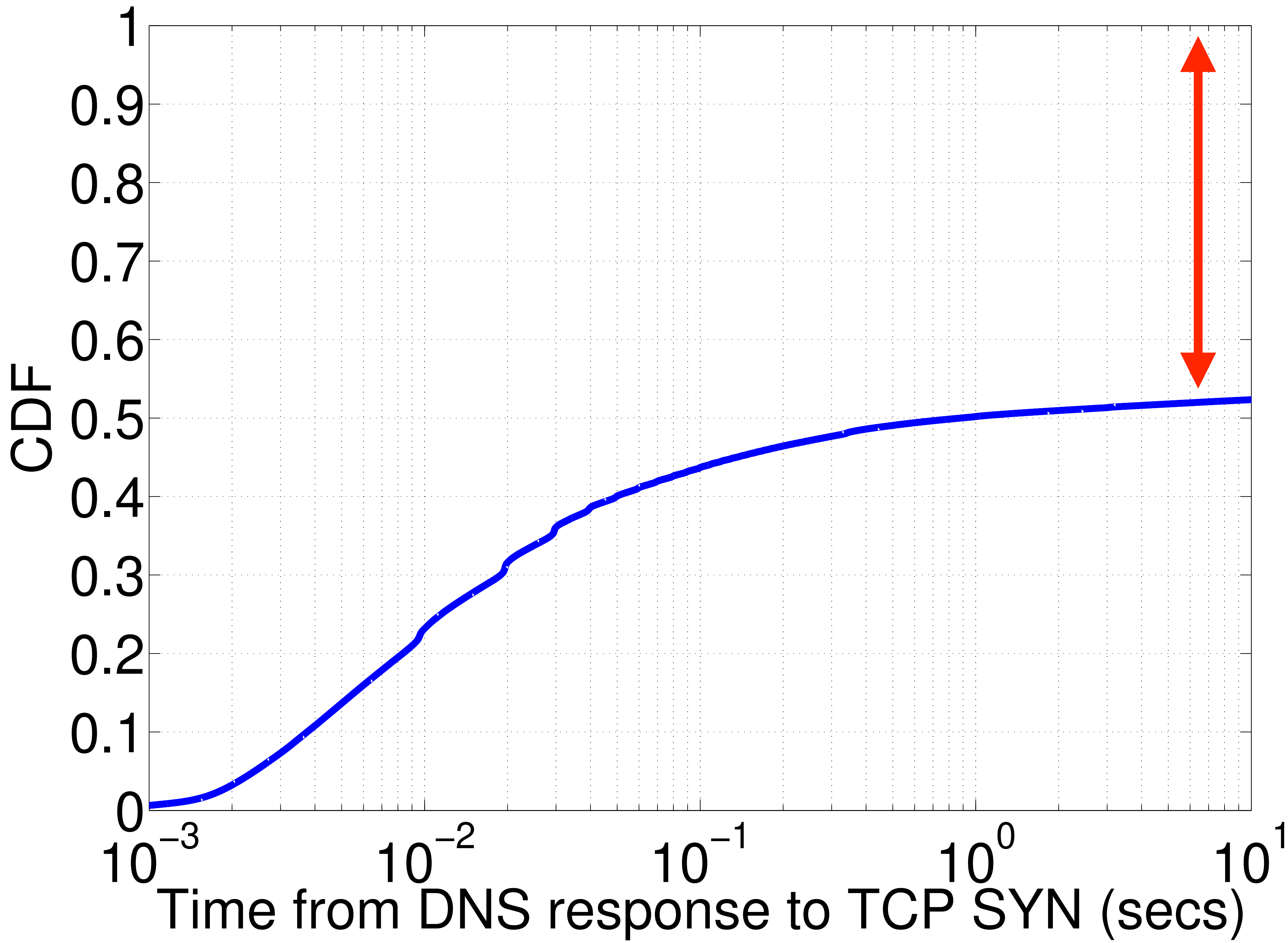
Hostnames in Links

```
lynx -dump -listonly http://www.cnn.com | […] |sort -u
bleacherreport.com
cnn.it
cnnnewsource.com
collection.cnn.com
com.cnn.mobile.android.phone
coupons.cnn.com
edition.cnn.com
instagram.com
money.cnn.com
plus.google.com
store.cnn.com
tours.cnn.com
twitter.com
www.cnn.com
www.facebook.com
www.turner.com
www.turnerjobs.com
```

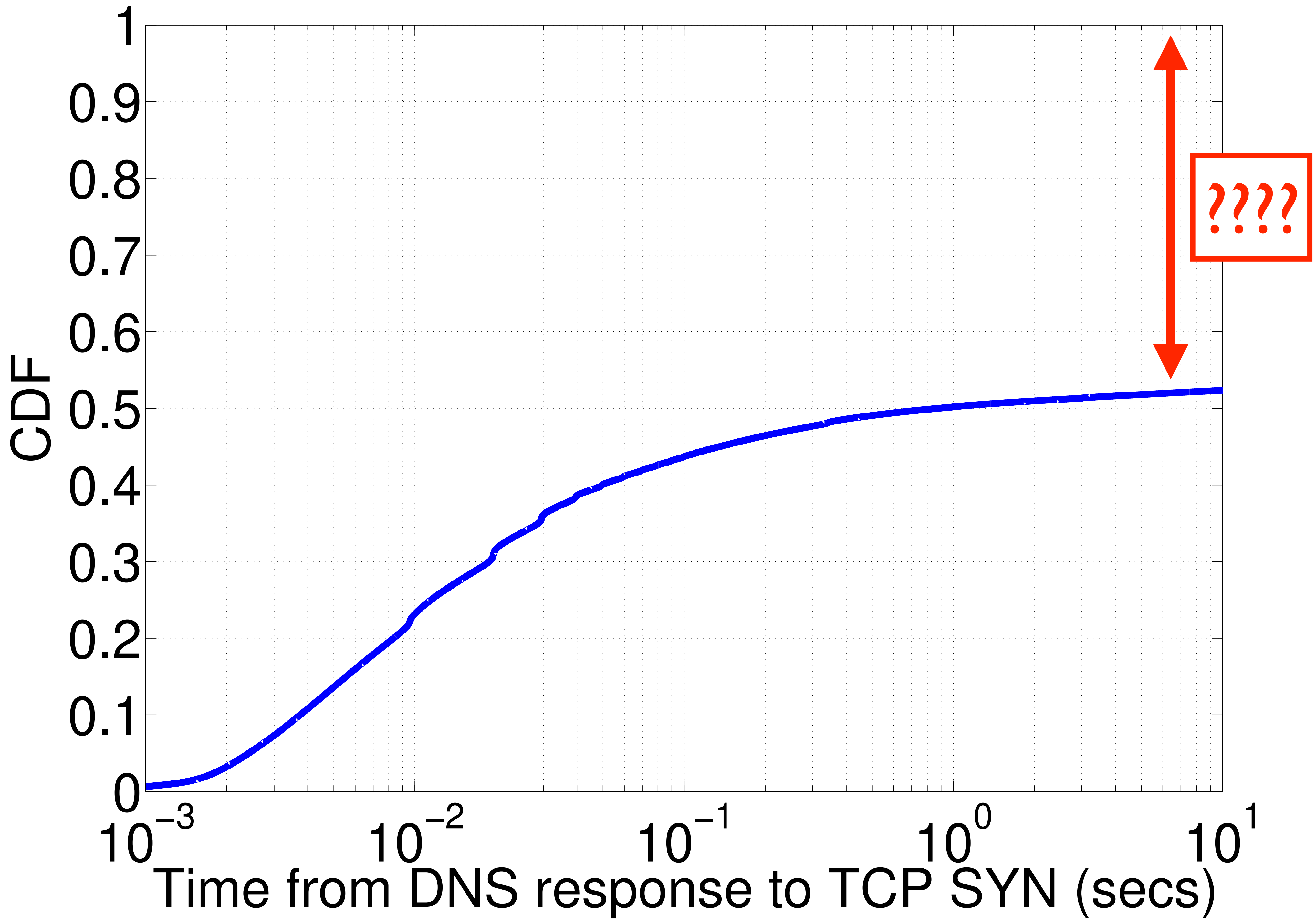
Using DNS Responses



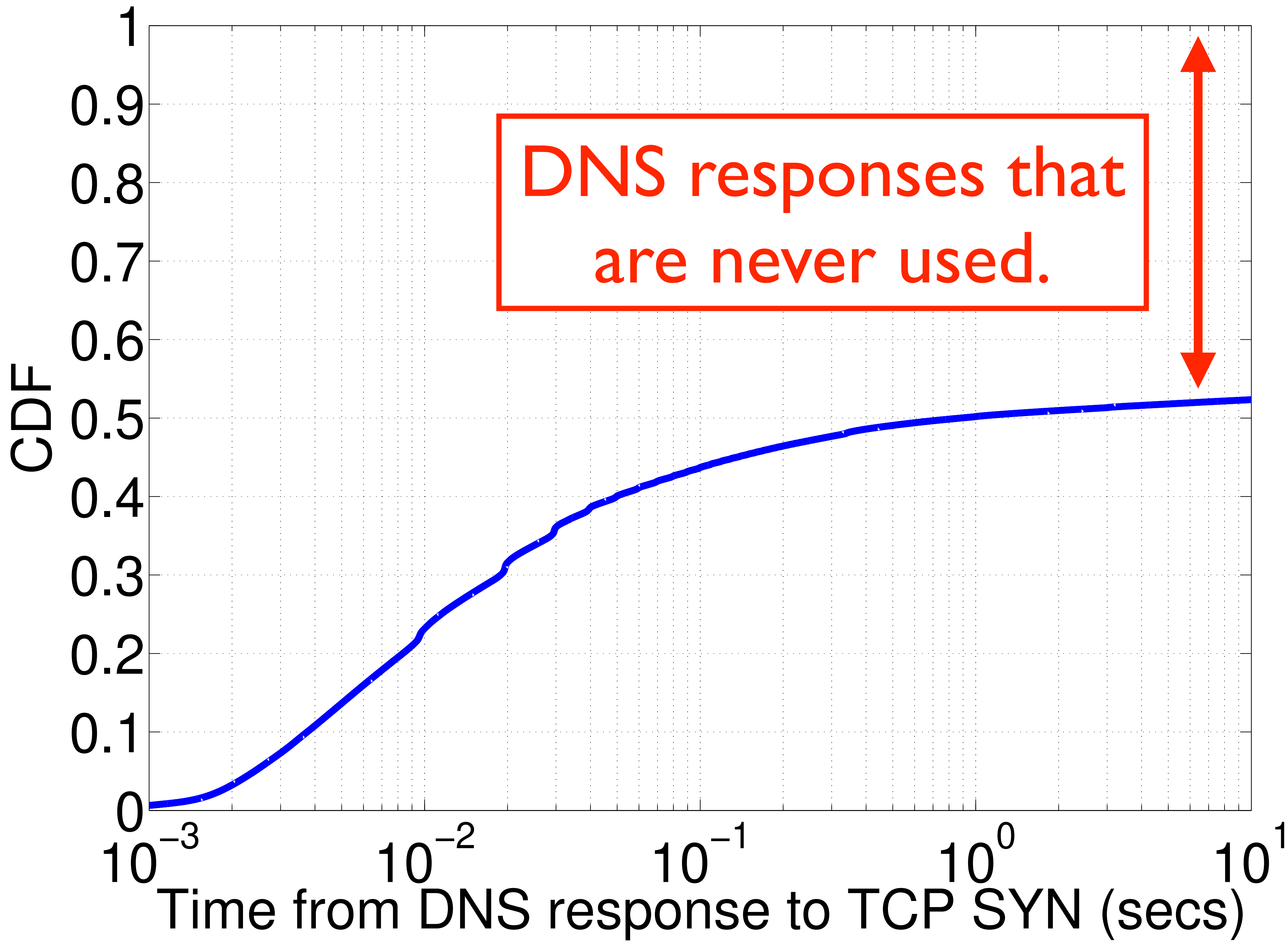
Using DNS Responses



Using DNS Responses



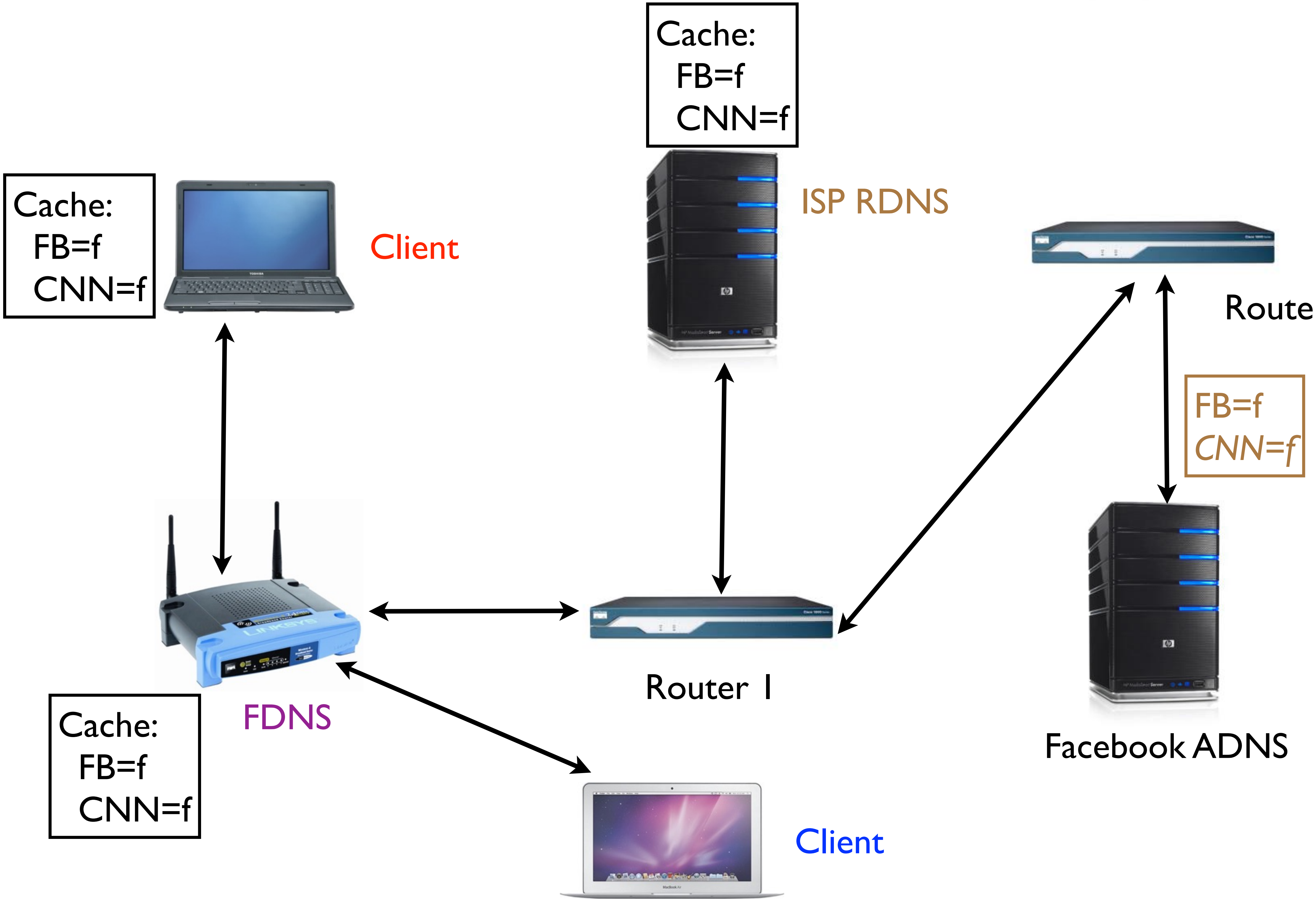
Using DNS Responses



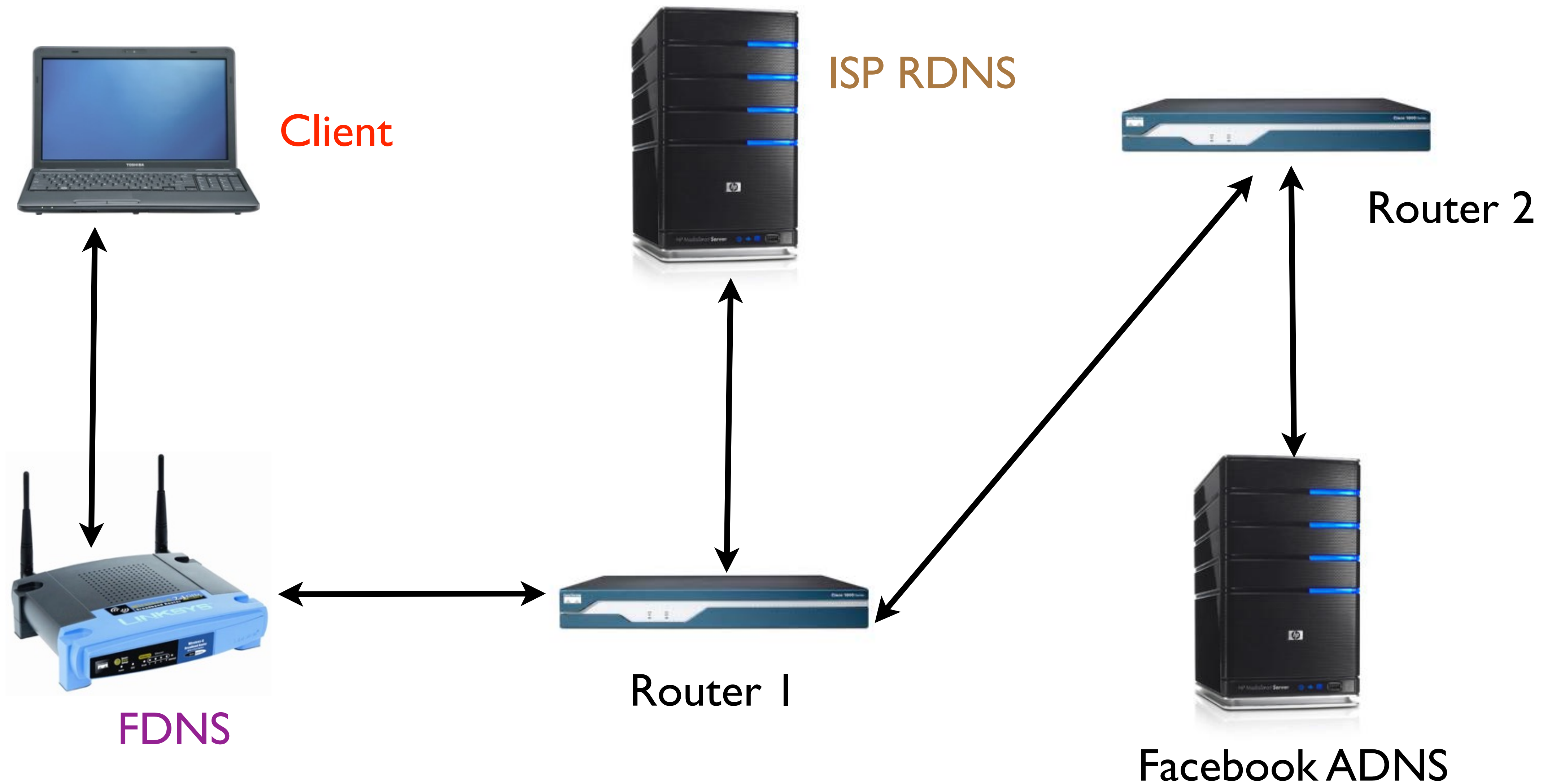
DNS Prefetching

- DNS prefetching is an optimization widely used by modern web browsers
  - (can be disabled)

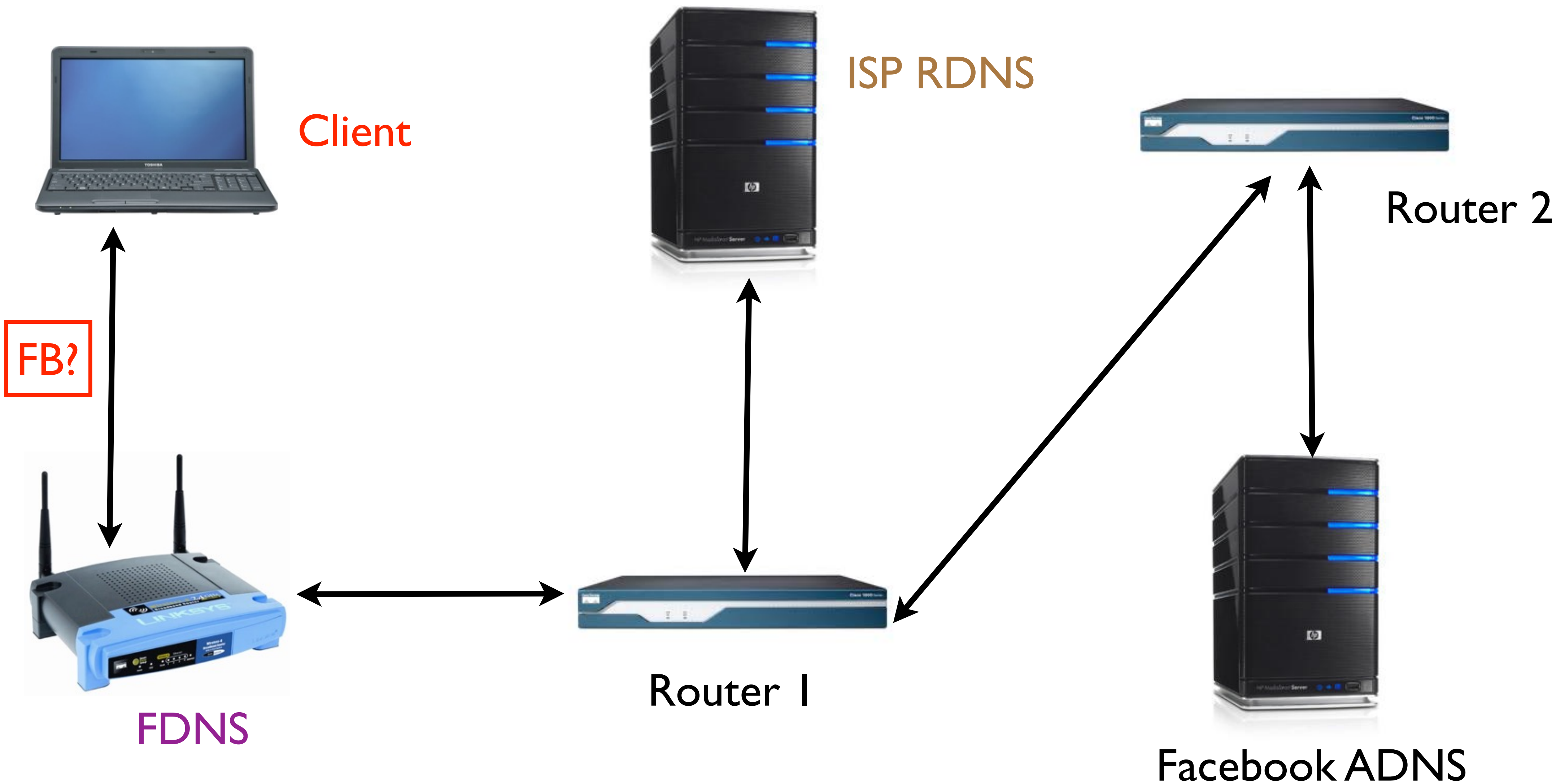
What Can Go Wrong? (1)



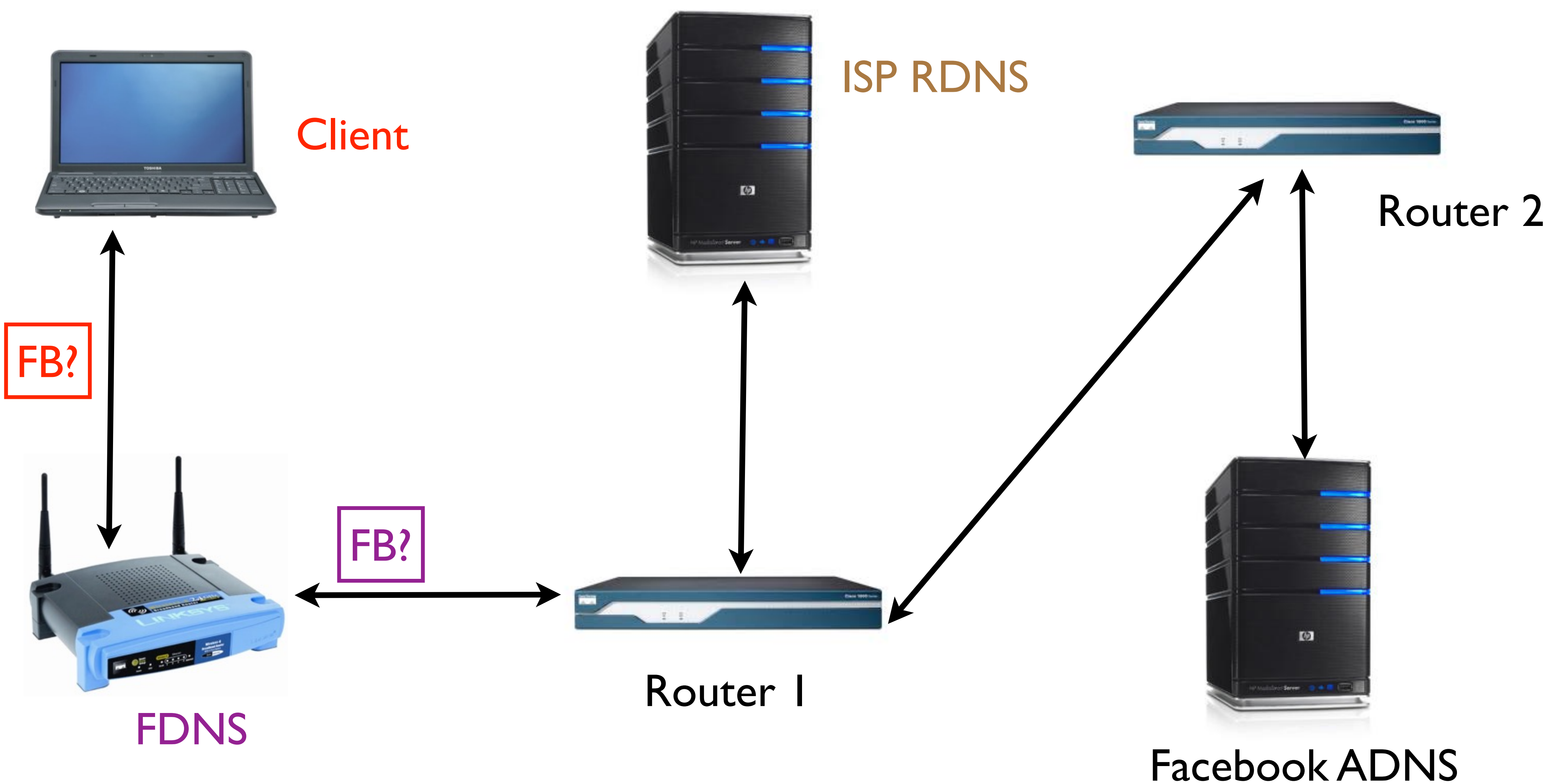
What Can Go Wrong? (2)



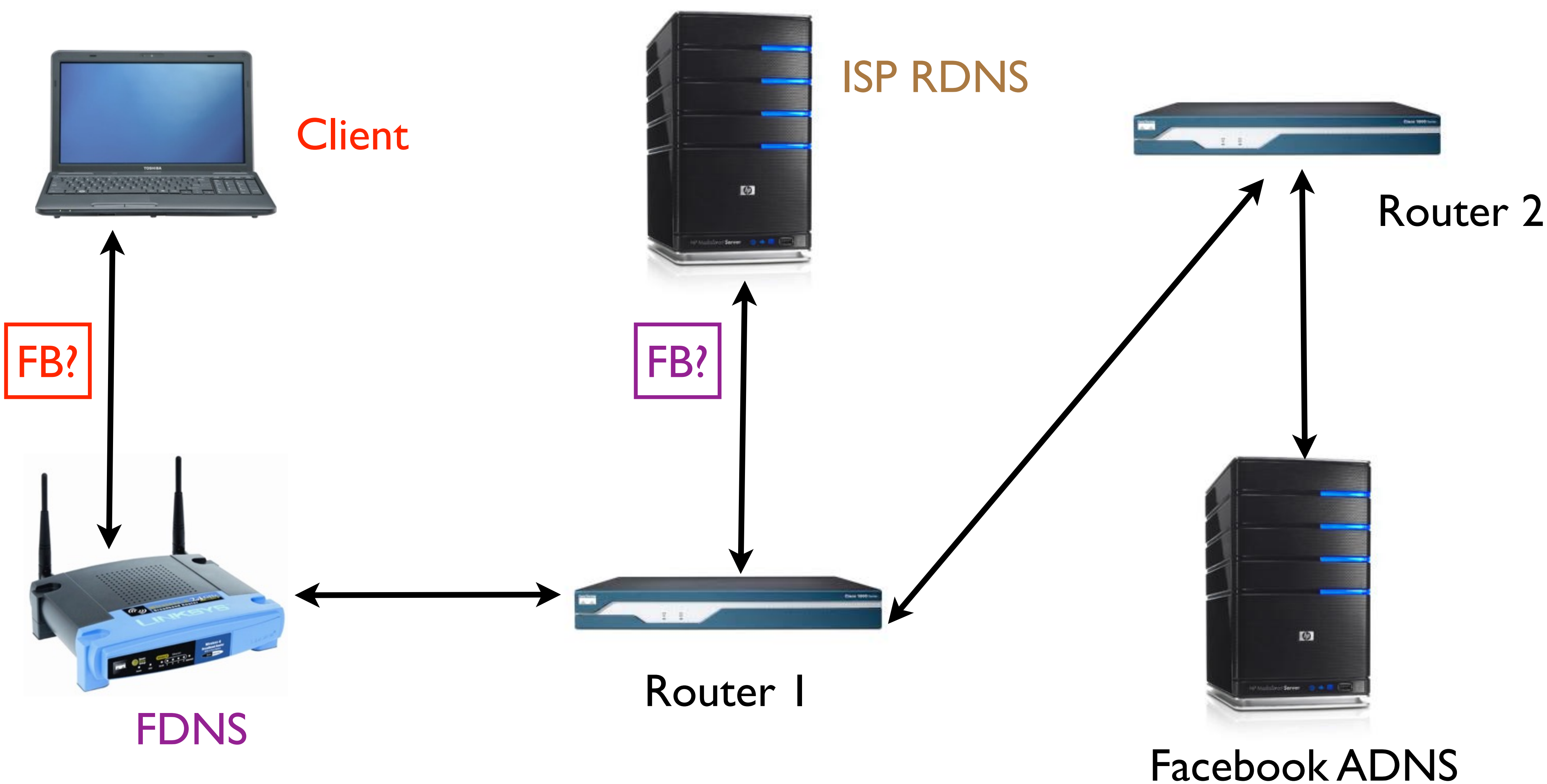
What Can Go Wrong? (2)



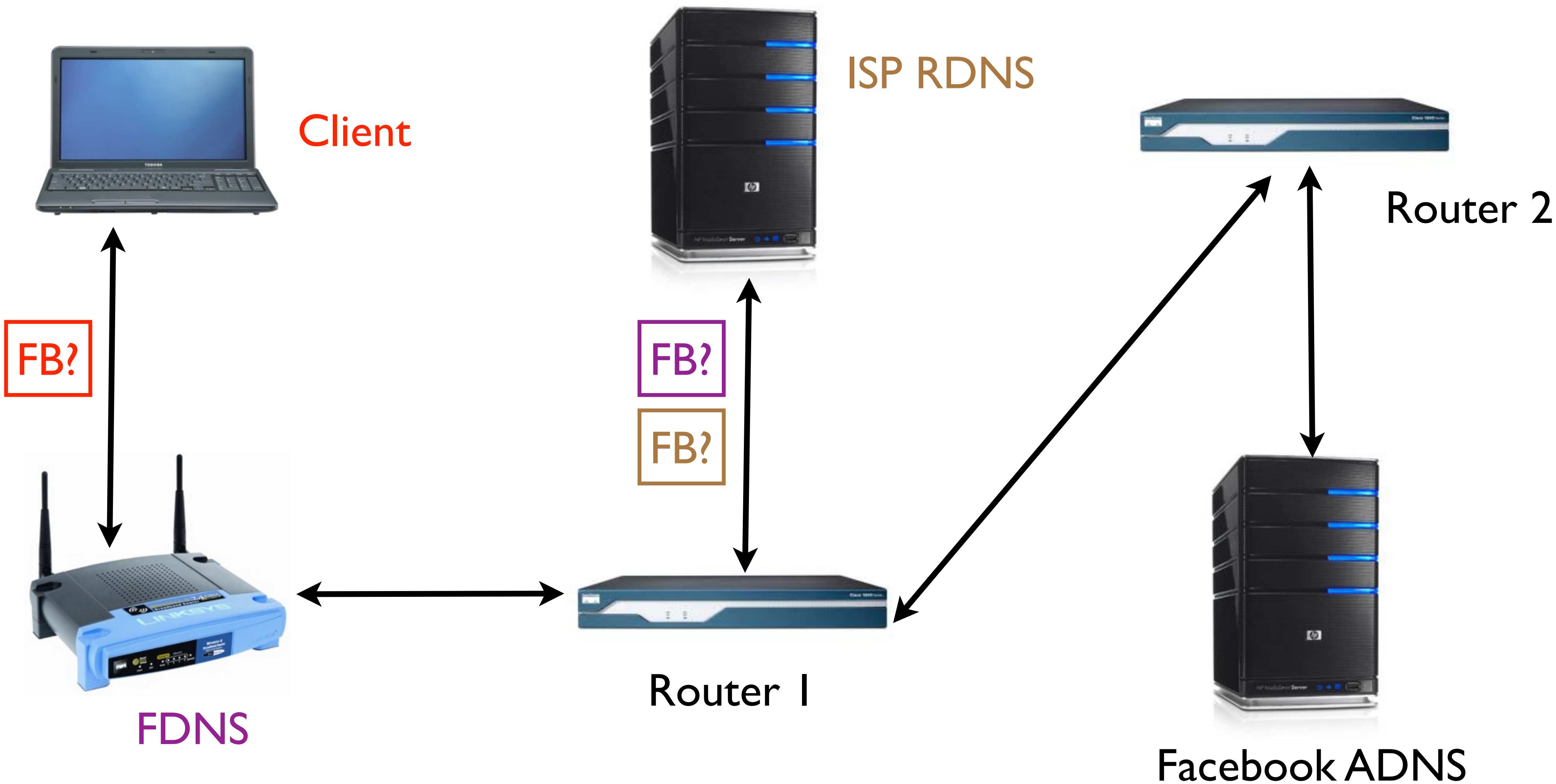
What Can Go Wrong? (2)



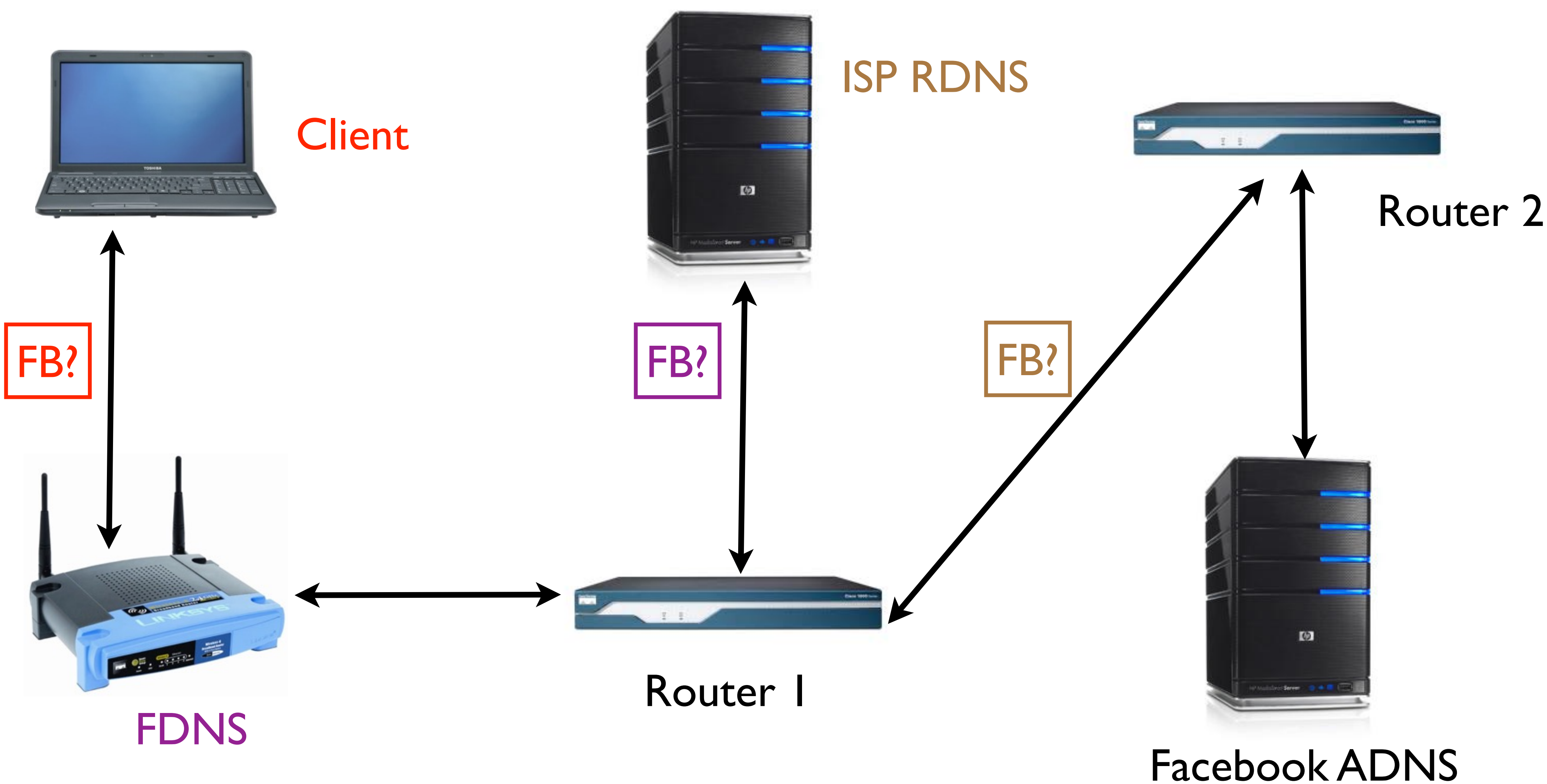
What Can Go Wrong? (2)



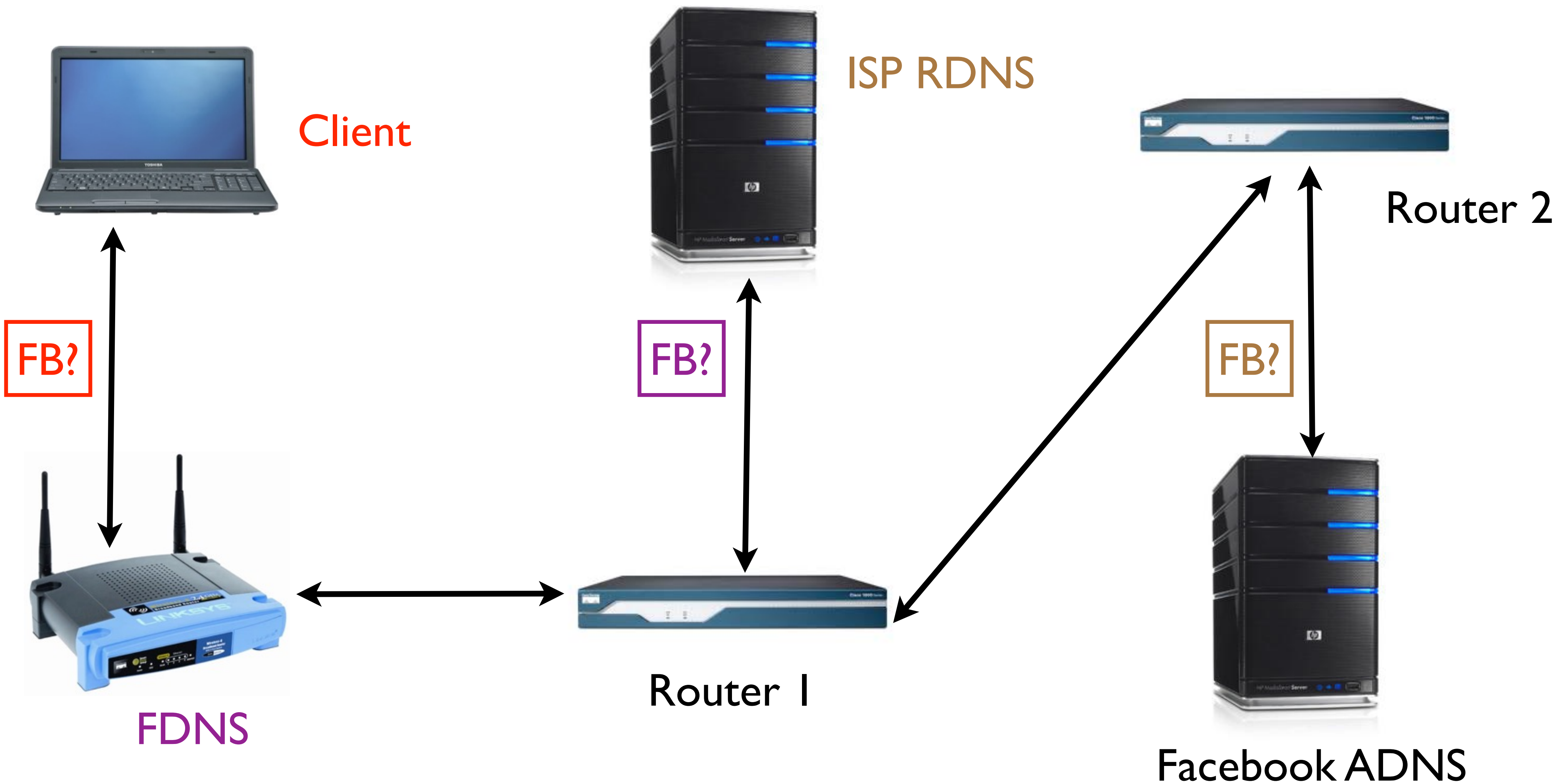
What Can Go Wrong? (2)



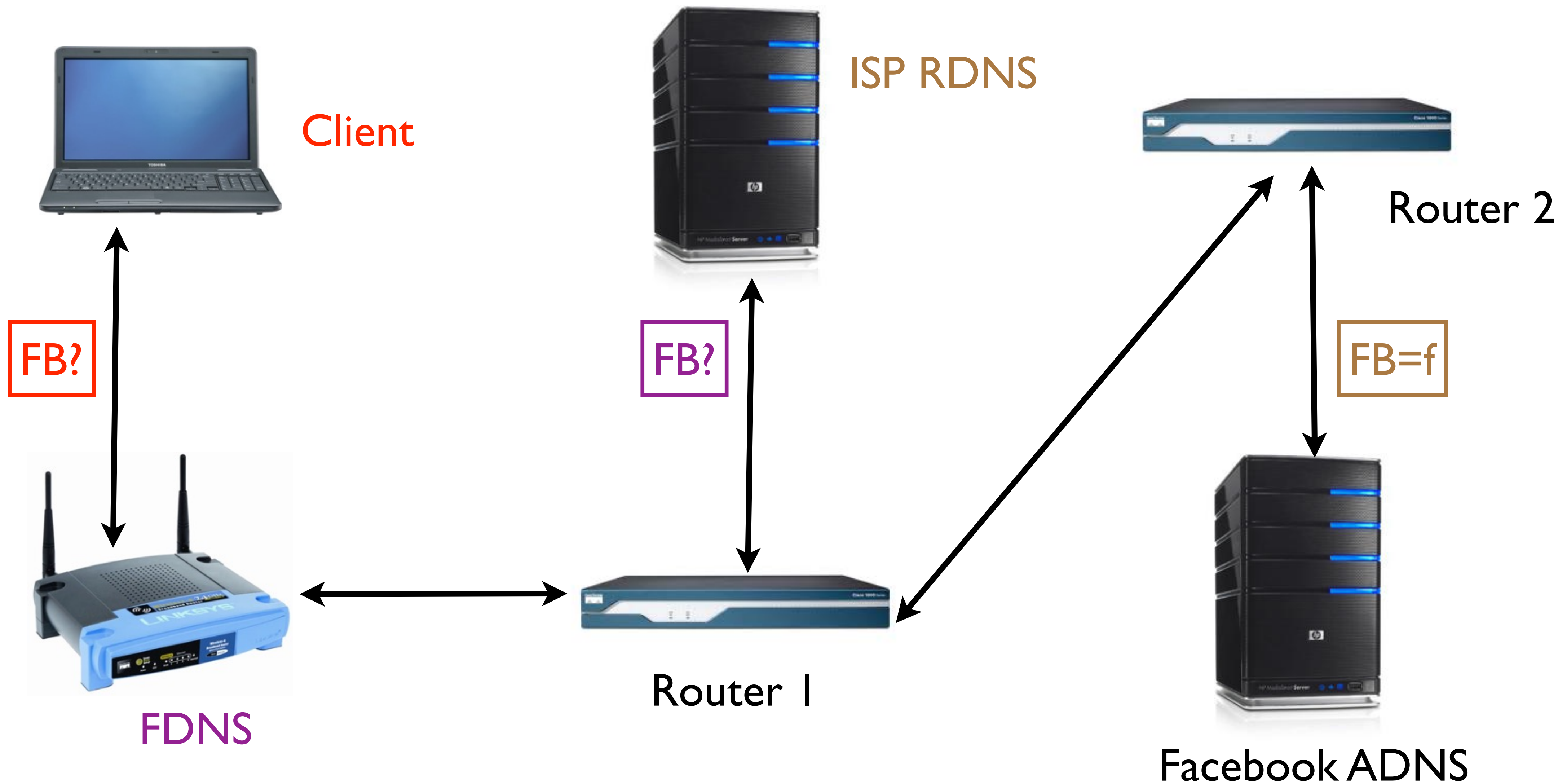
What Can Go Wrong? (2)



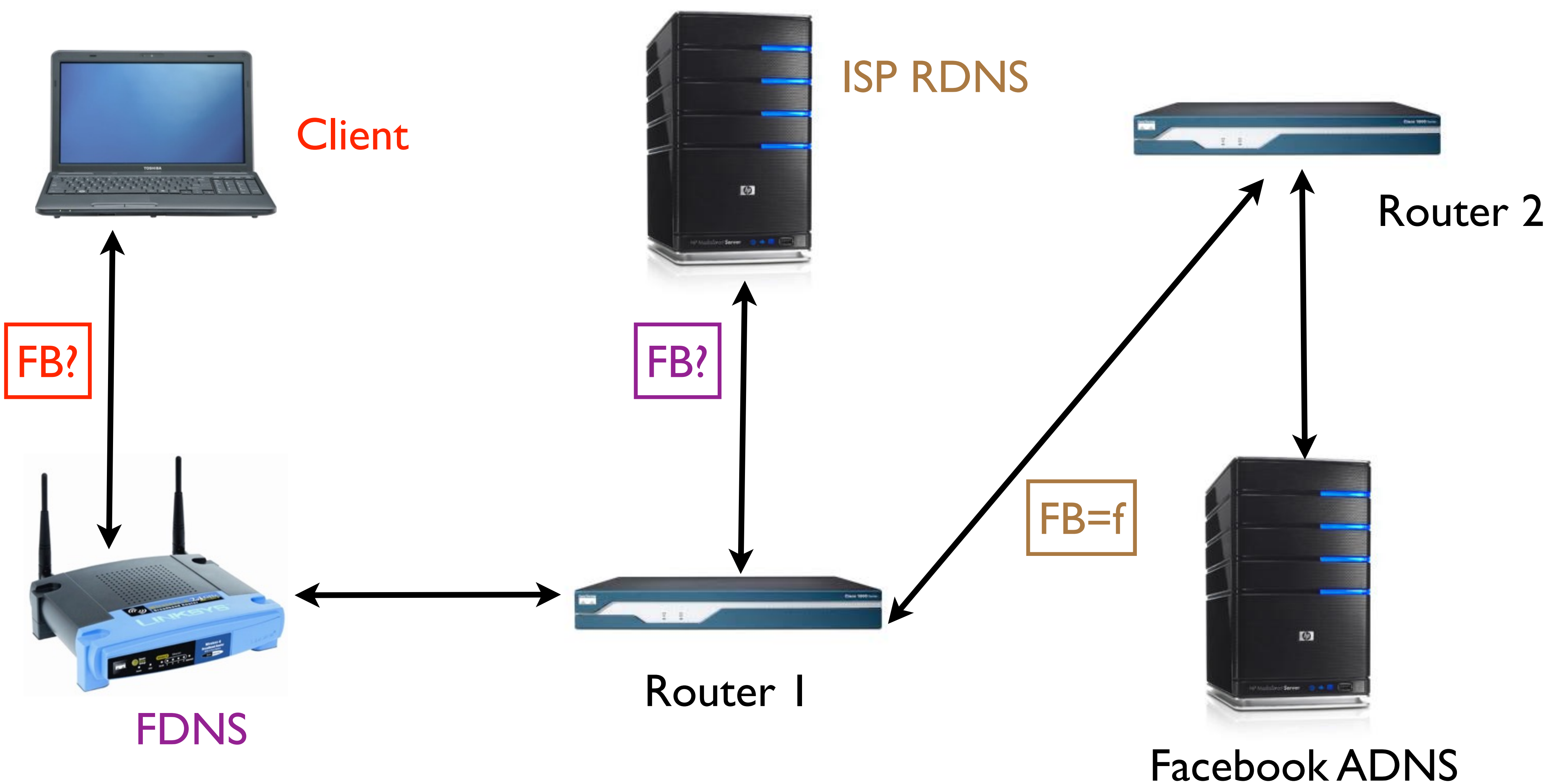
What Can Go Wrong? (2)



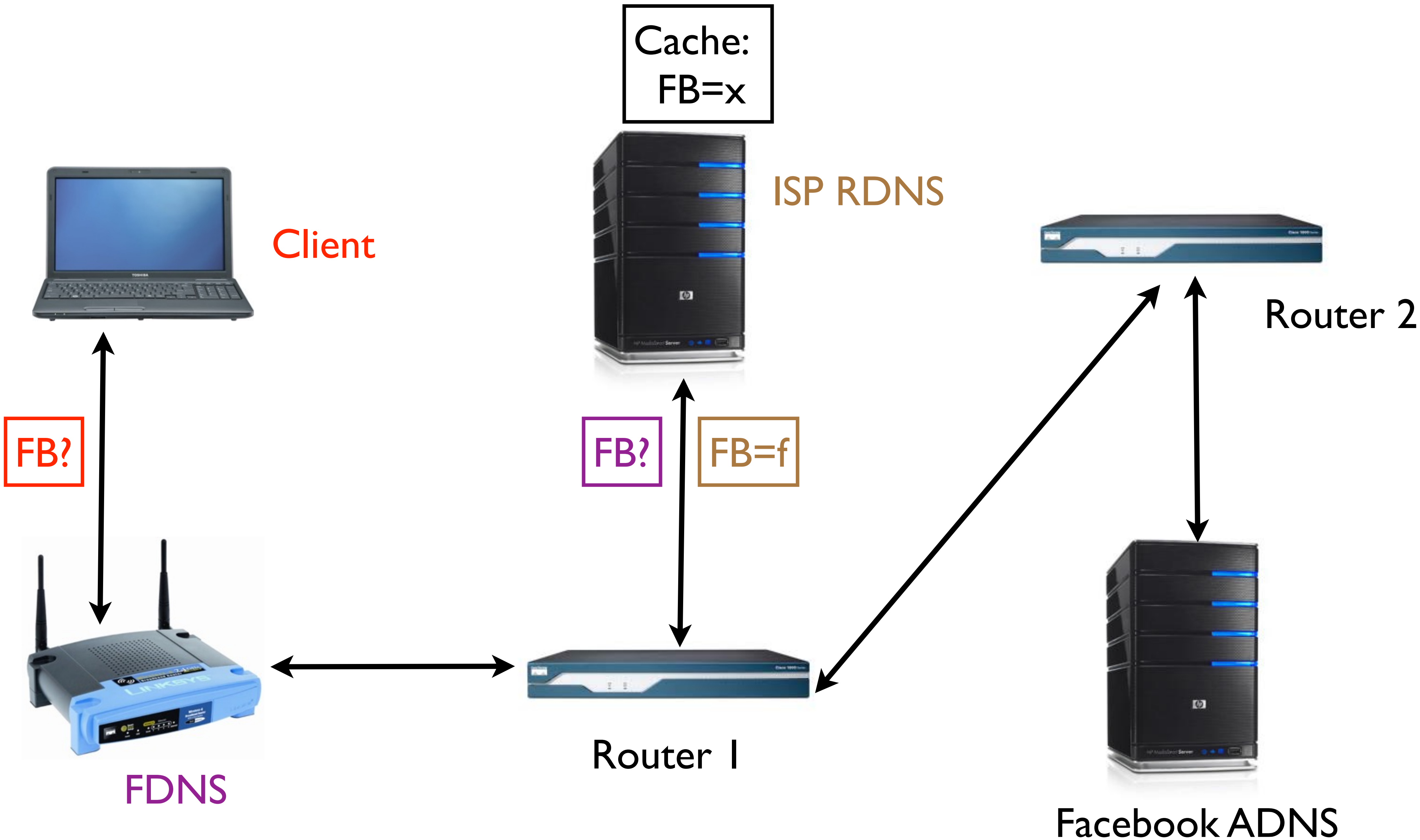
What Can Go Wrong? (2)



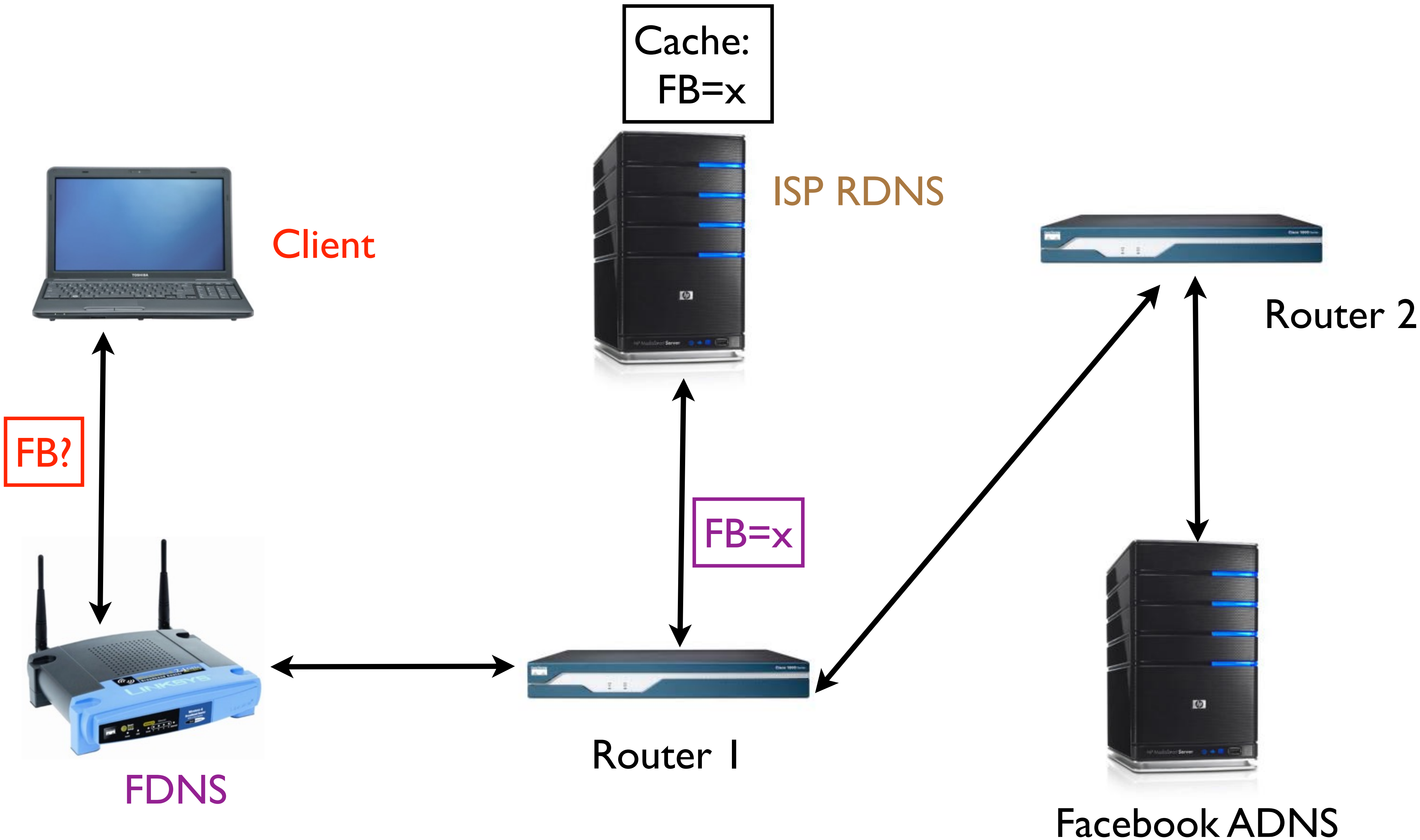
What Can Go Wrong? (2)



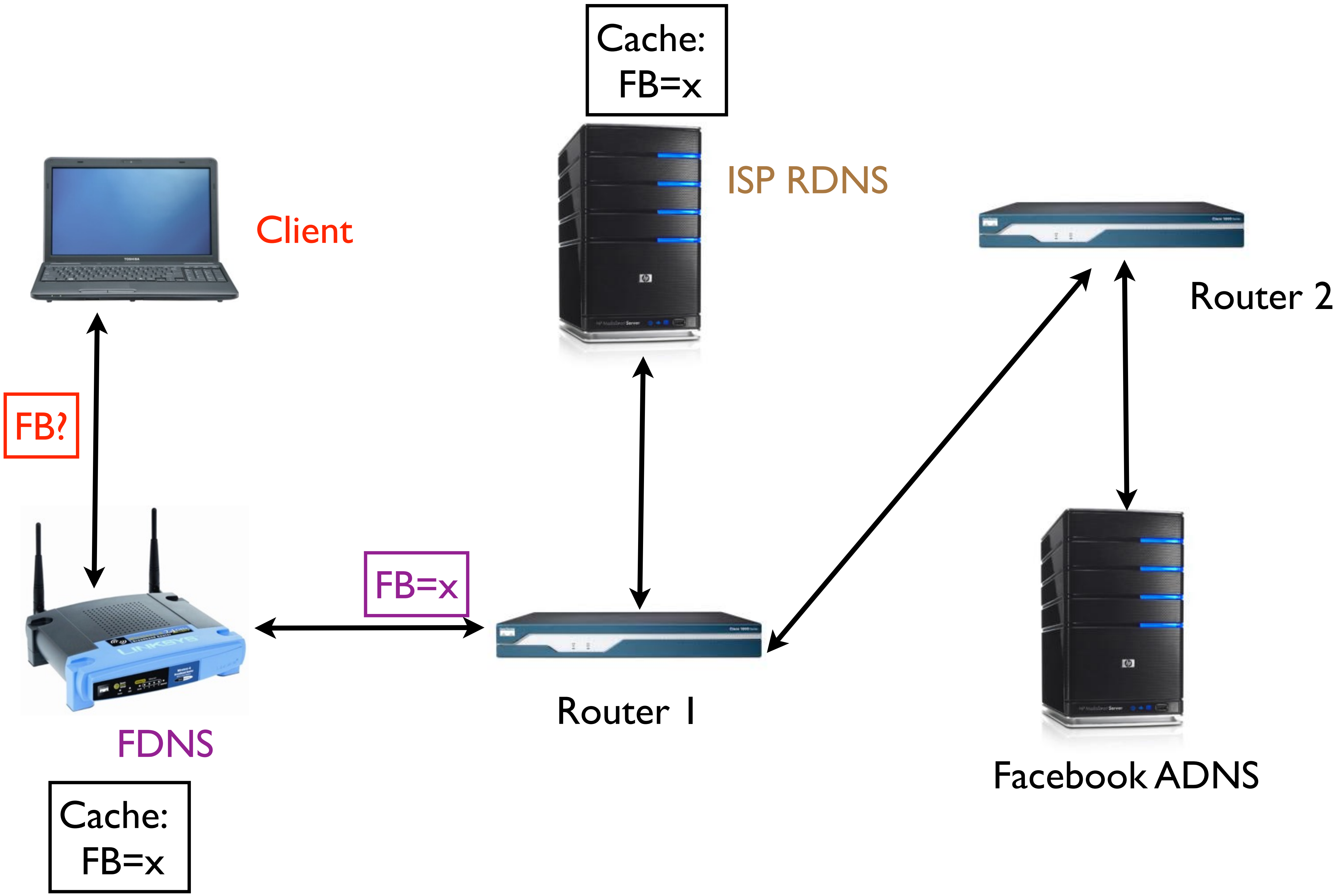
What Can Go Wrong? (2)



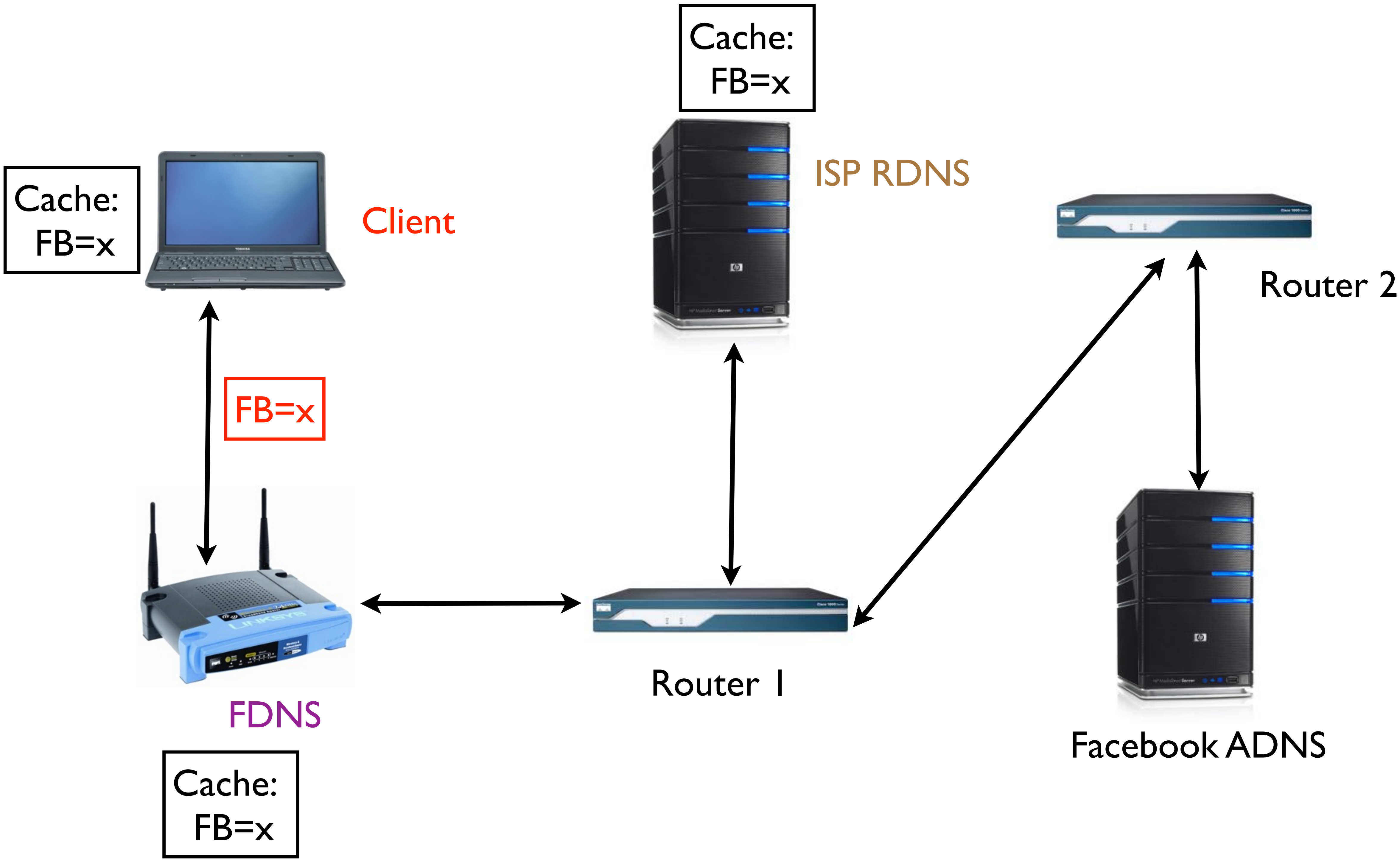
What Can Go Wrong? (2)



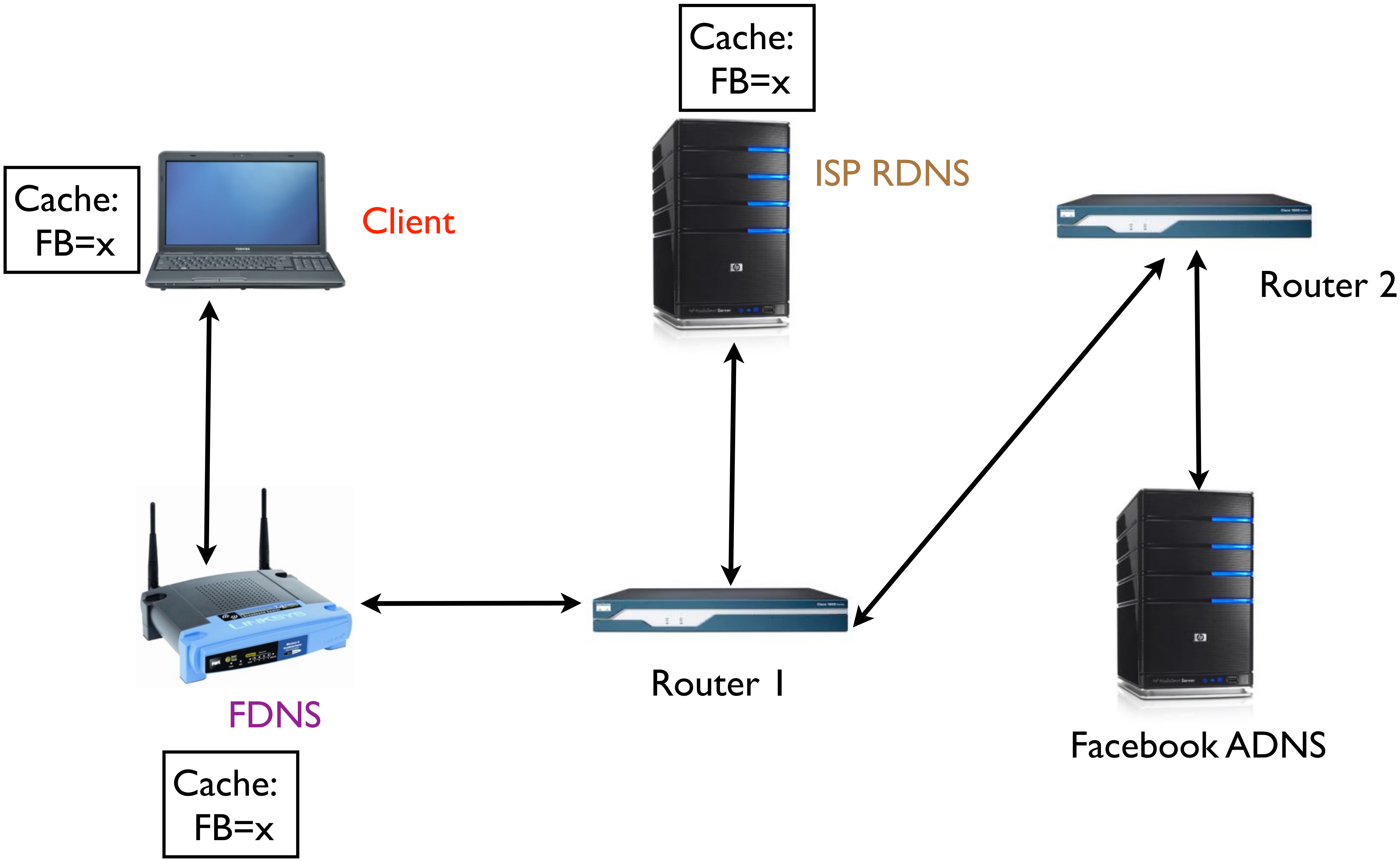
What Can Go Wrong? (2)



What Can Go Wrong? (2)



What Can Go Wrong? (2)



RDNS Rewriting

RDNS Rewriting

- NXDOMAIN re-writing:

RDNS Rewriting

- NXDOMAIN re-writing:
  - e.g., instead of “googgle.com” → ERROR

RDNS Rewriting

- NXDOMAIN re-writing:
  - e.g., instead of "googgle.com" → ERROR
  - .... "googgle.com" → "bing.com"

RDNS Rewriting

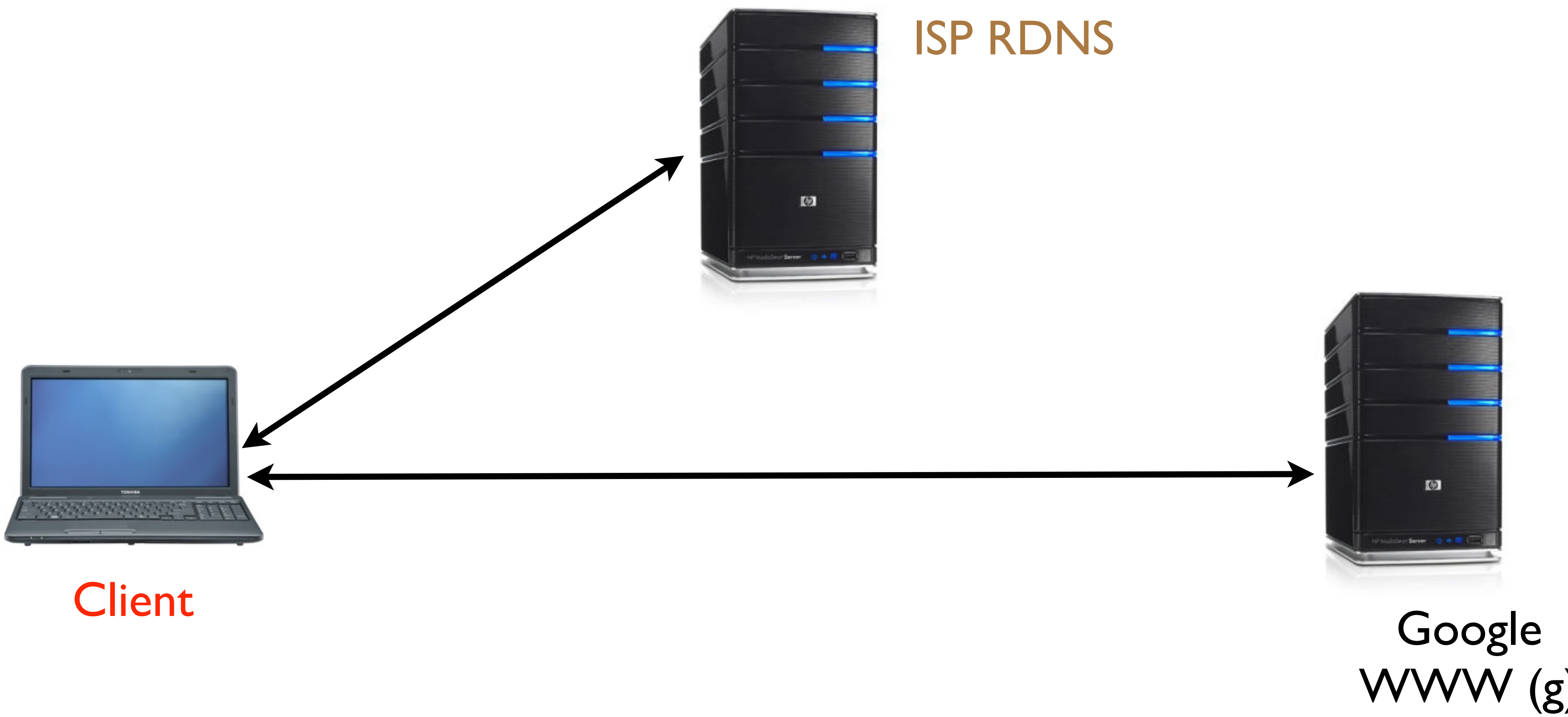
- NXDOMAIN re-writing:
  - e.g., instead of "googgle.com" → ERROR
  - .... "googgle.com" → "bing.com"
  - goal: monetize errors

RDNS Rewriting

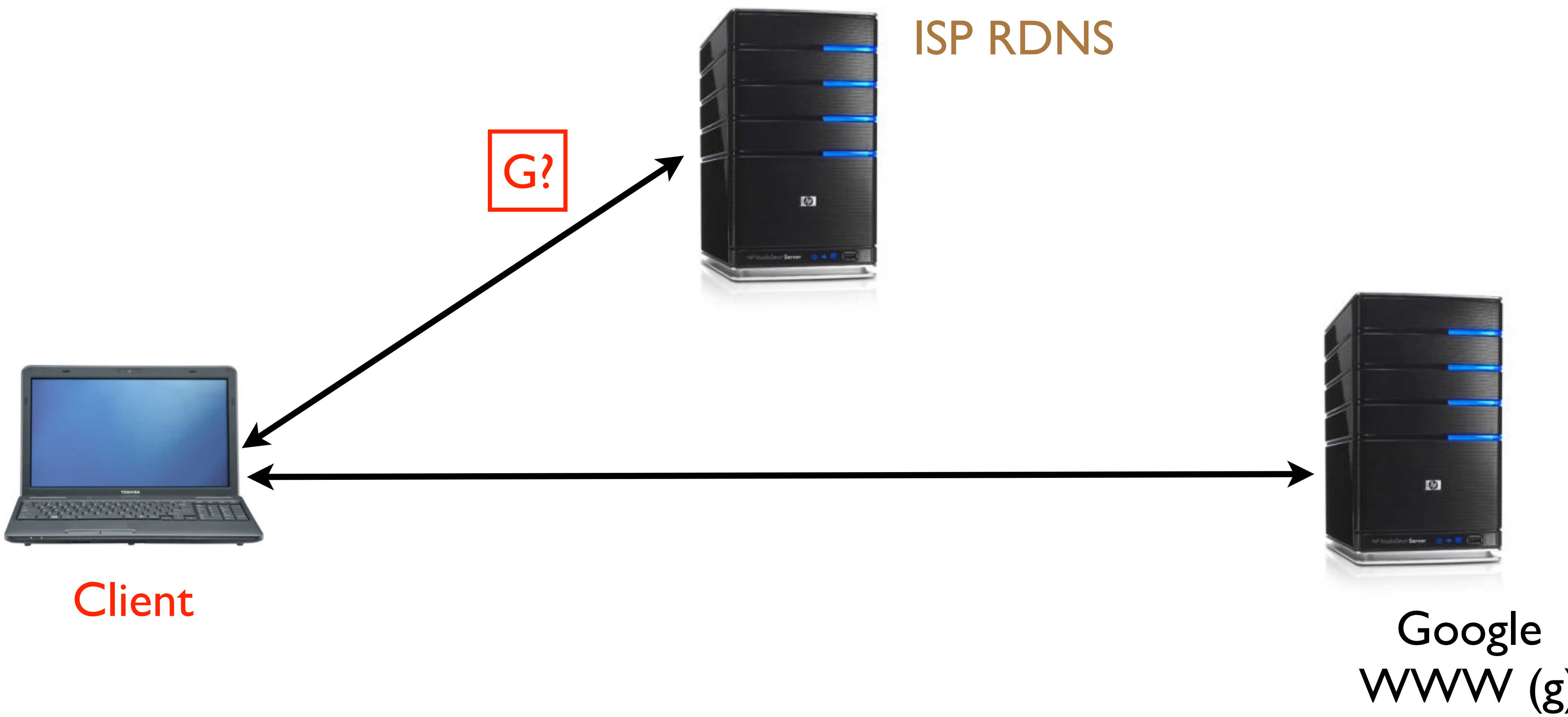
- NXDOMAIN re-writing:
  - e.g., instead of "googgle.com" → ERROR
  - .... "googgle.com" → "bing.com"
  - goal: monetize errors

- Roughly 24% of open resolvers experience NXDOMAIN re-writing
  - across over 100 ISPs

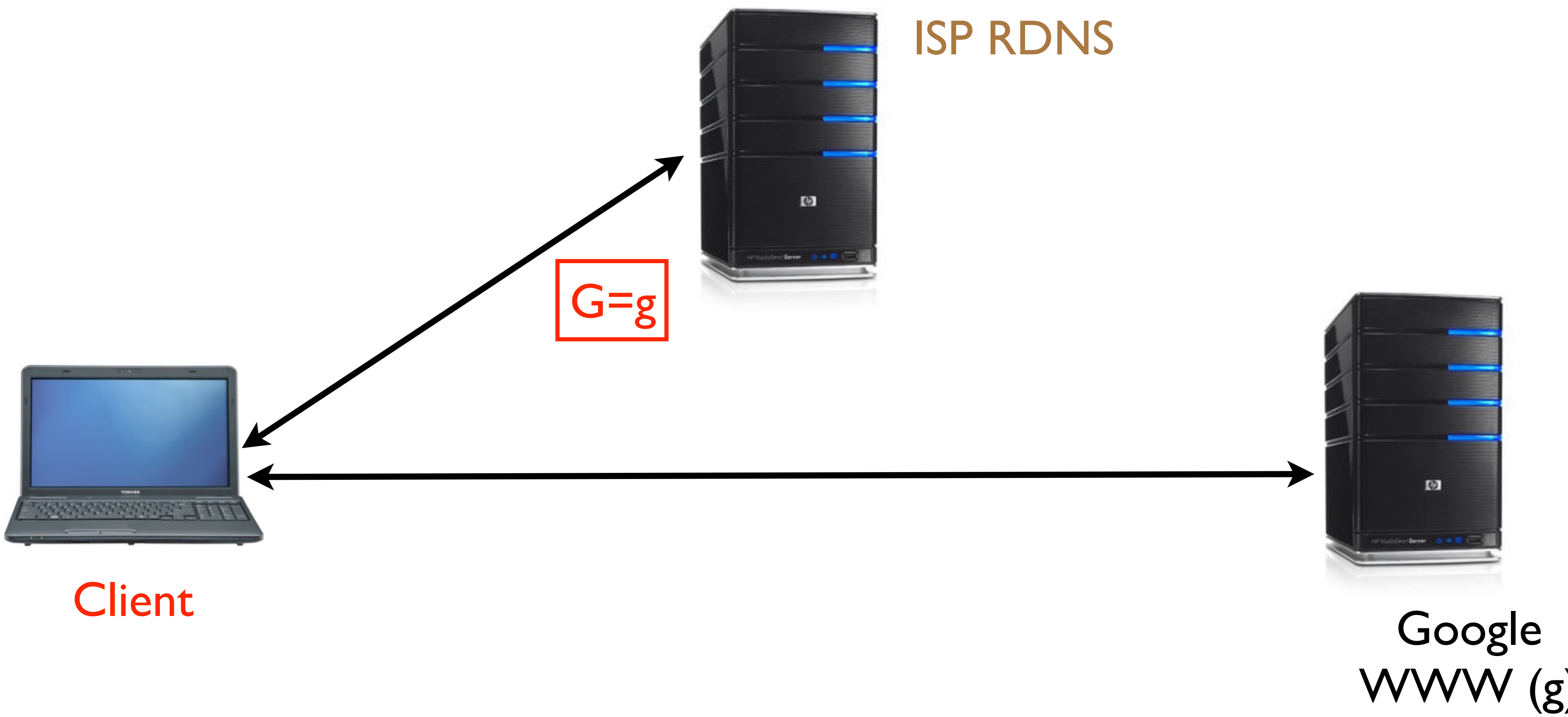
Search Engine Hijacking



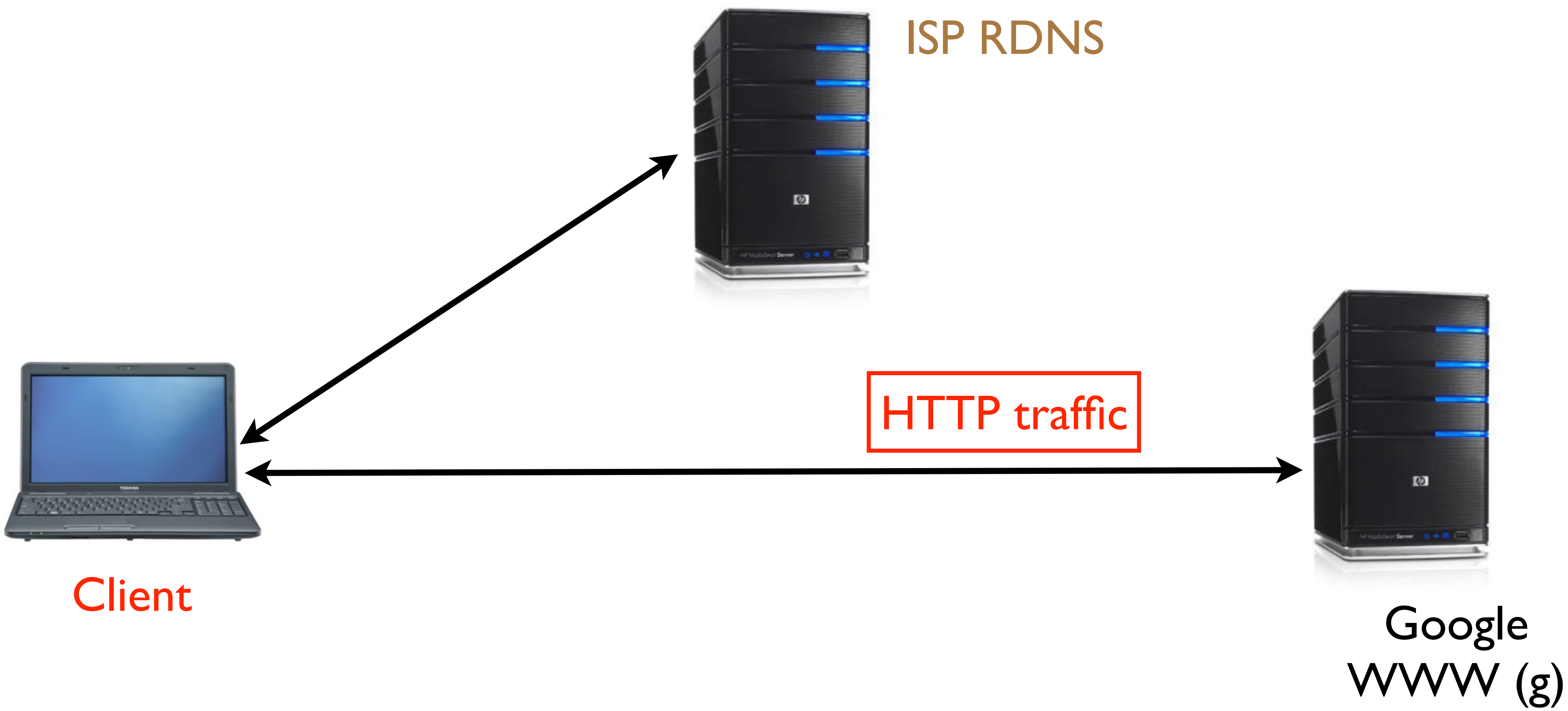
Search Engine Hijacking



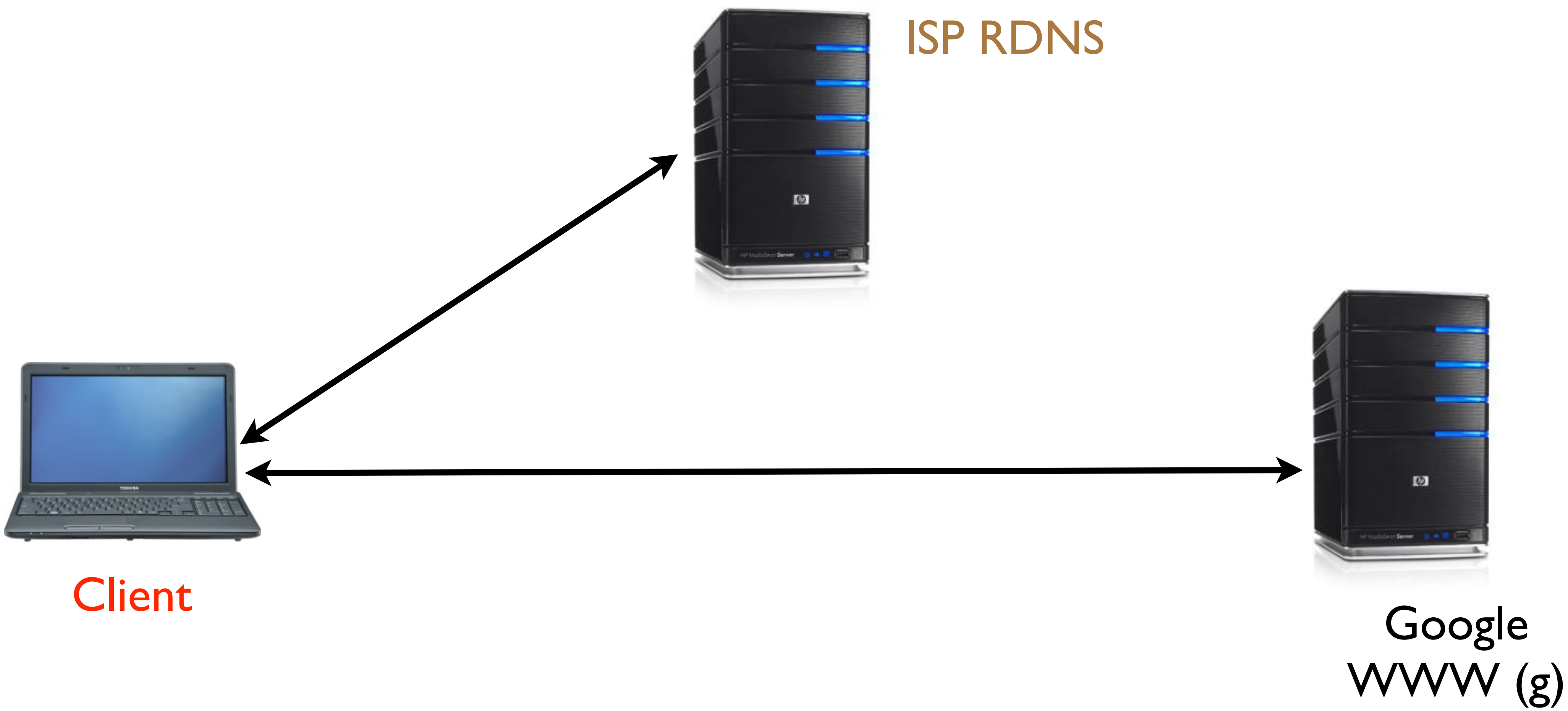
Search Engine Hijacking



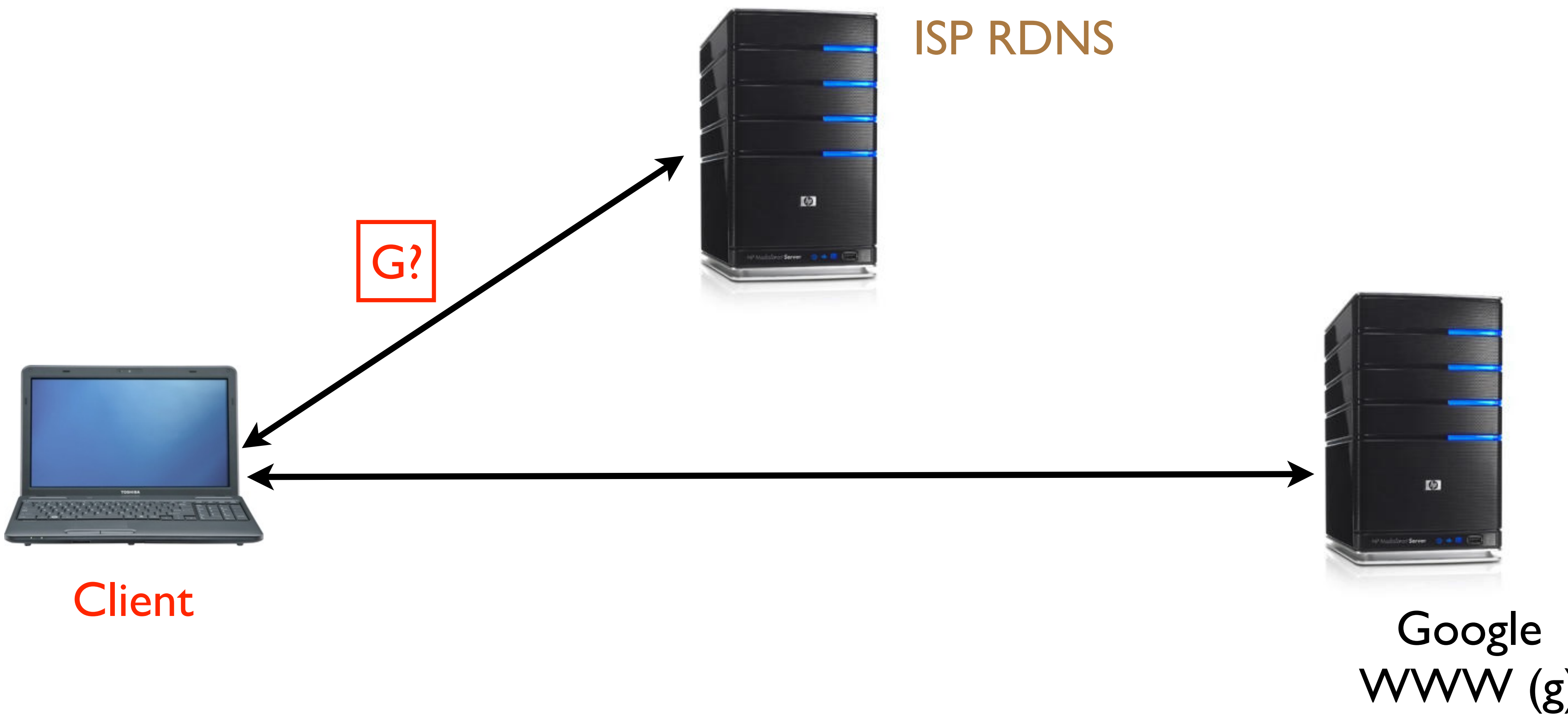
Search Engine Hijacking



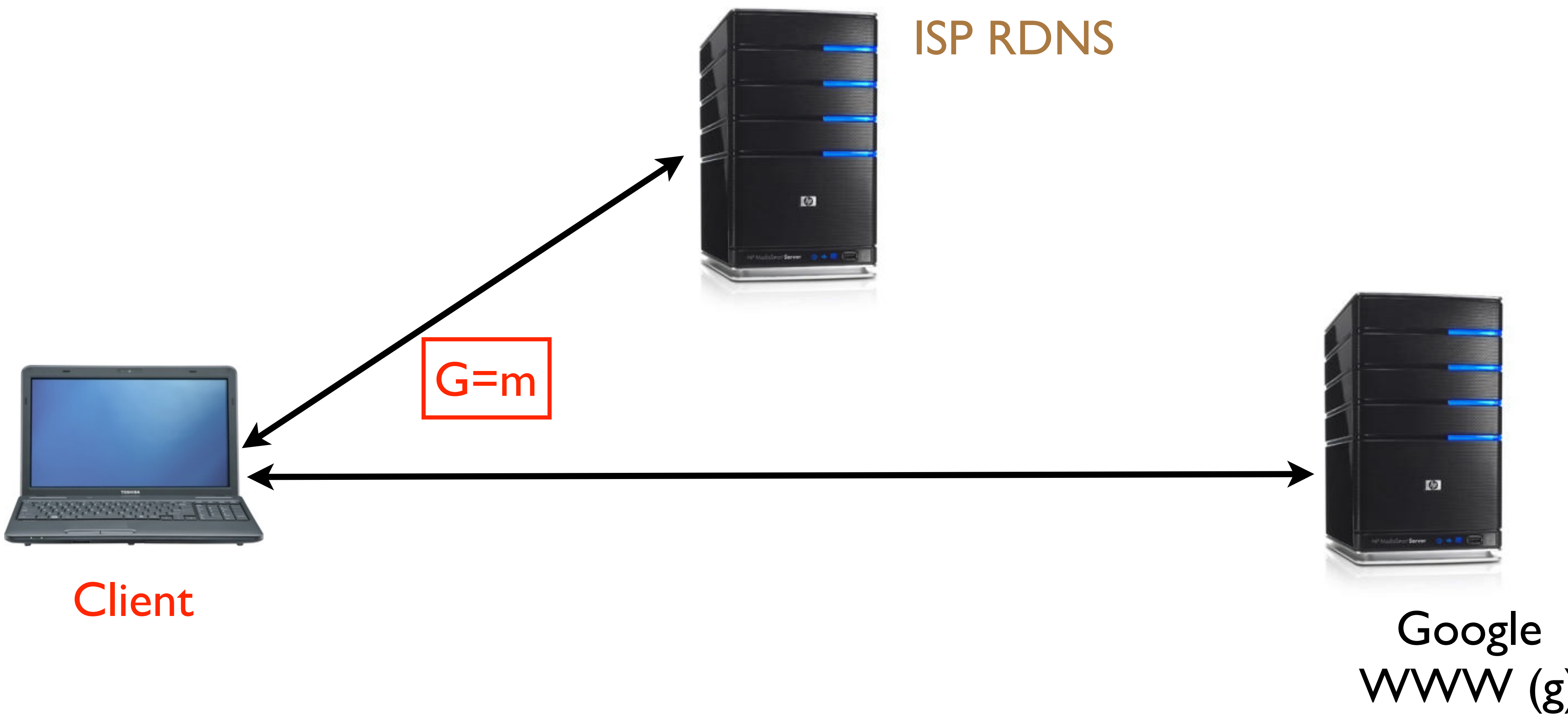
Search Engine Hijacking



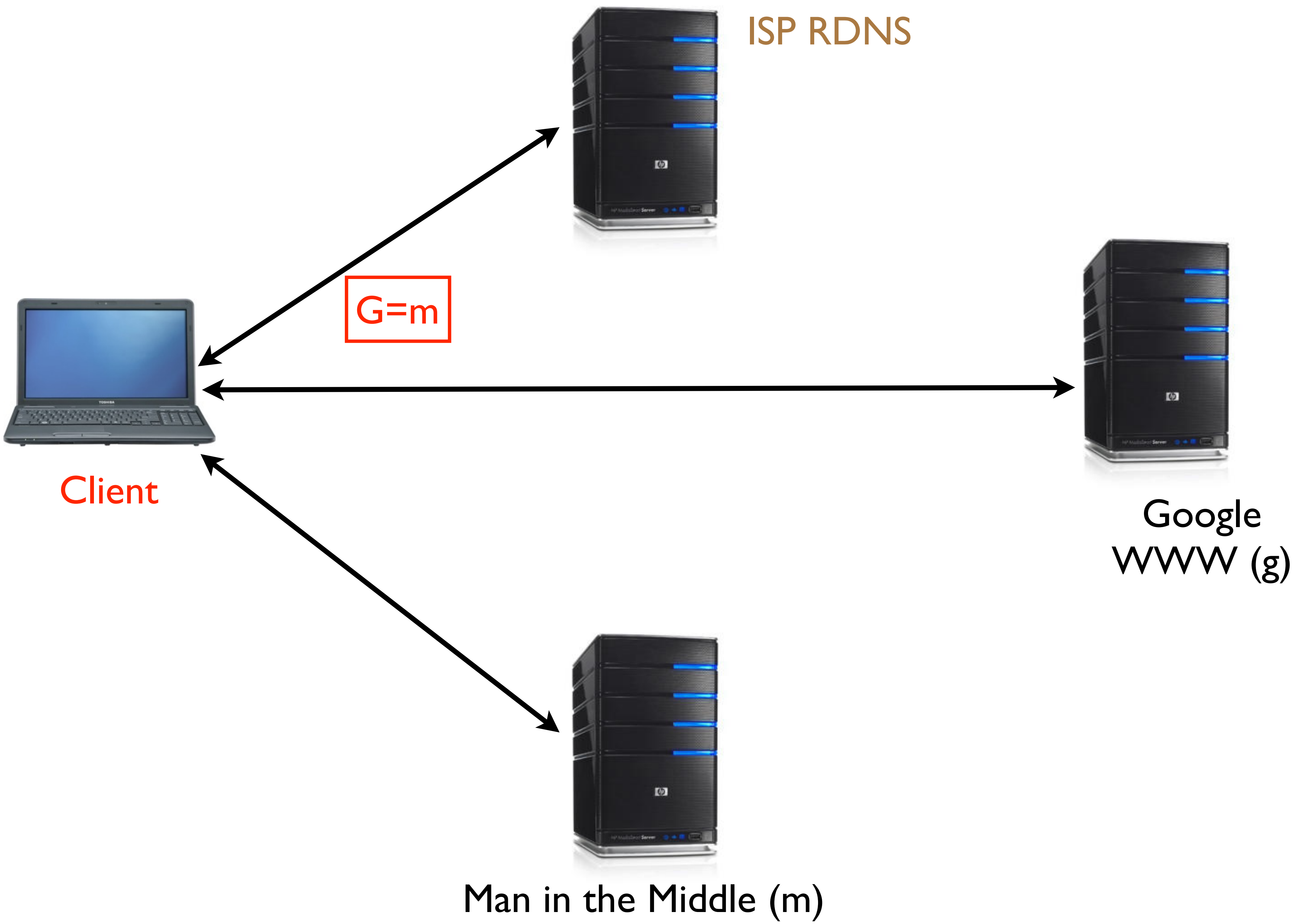
Search Engine Hijacking



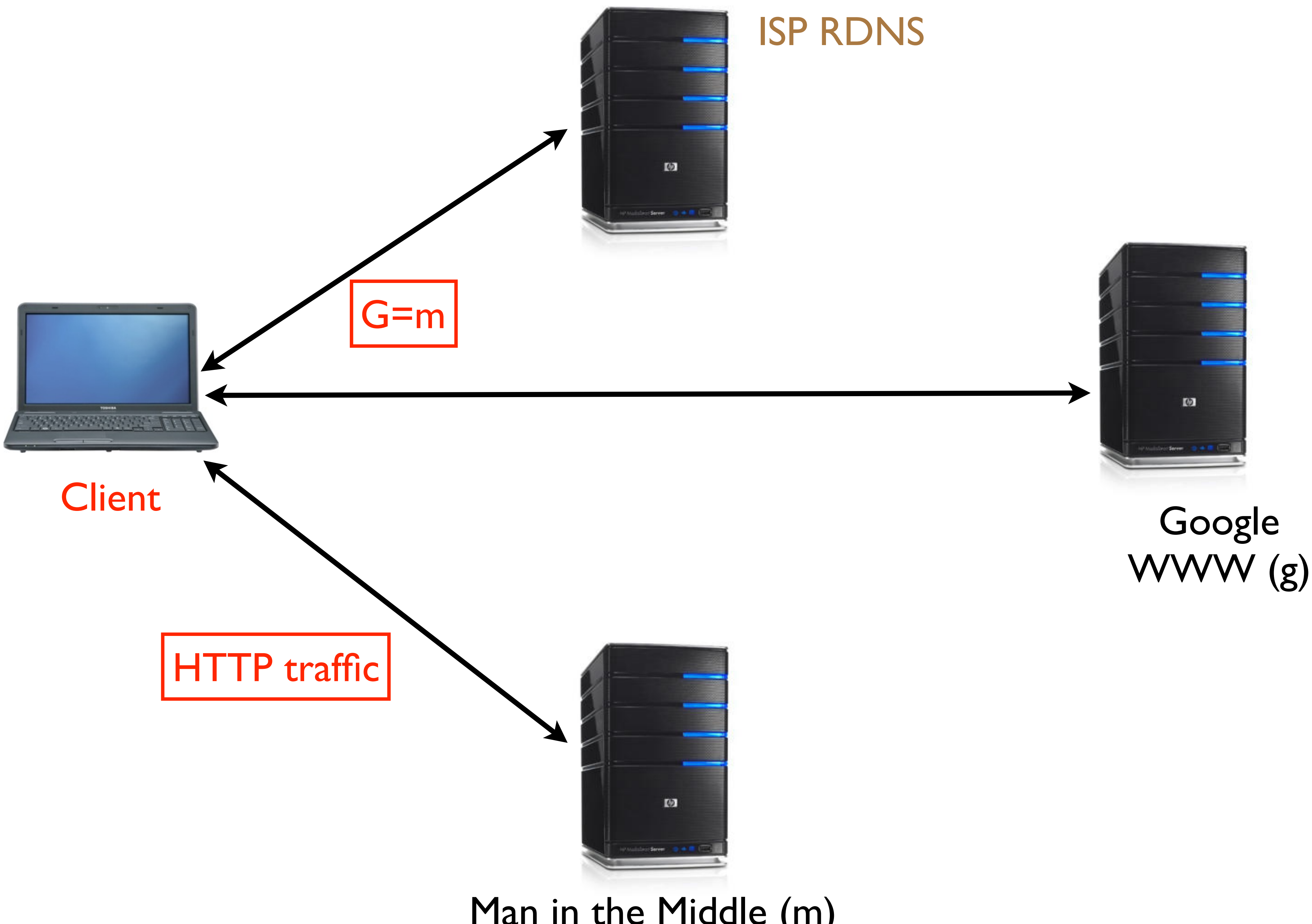
Search Engine Hijacking



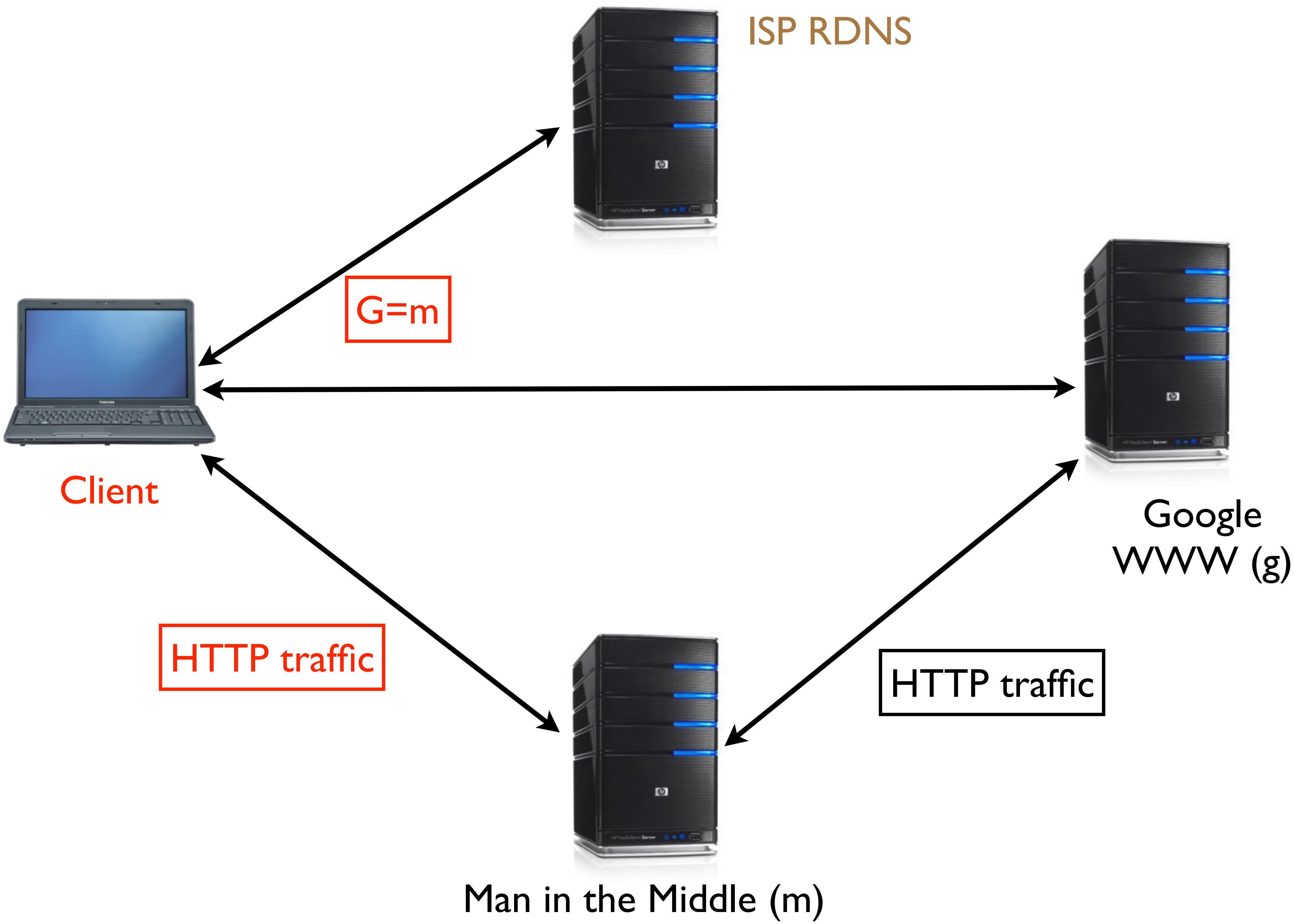
Search Engine Hijacking



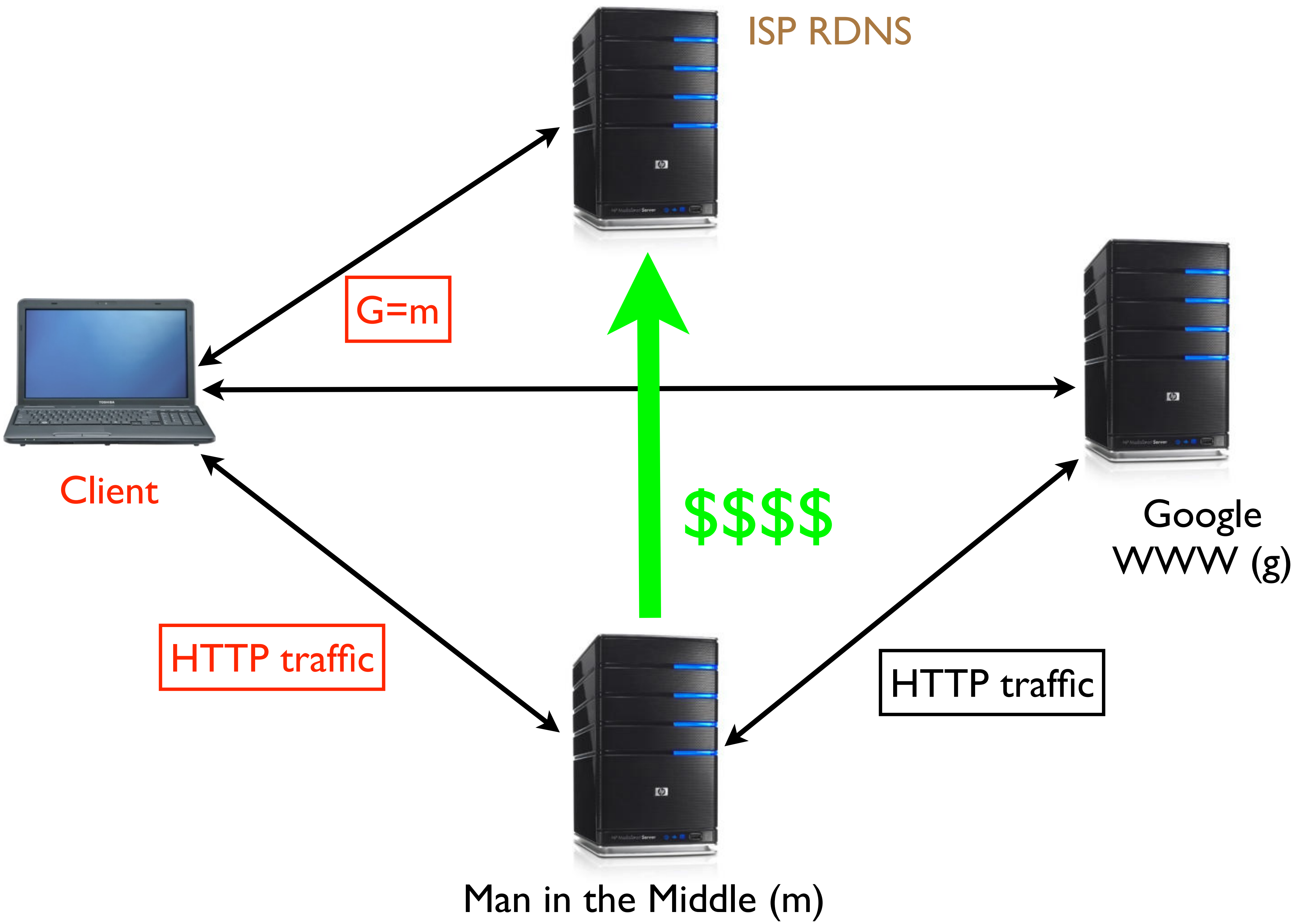
Search Engine Hijacking



Search Engine Hijacking



Search Engine Hijacking



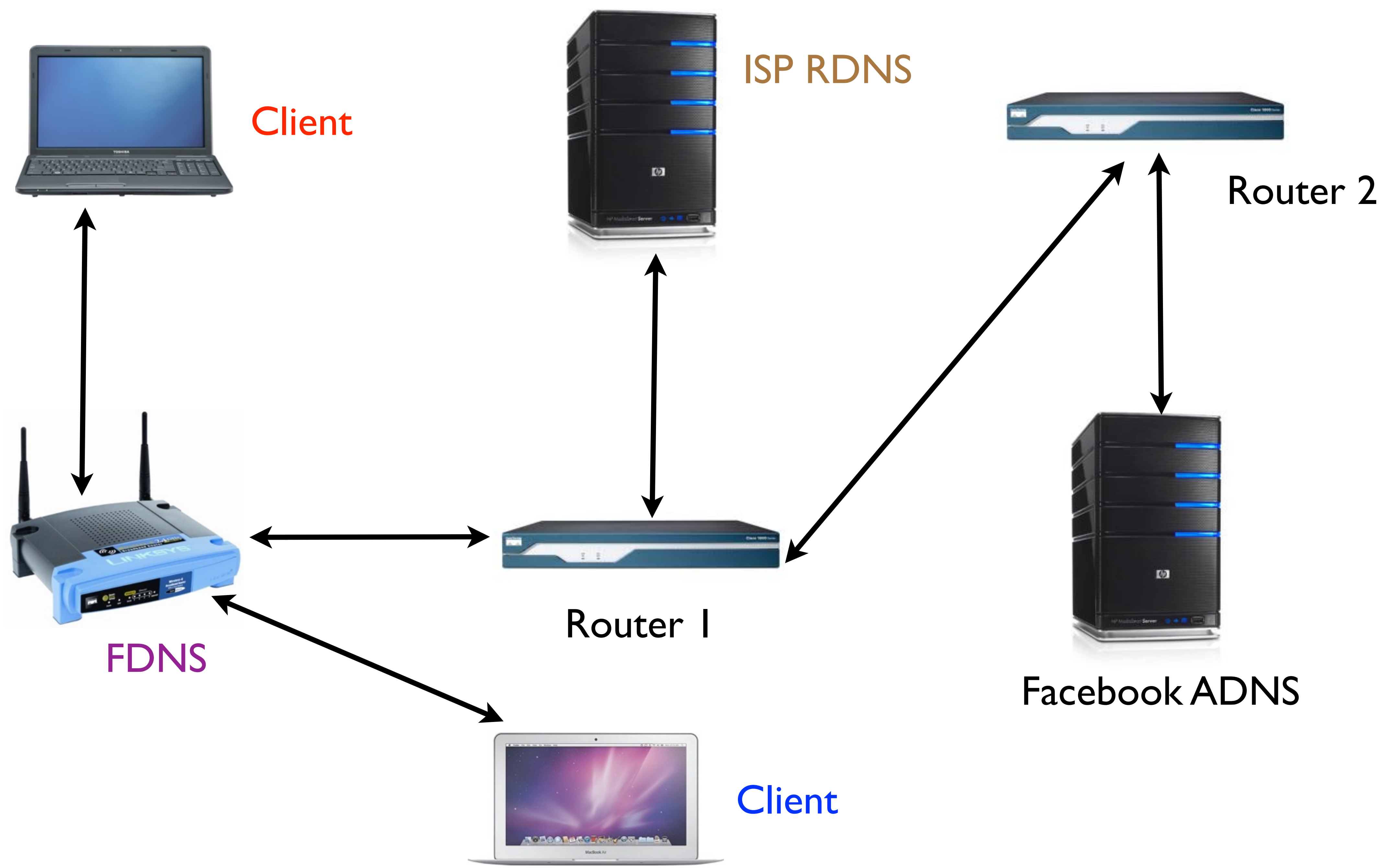
Search Engine Hijacking

Search Engine Hijacking

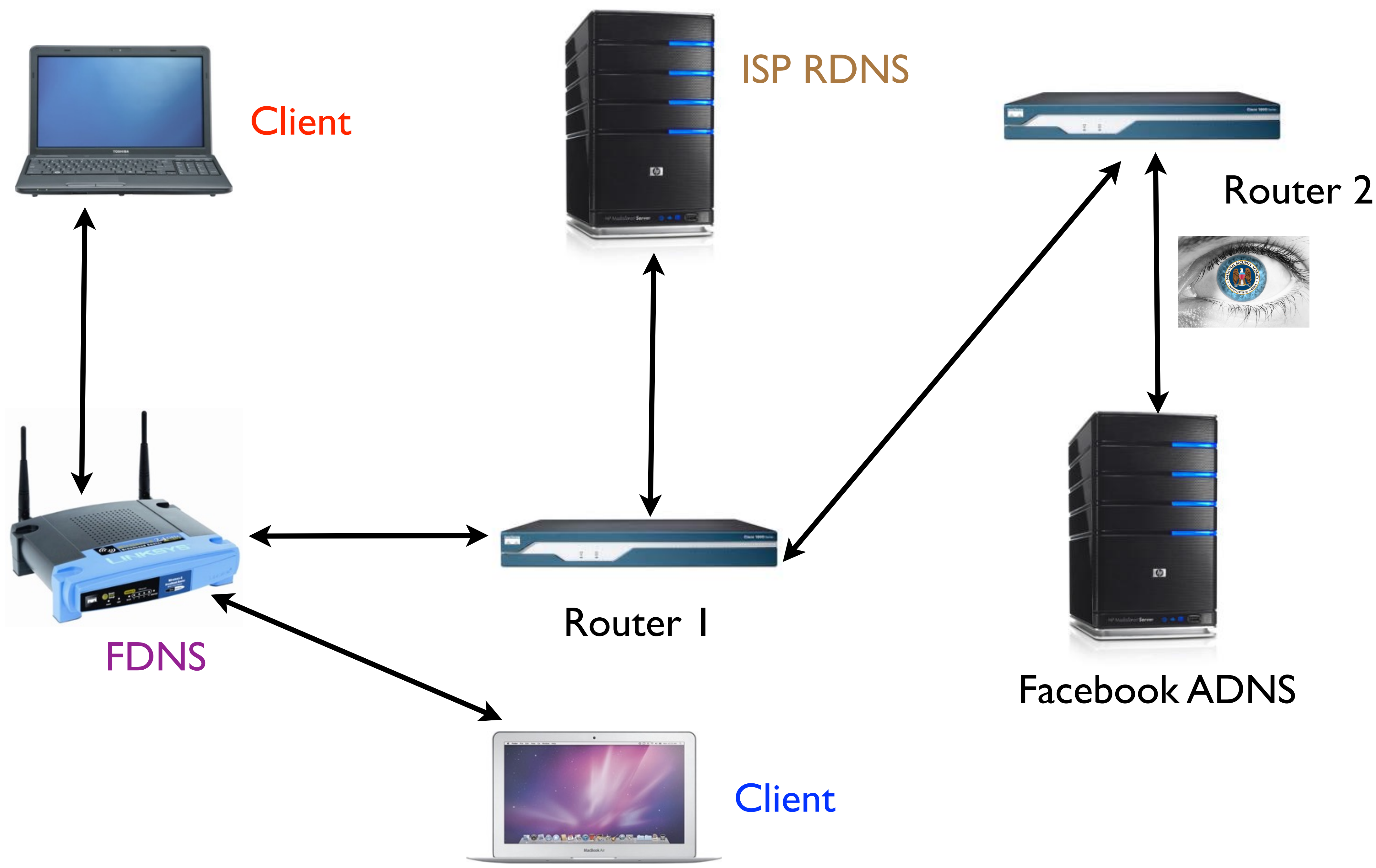
- Paxfire boxes once prevalent
  - no longer widespread
  - we detect small bits in 18 regional ISPs

- In general, our measurements do not show much of this form of attack

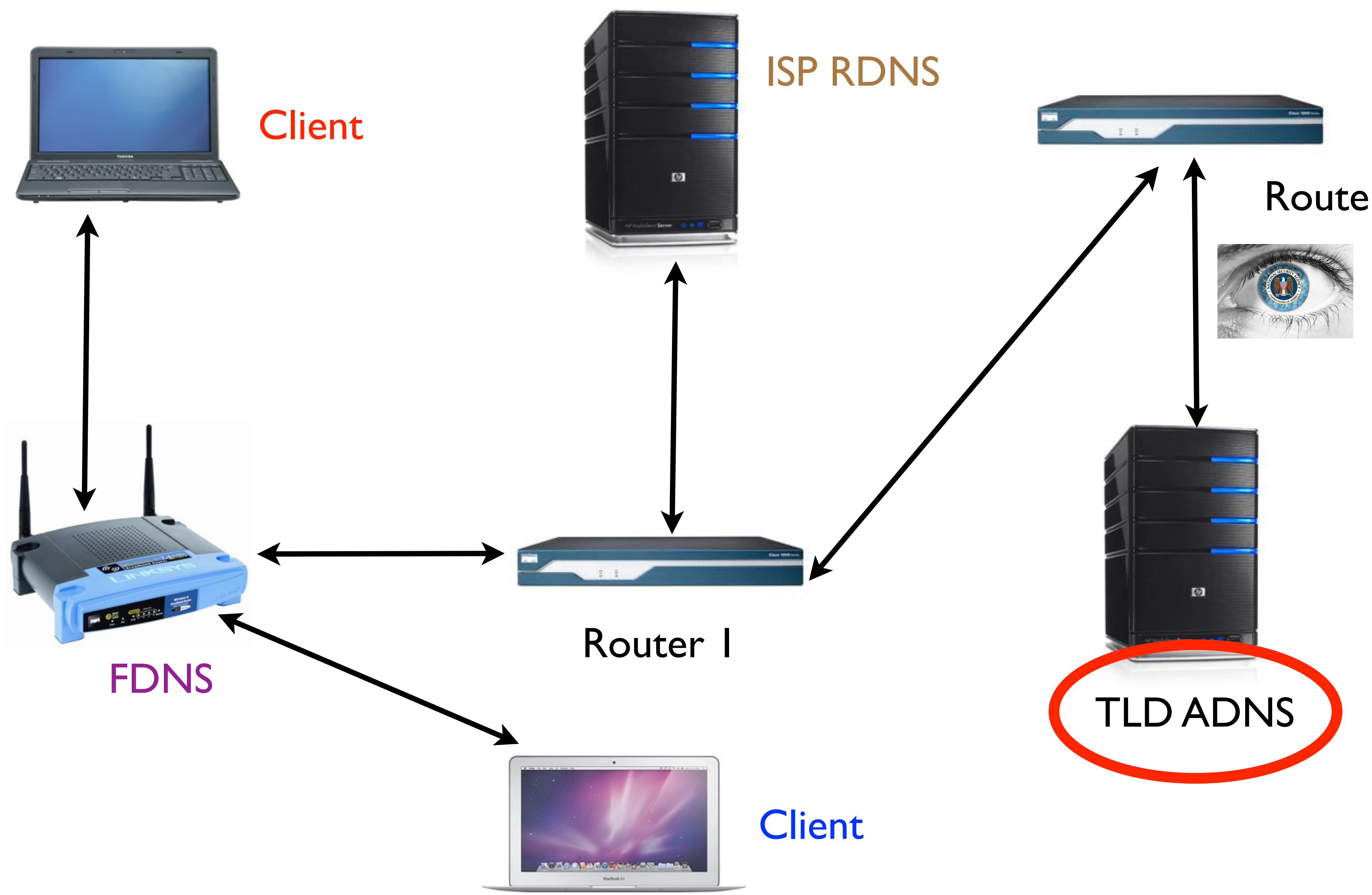
What Can Go Wrong? (3)



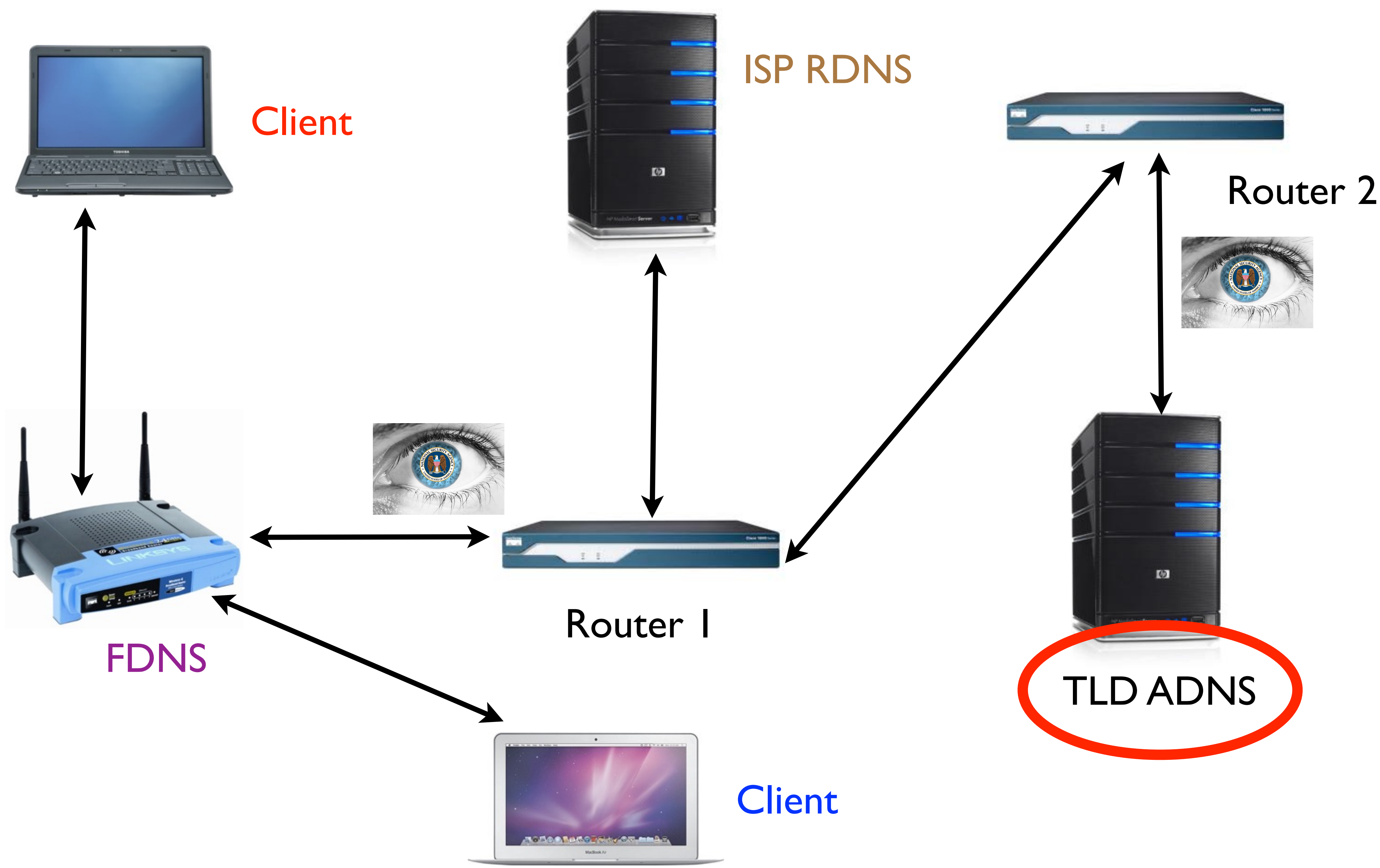
What Can Go Wrong? (3)



What Can Go Wrong? (3)



What Can Go Wrong? (3)



What Can Go Wrong? (3)

An Observation ...

- Thus far the attacks involve legitimate components being fraudulent or simple passive observation

- But, DNS is a simple, clear text protocol ...
- ... as are UDP and IP underneath

- So, crafting technically acceptable portions of the transaction is possible ...
- ... but, how difficult is it?

Key Transaction Elements

Key Transaction Elements

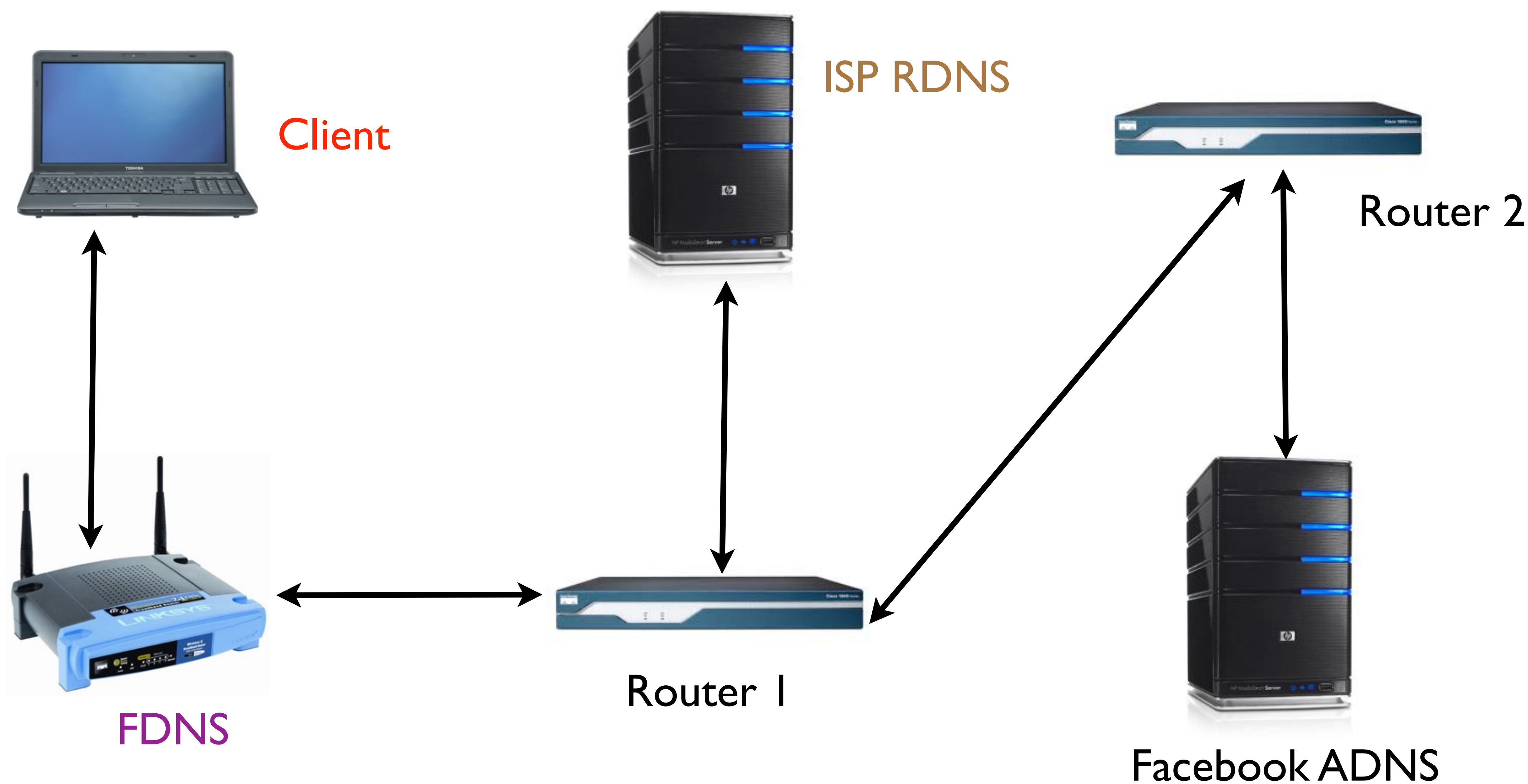
- IP: local IP address, remote IP address
- UDP: local port, remote port
- DNS: transaction ID, query string

Key Transaction Elements

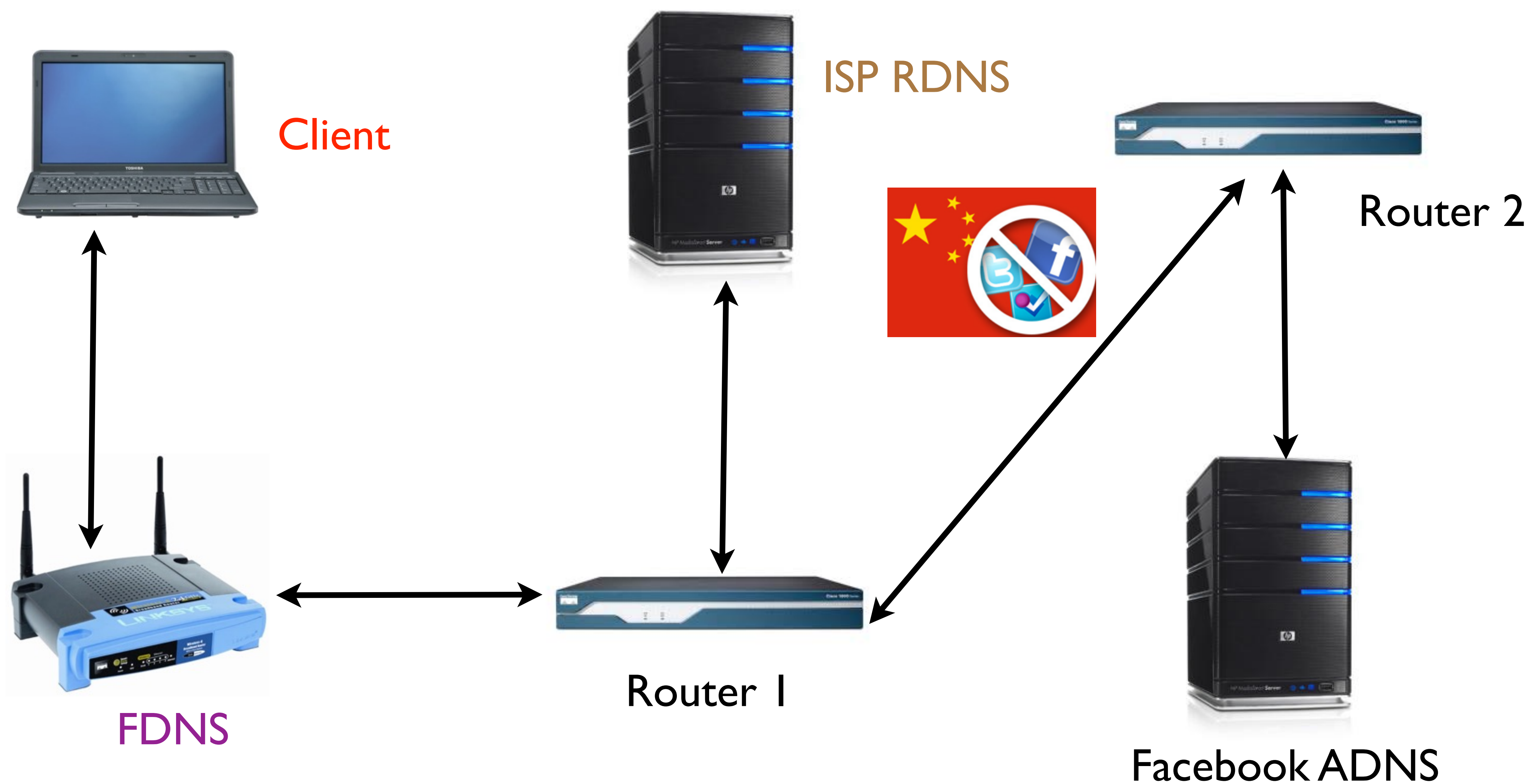
- IP: local IP address, remote IP address
- UDP: local port, remote port
- DNS: transaction ID, query string

- If we know all of these we can create components of the transactions that are ...
  .... *technically acceptable*
  .... *but, contain fraudulent content*

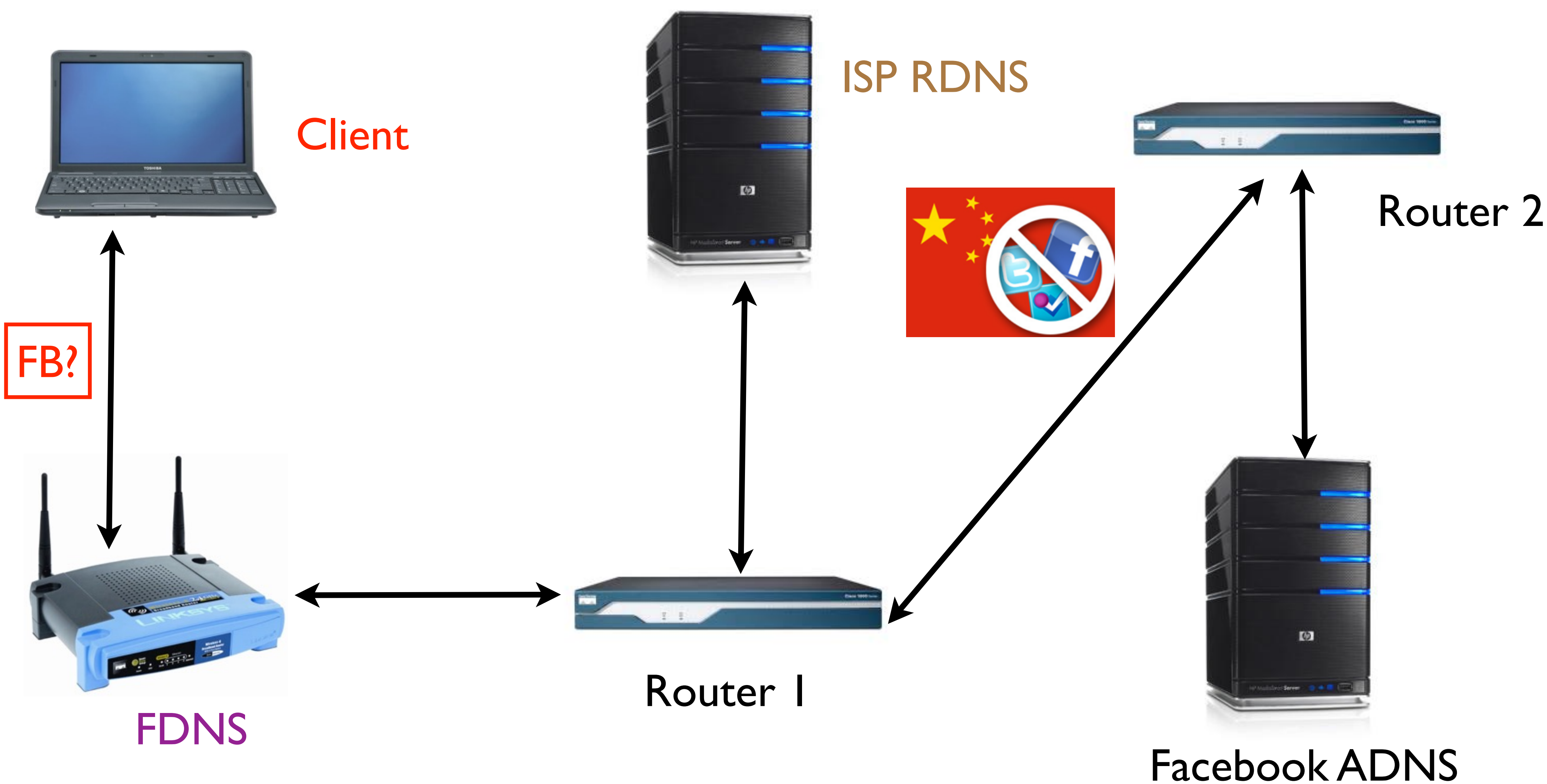
What Can Go Wrong? (3.5)



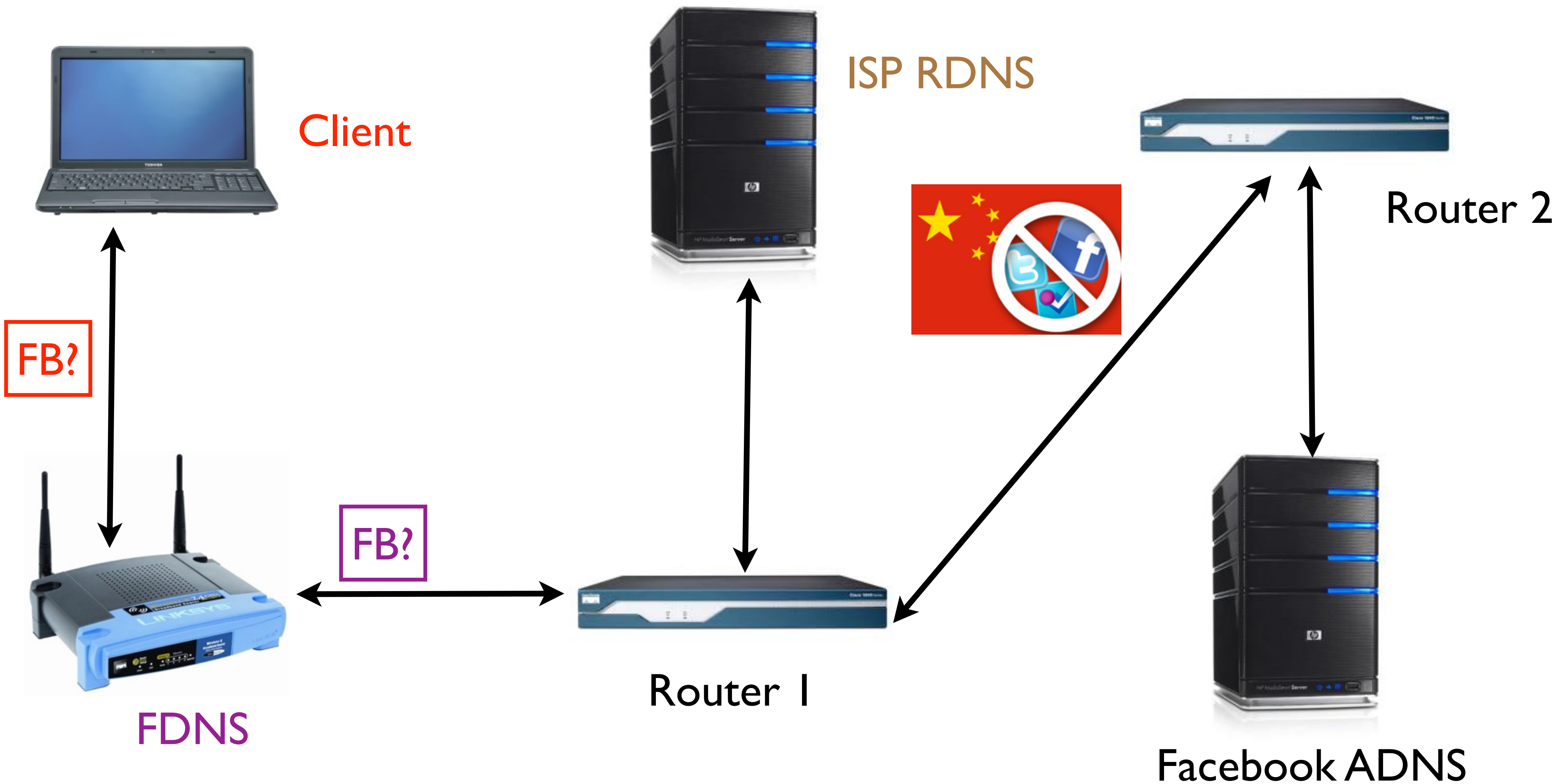
What Can Go Wrong? (3.5)



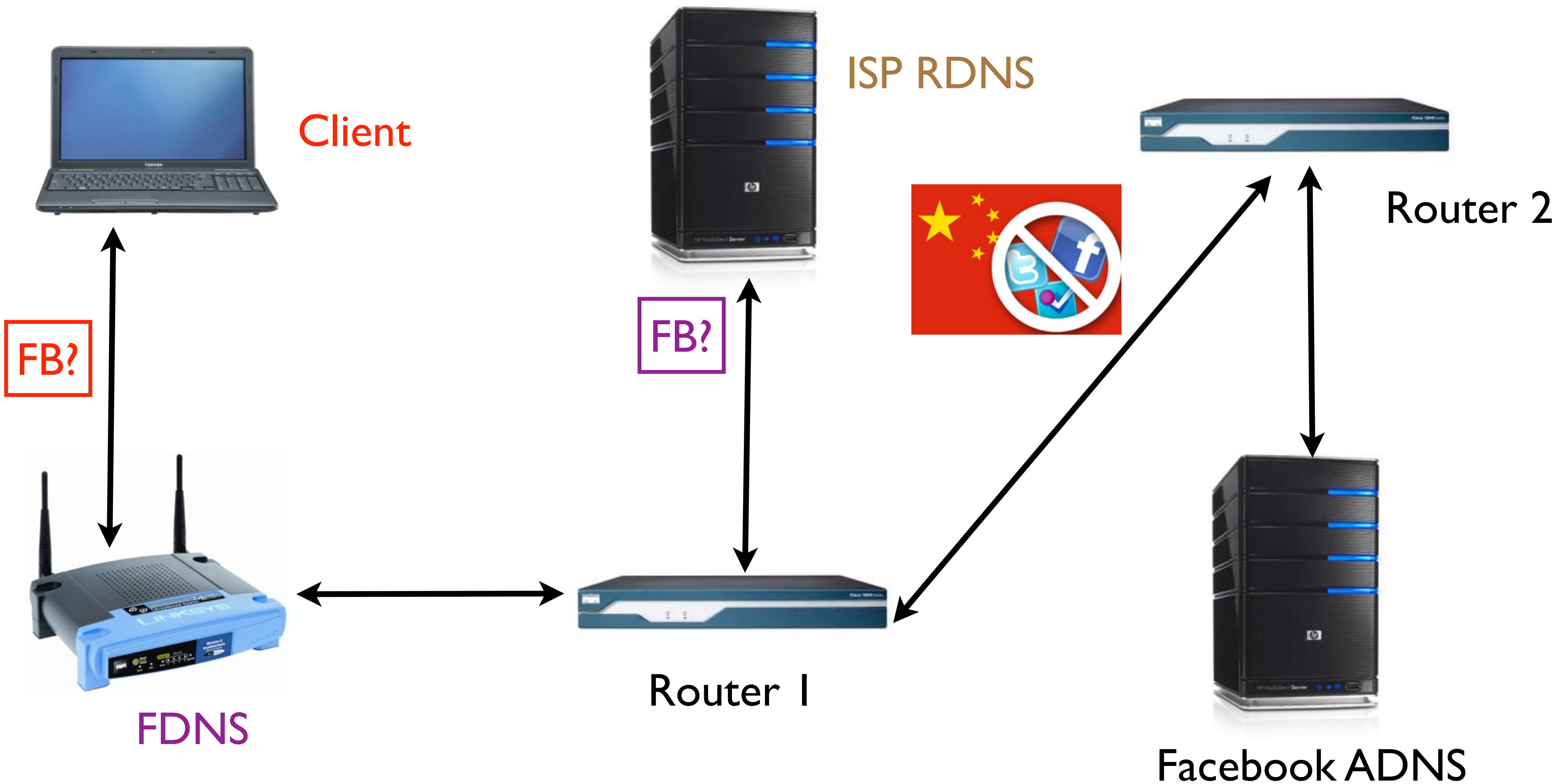
What Can Go Wrong? (3.5)



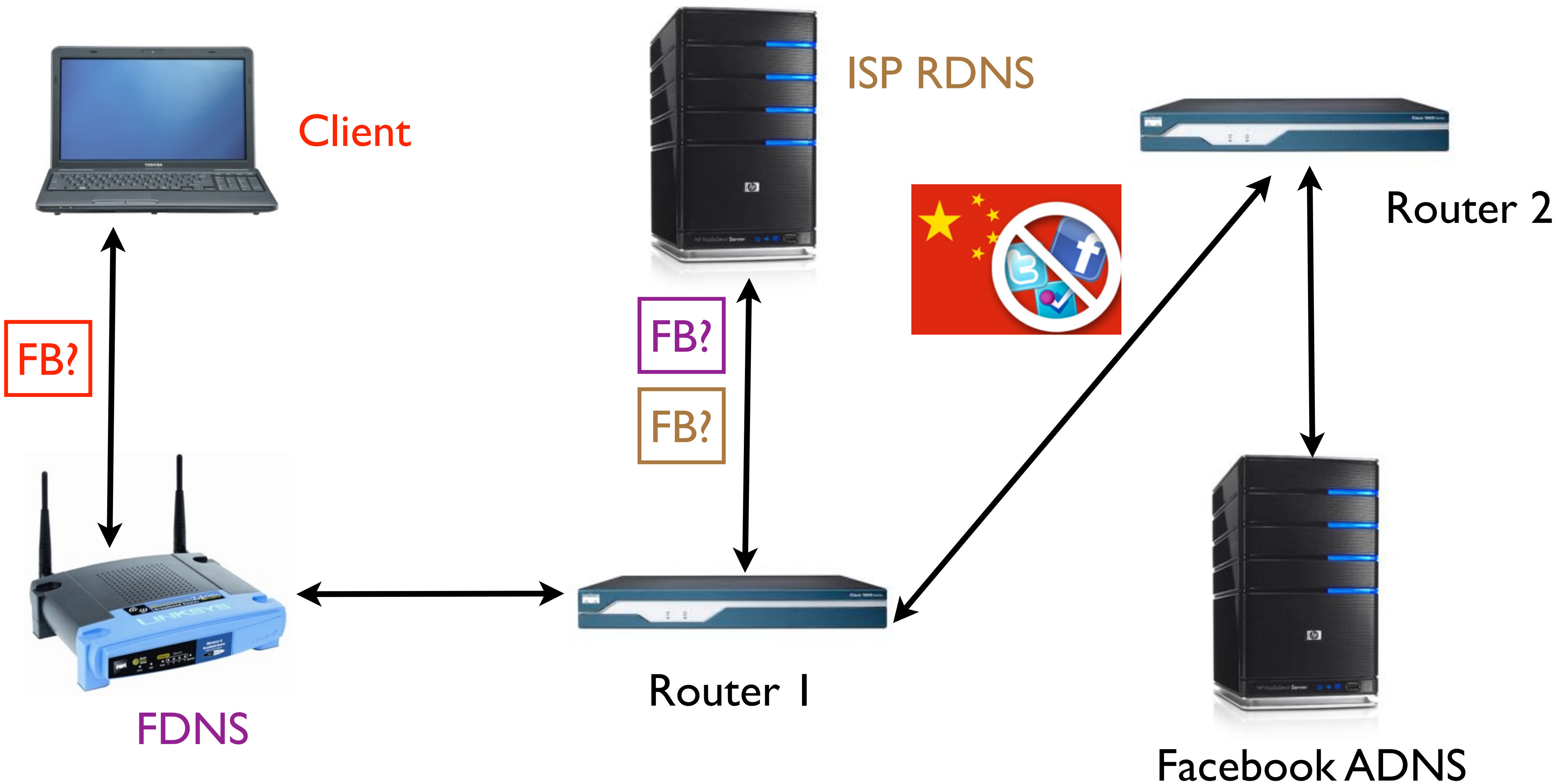
What Can Go Wrong? (3.5)



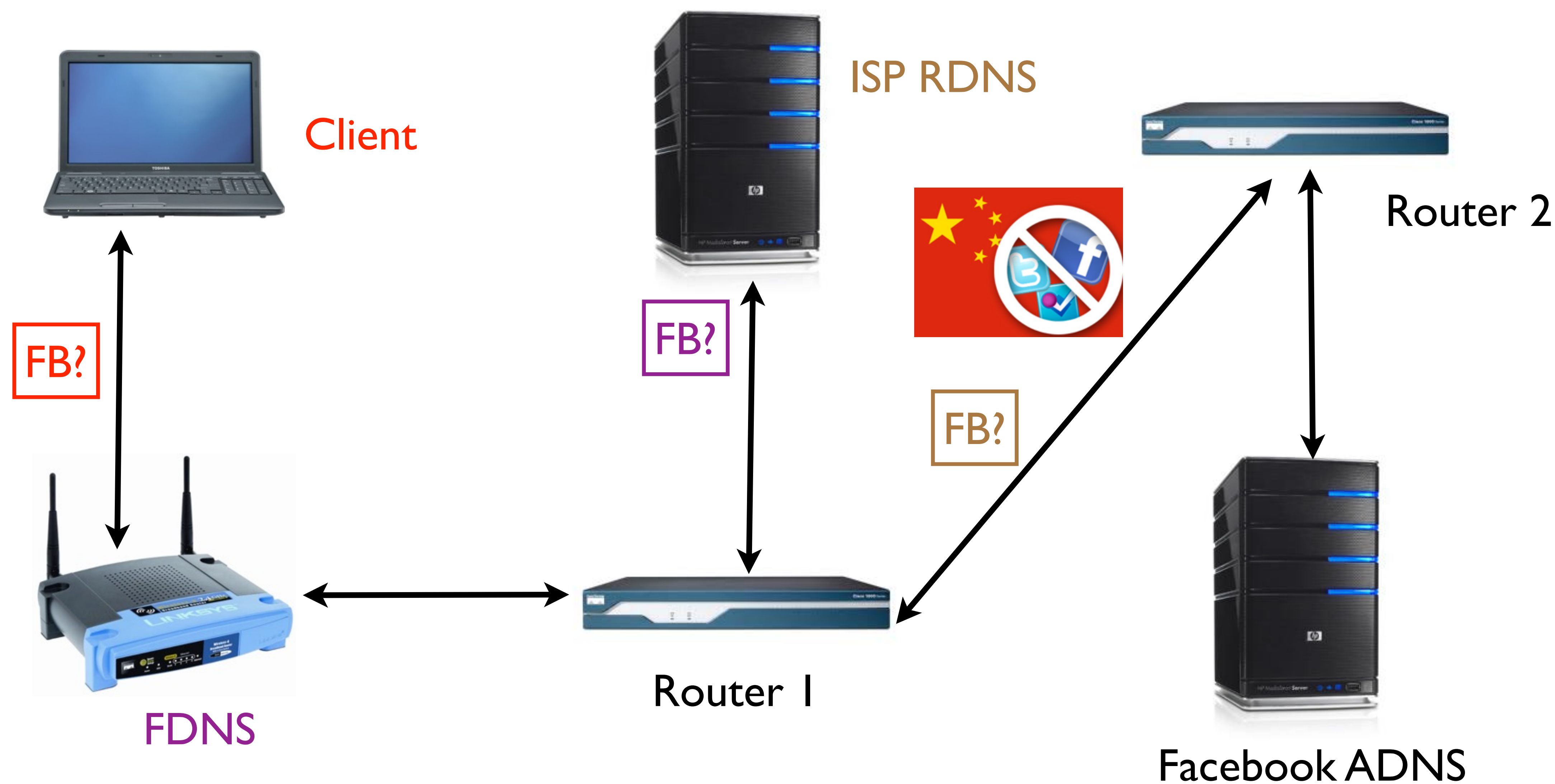
What Can Go Wrong? (3.5)



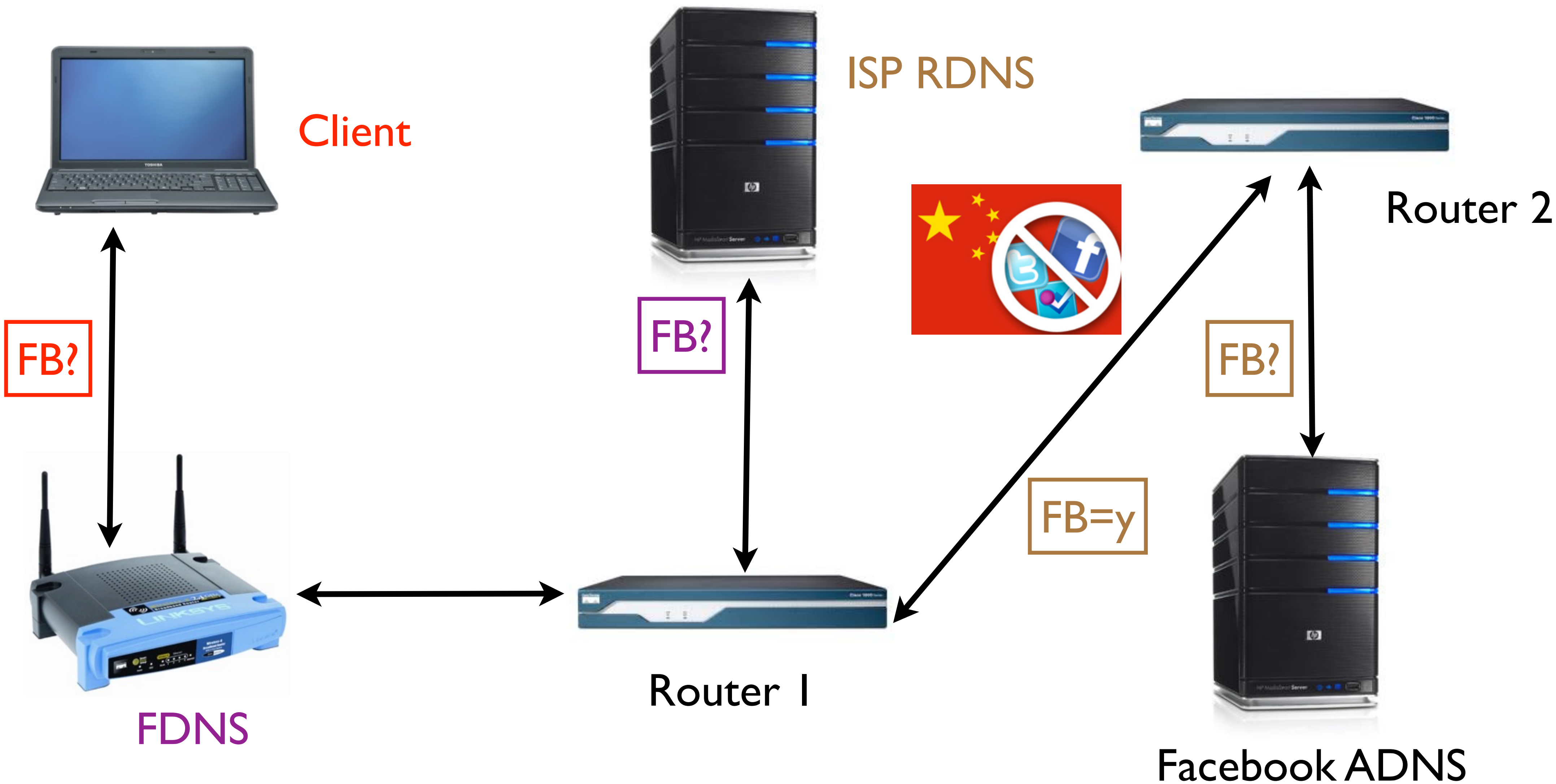
What Can Go Wrong? (3.5)



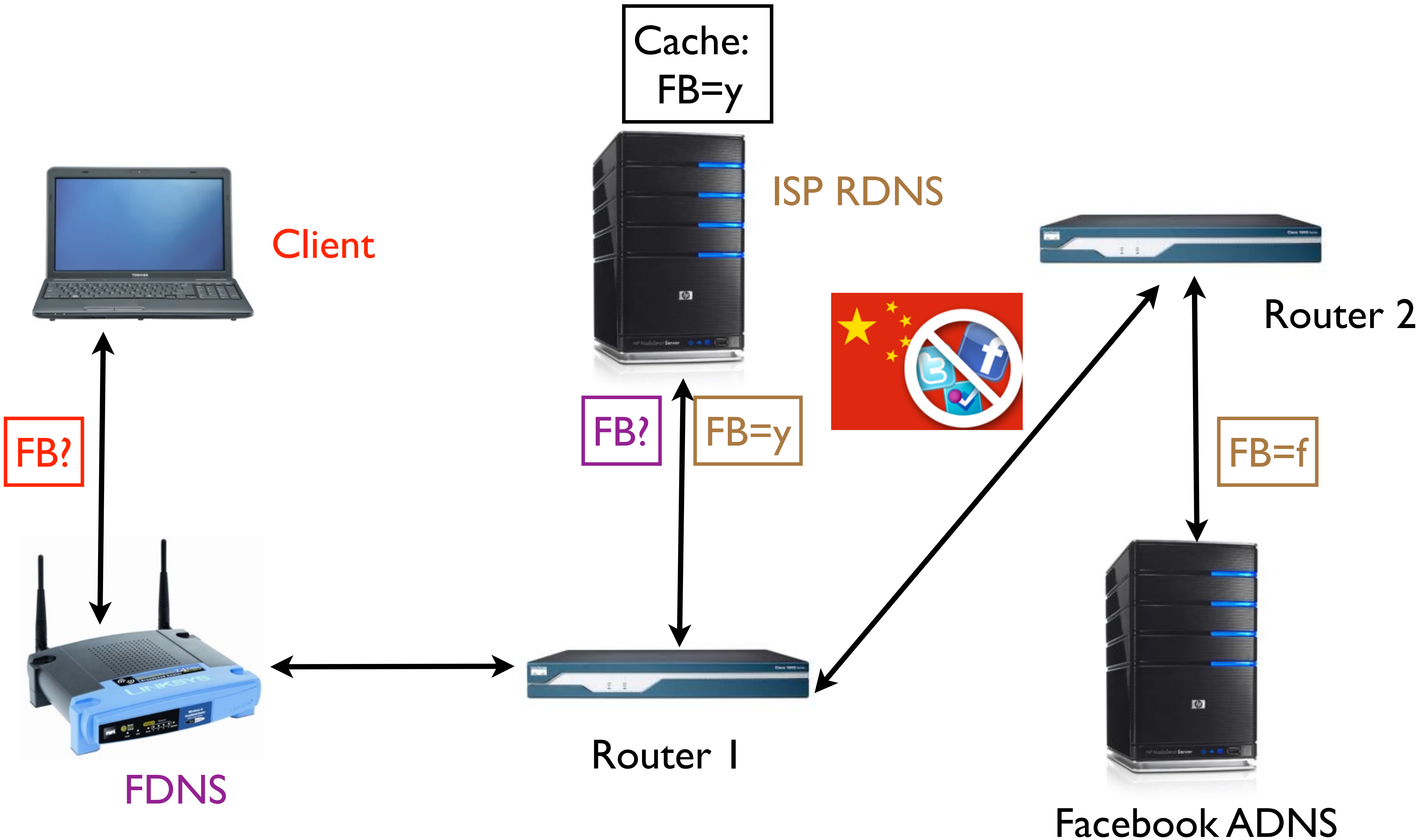
What Can Go Wrong? (3.5)



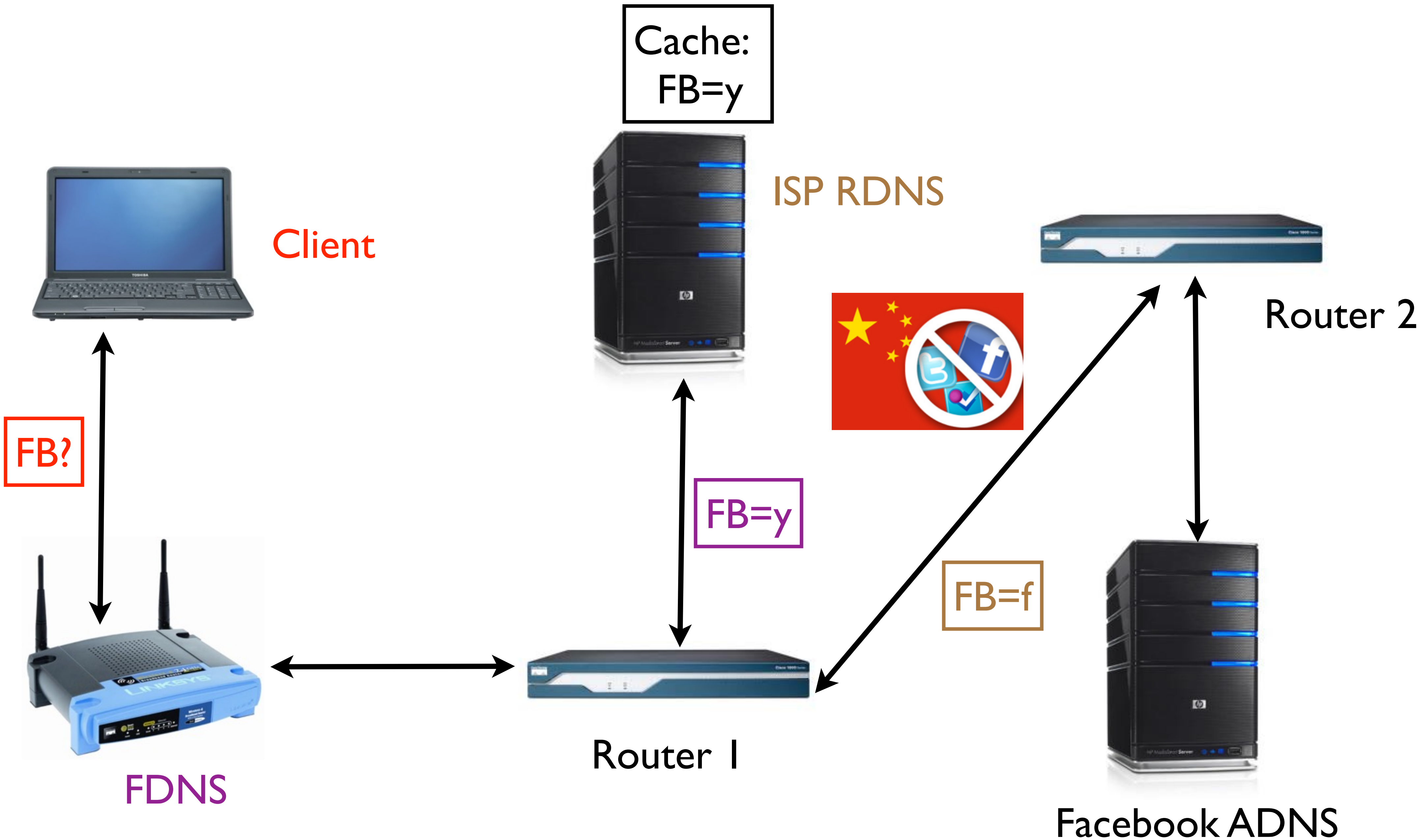
What Can Go Wrong? (3.5)



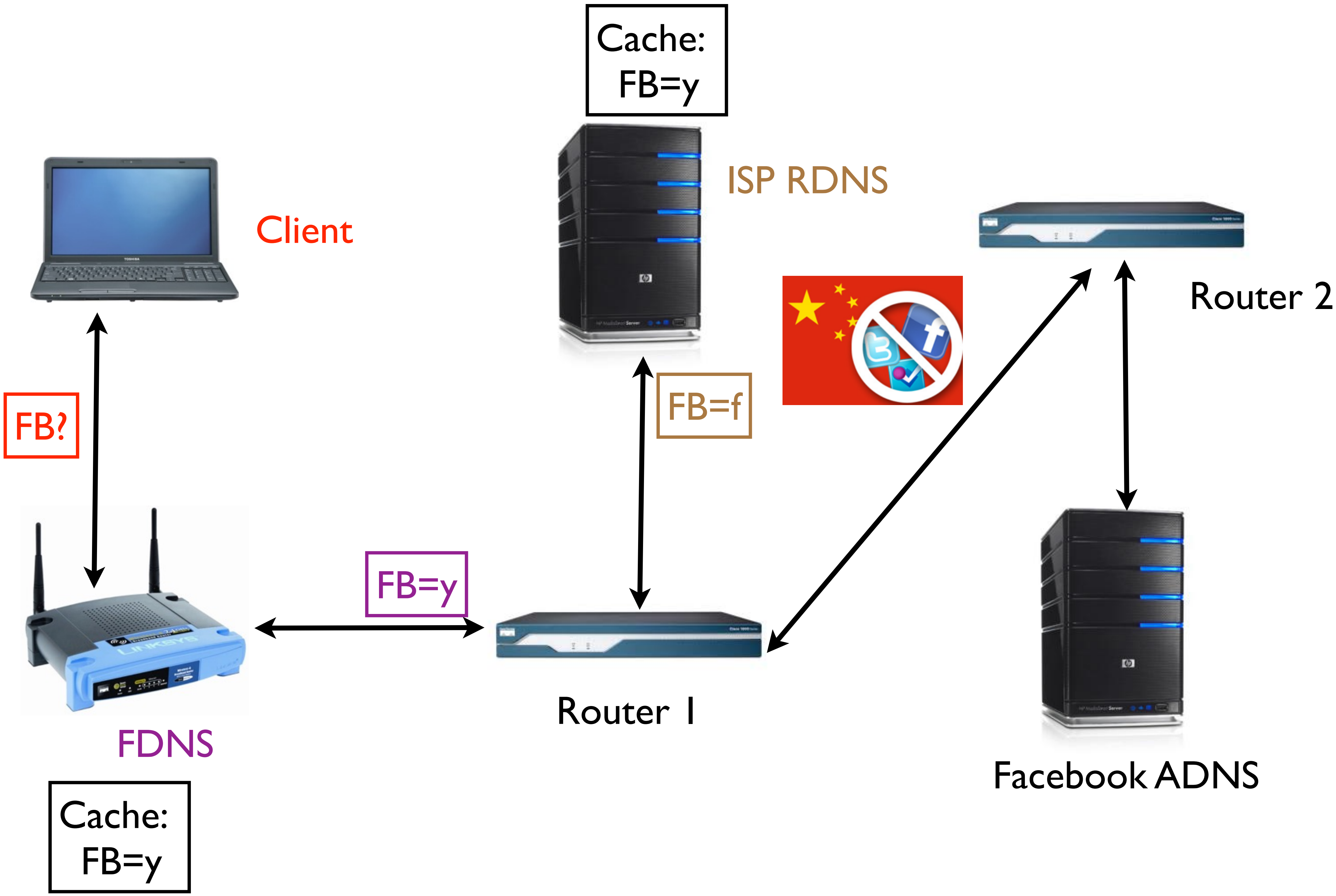
What Can Go Wrong? (3.5)



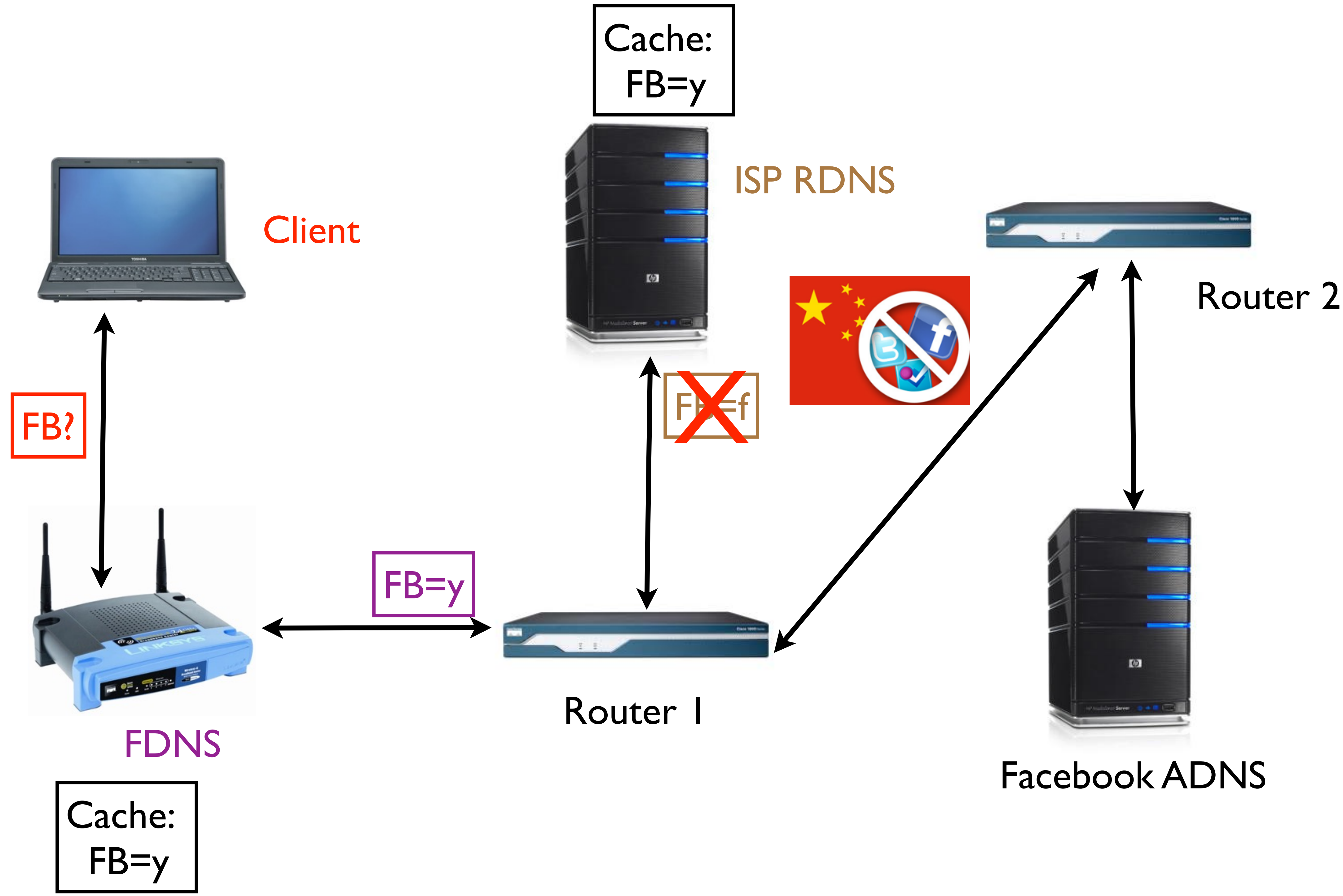
What Can Go Wrong? (3.5)



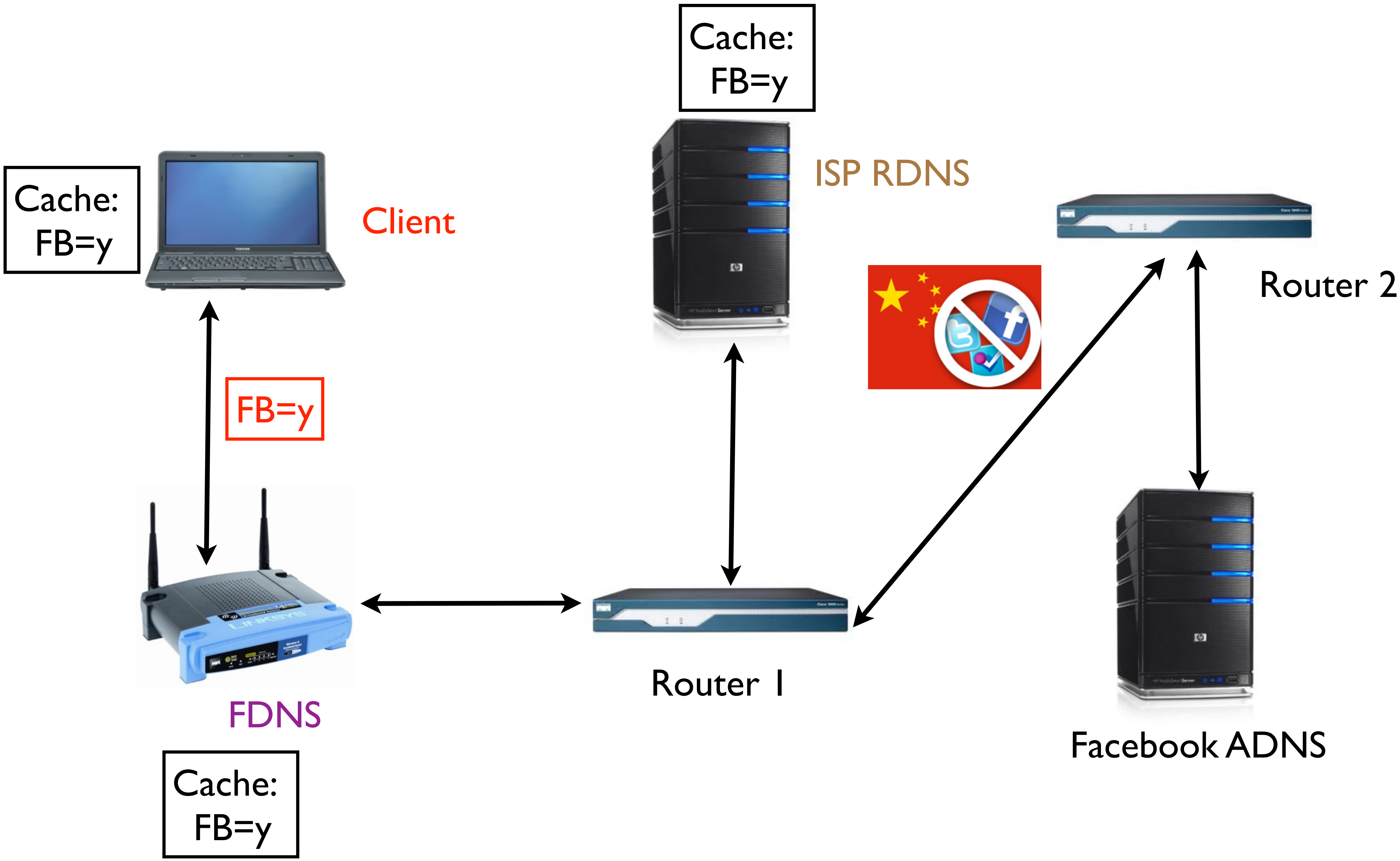
What Can Go Wrong? (3.5)



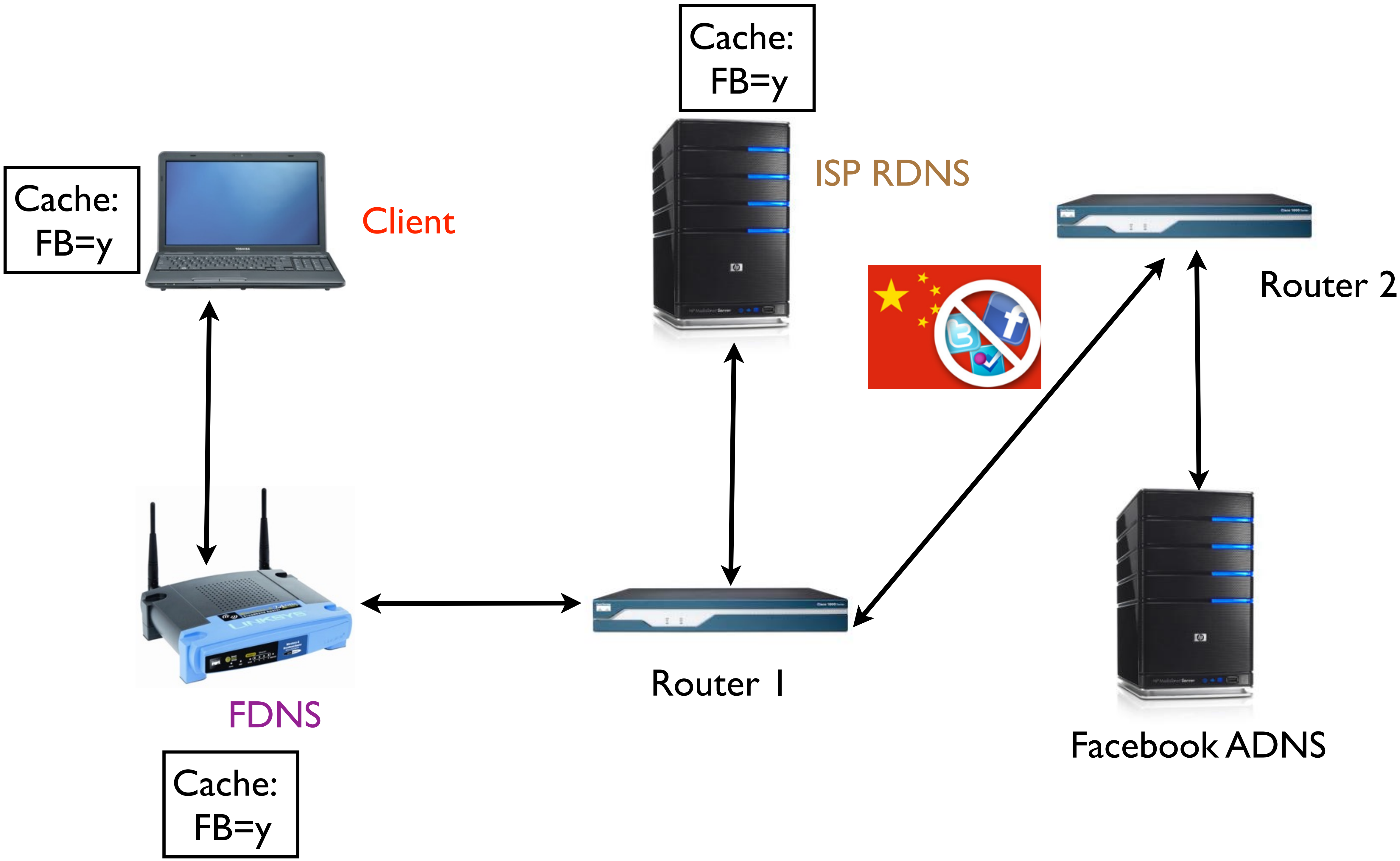
What Can Go Wrong? (3.5)



What Can Go Wrong? (3.5)



What Can Go Wrong? (3.5)



Key Transaction Elements

Key Transaction Elements

- IP: local IP address, remote IP address
- UDP: local port, remote port
- DNS: transaction ID, query string

- If we observe a request, we can create a fraudulent response

Key Transaction Elements

- IP: local IP address, remote IP address
- UDP: local port, remote port
- DNS: transaction ID, query string

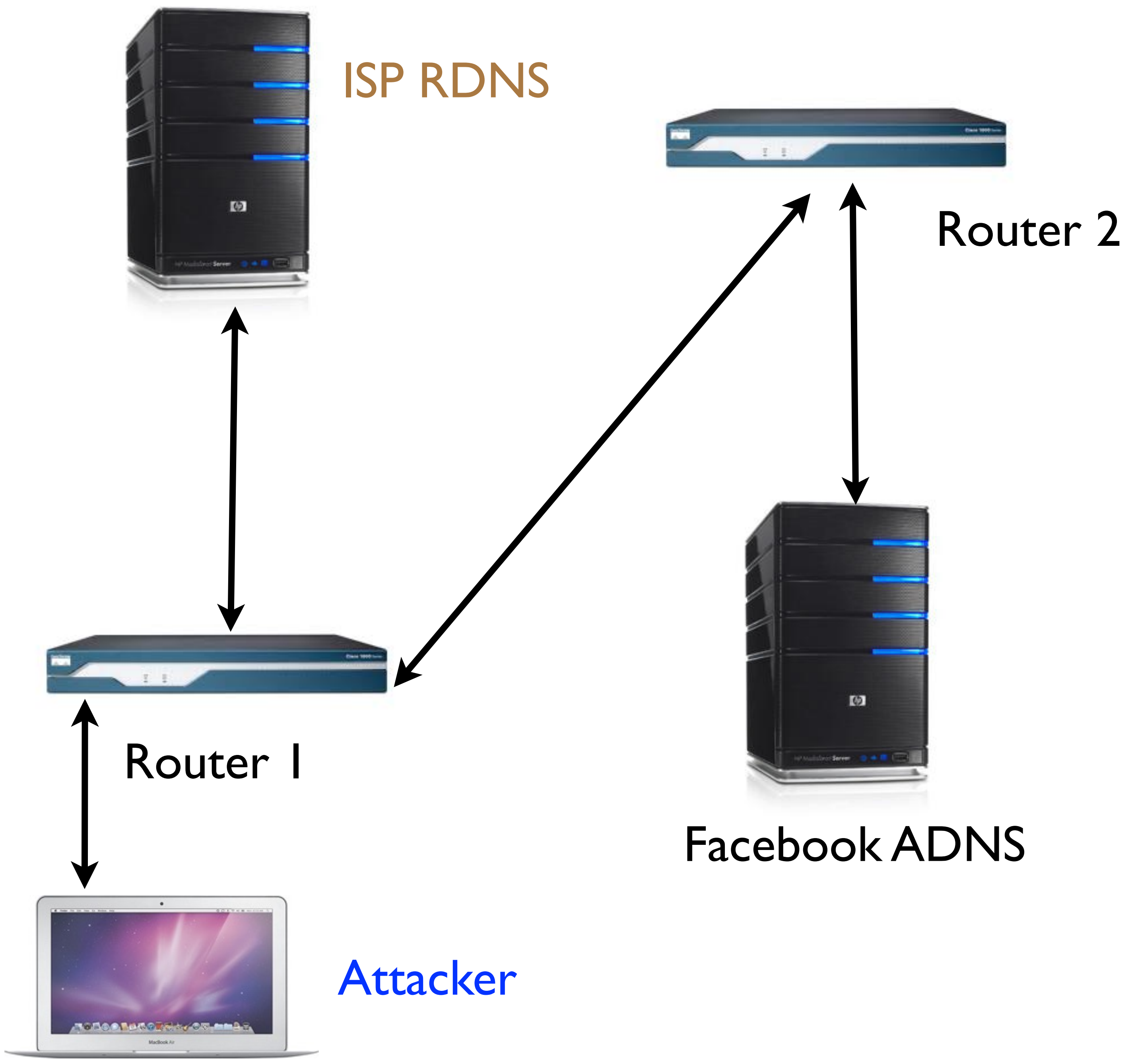
- If we observe a request, we can create a fraudulent response
- But, what if we don't observe the request?

Key Transaction Elements

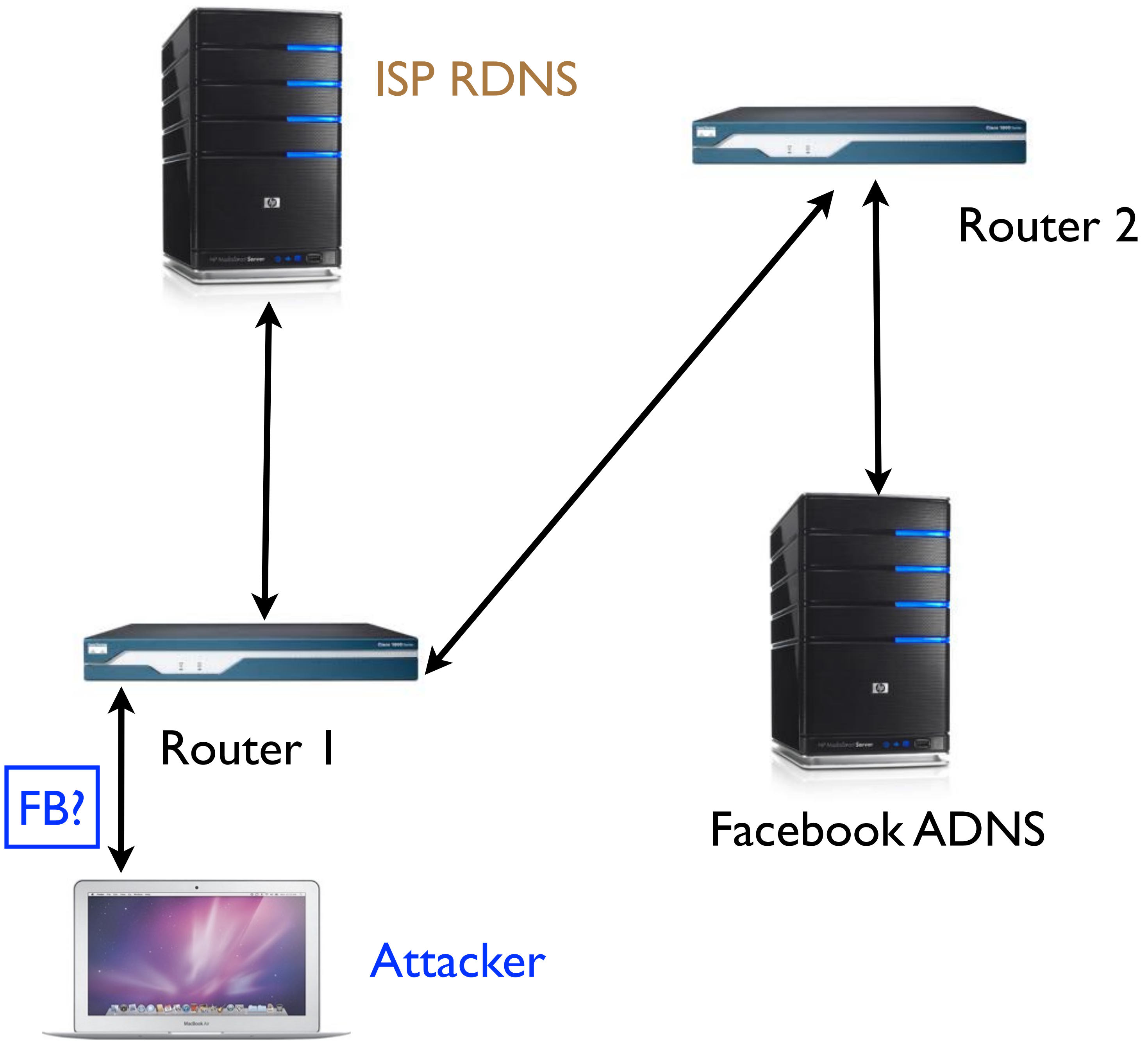
- IP: local IP address, remote IP address
- UDP: local port, remote port
- DNS: transaction ID, query string

- If we observe a request, we can create a fraudulent response
- But, what if we don't observe the request?
  - Forging a response is not as hard as one might imagine!

What Can Go Wrong? (4)

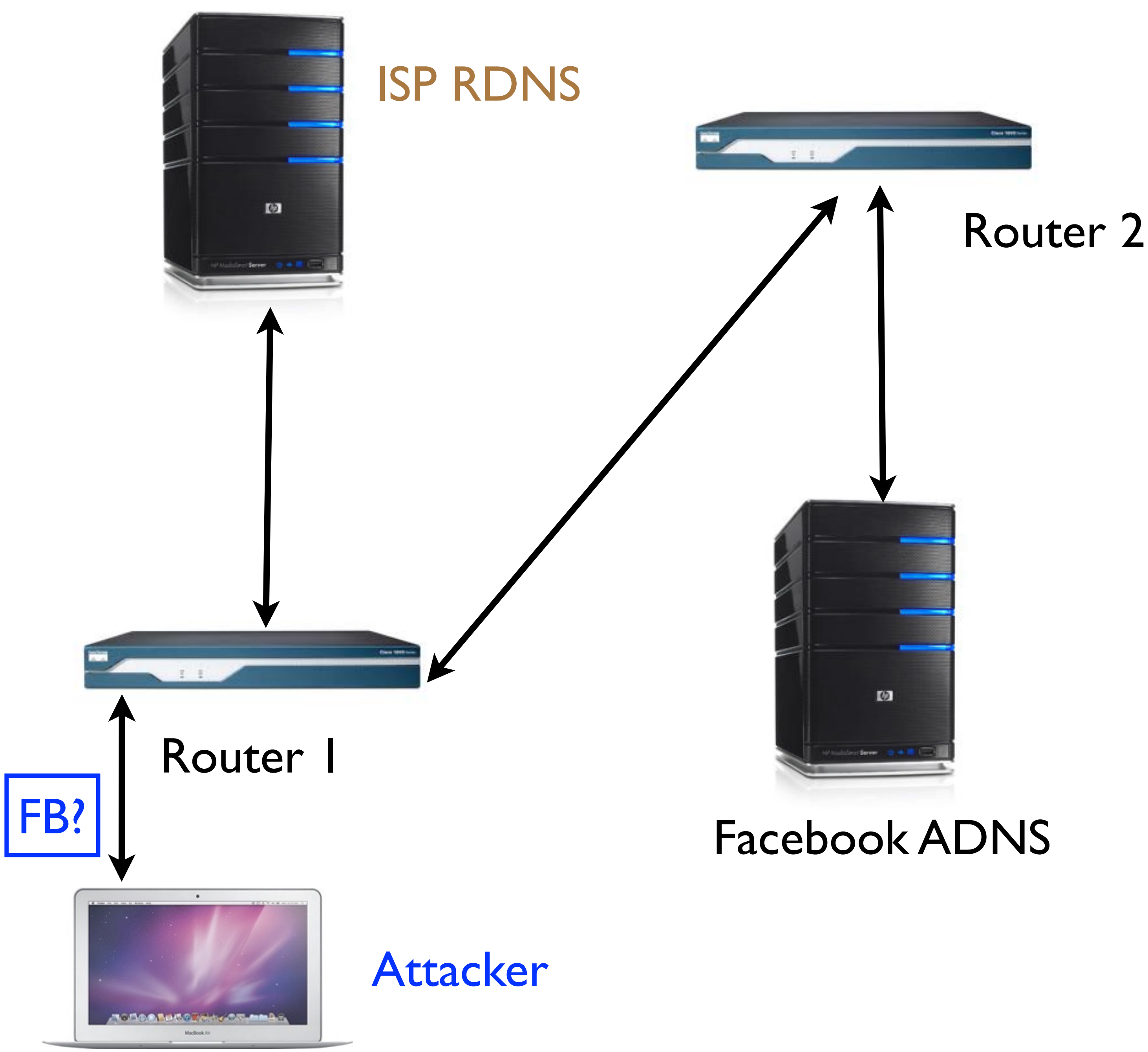


What Can Go Wrong? (4)



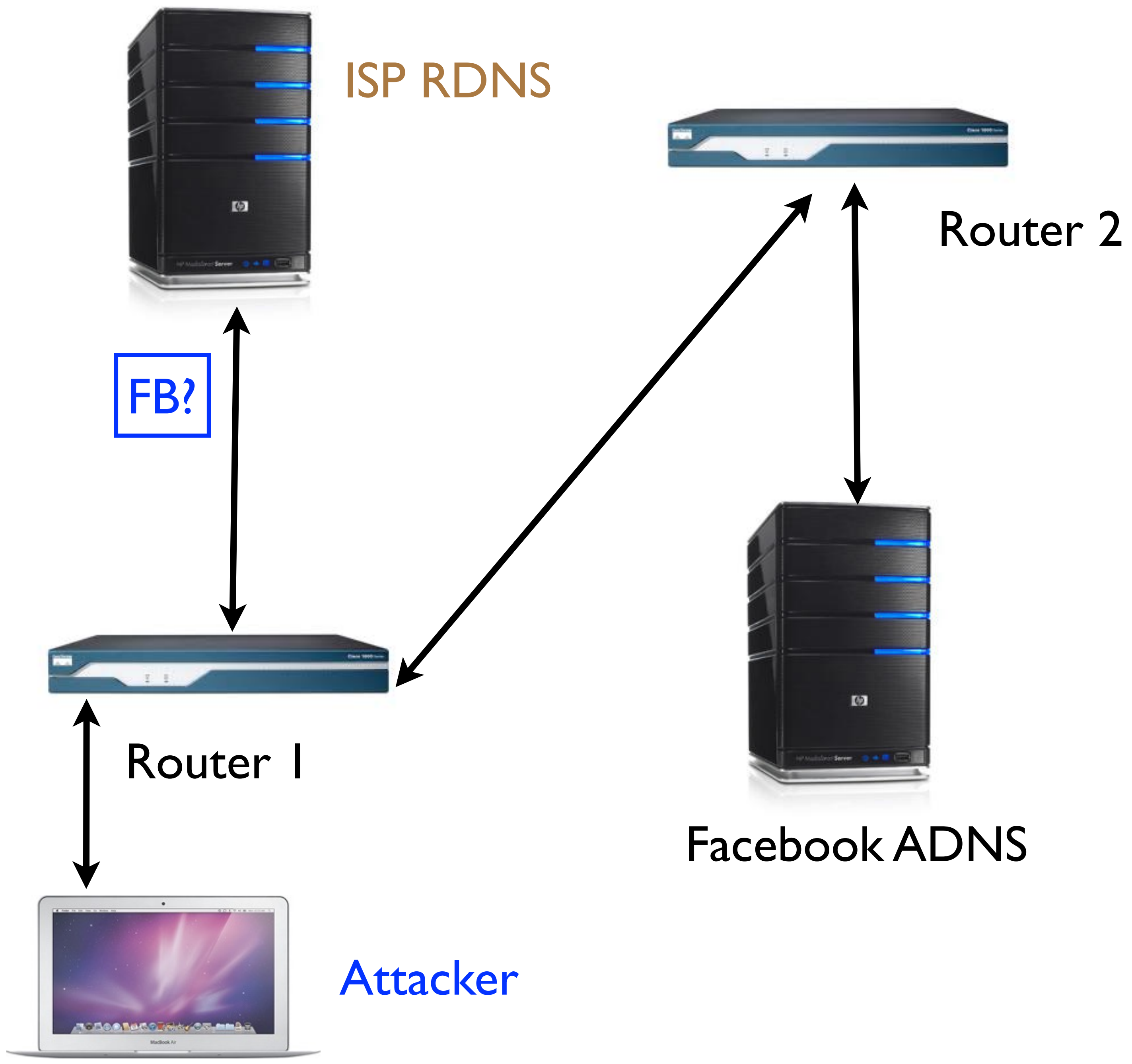
What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com



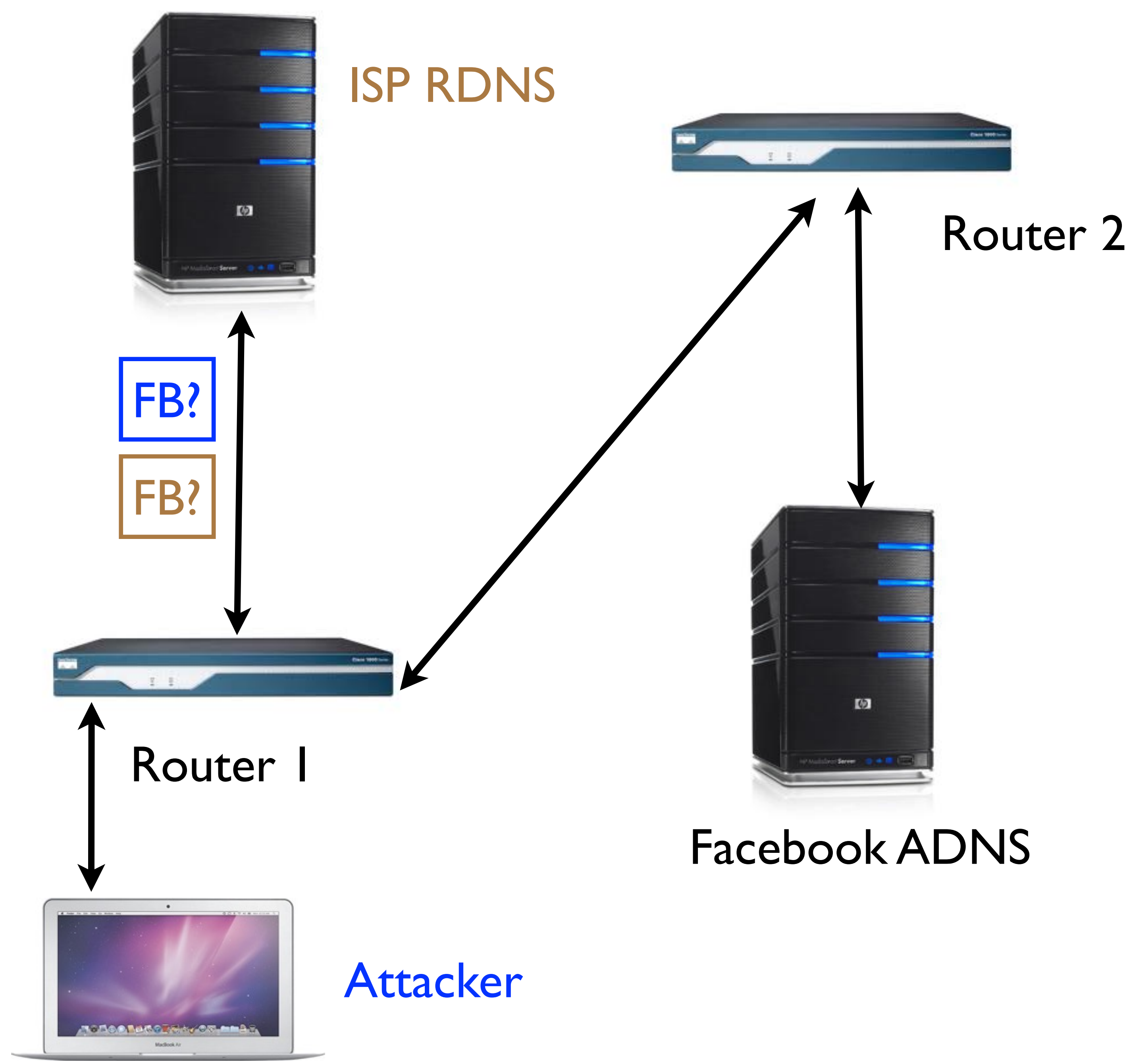
What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com



What Can Go Wrong? (4)

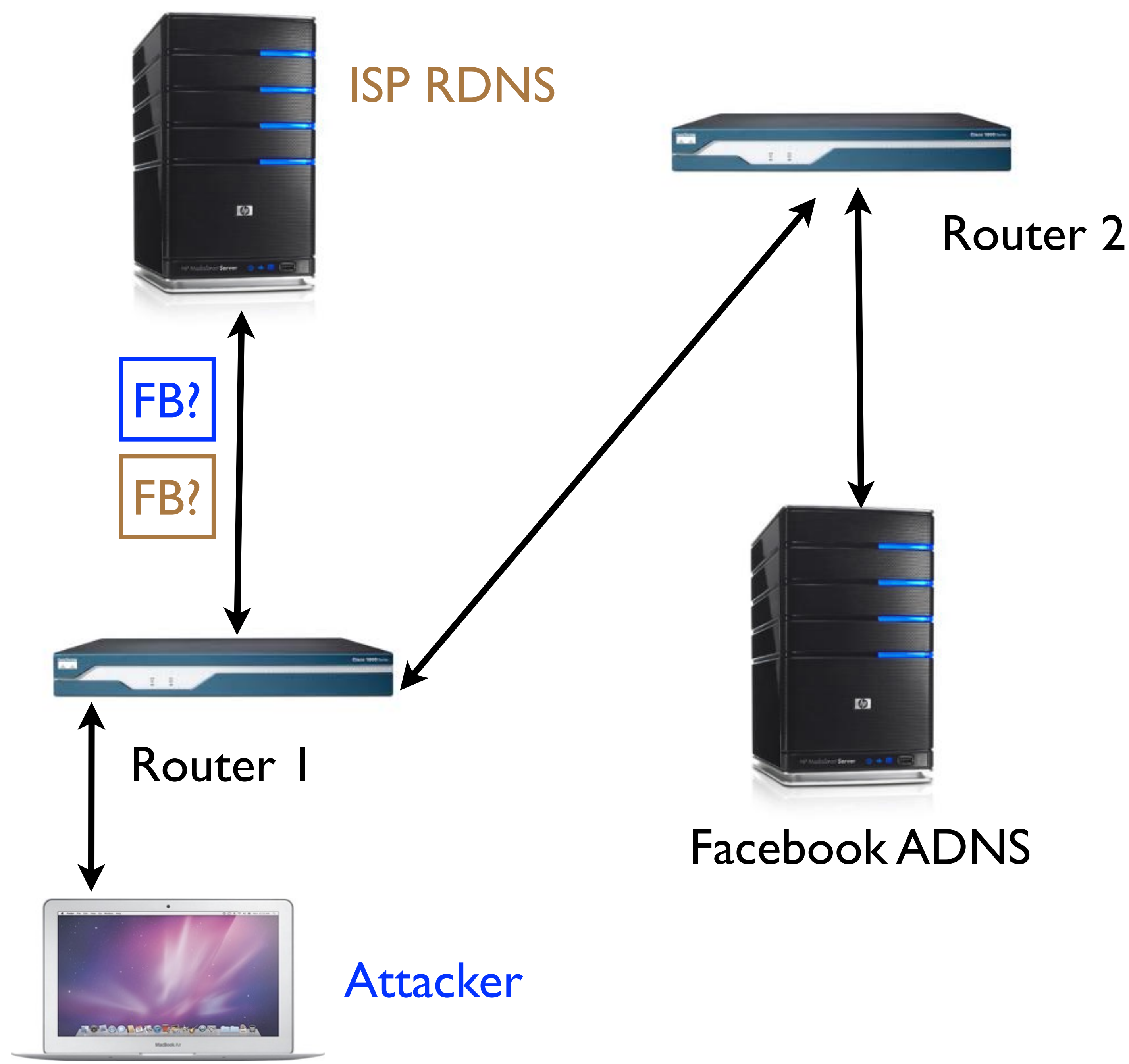
local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com



What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com

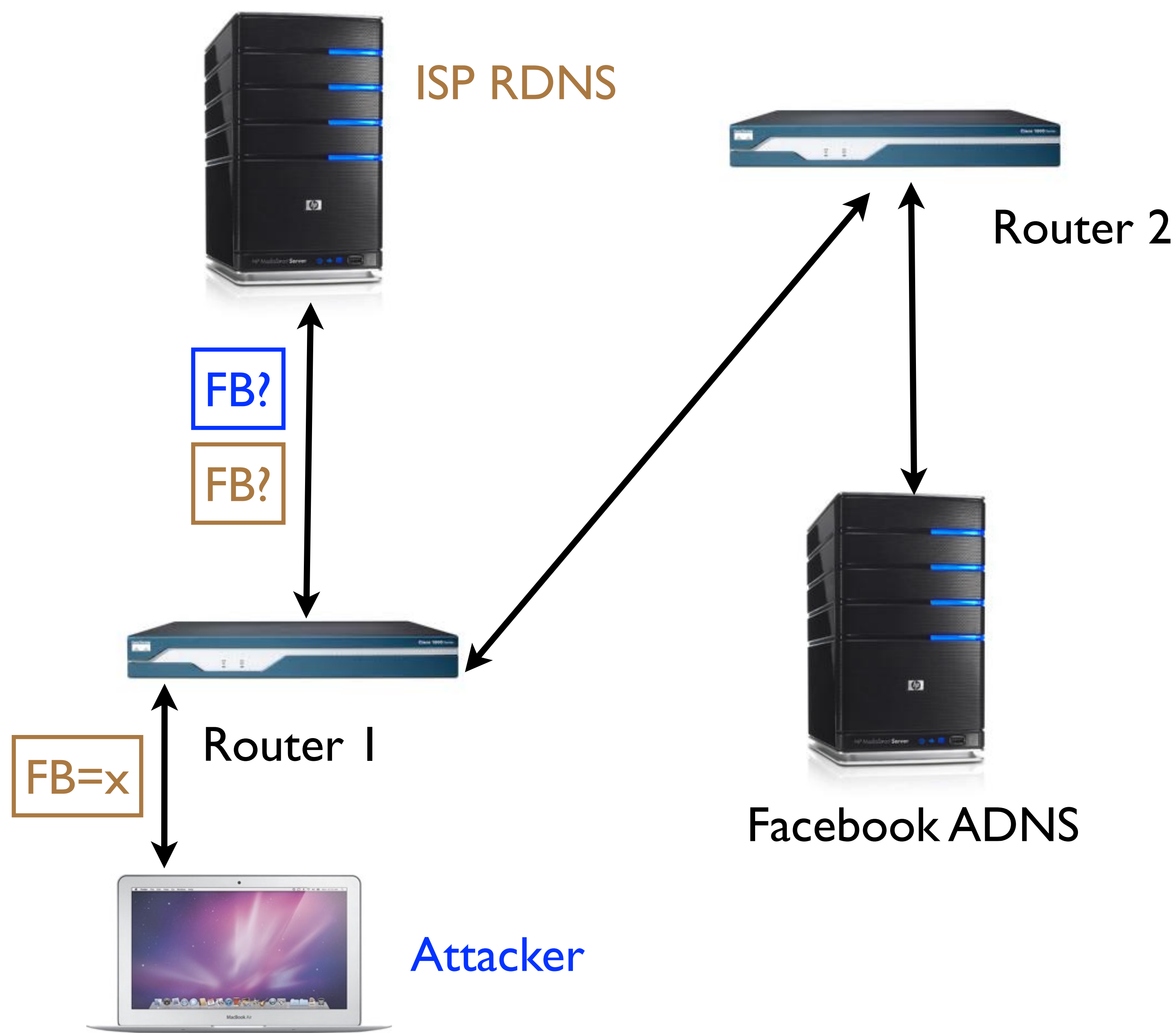
local IP: RDNS
remote IP: FB ADNS
local port: ???
remote port: 53
txID: ???
query: www.facebook.com



What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com

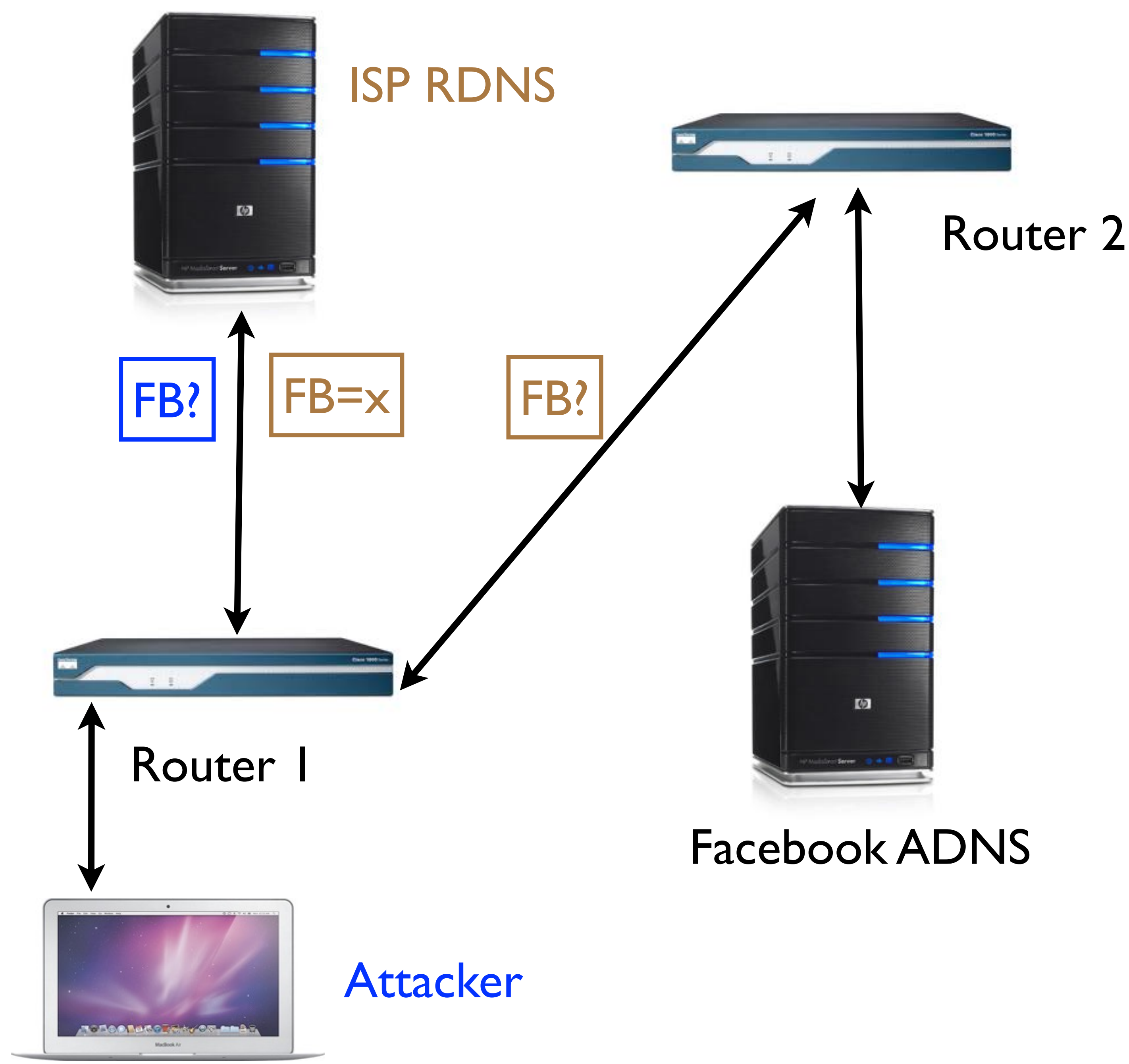
local IP: RDNS
remote IP: FB ADNS
local port: ???
remote port: 53
txID: ???
query: www.facebook.com



What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com

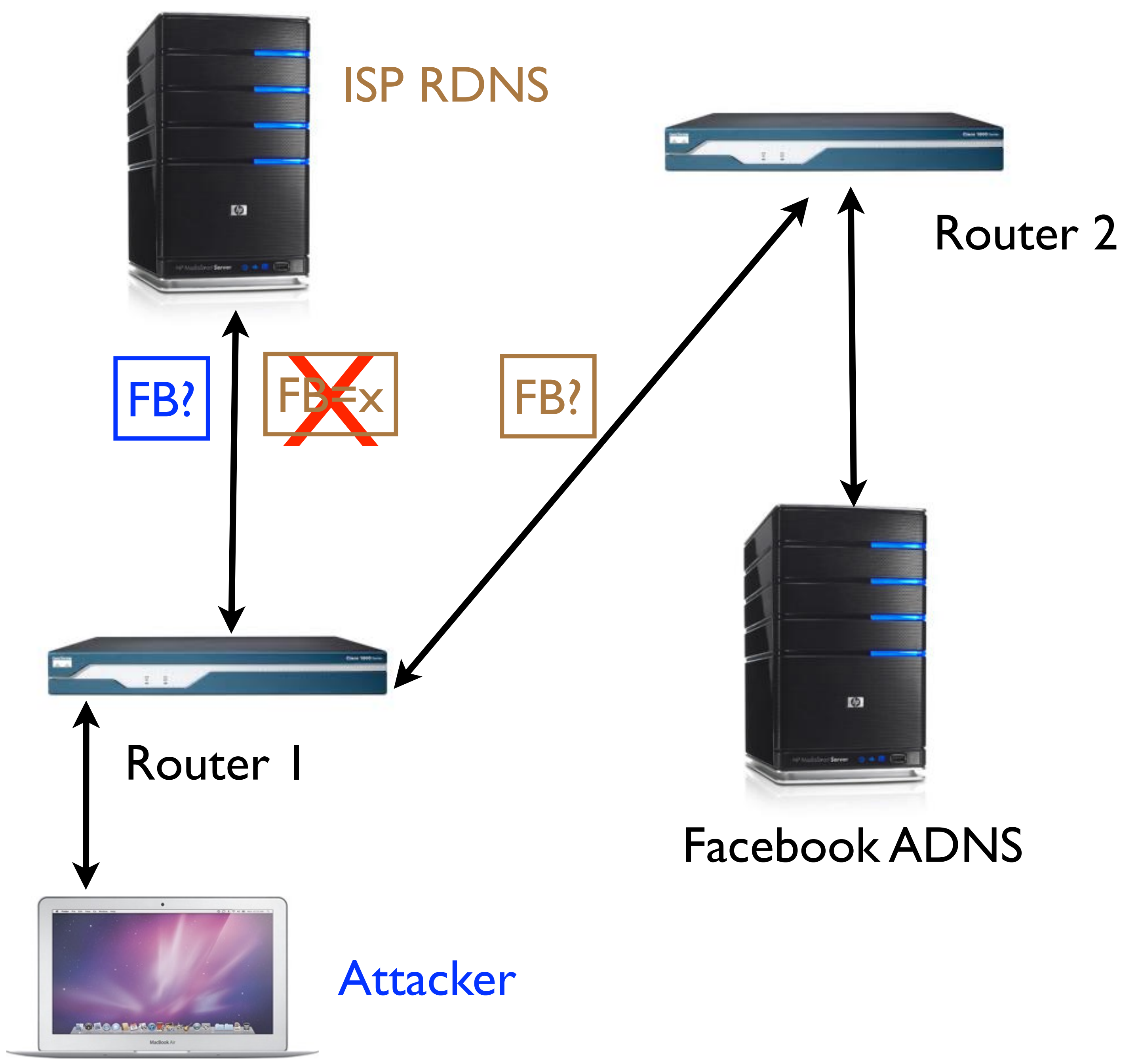
local IP: RDNS
remote IP: FB ADNS
local port: ???
remote port: 53
txID: ???
query: www.facebook.com



What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com

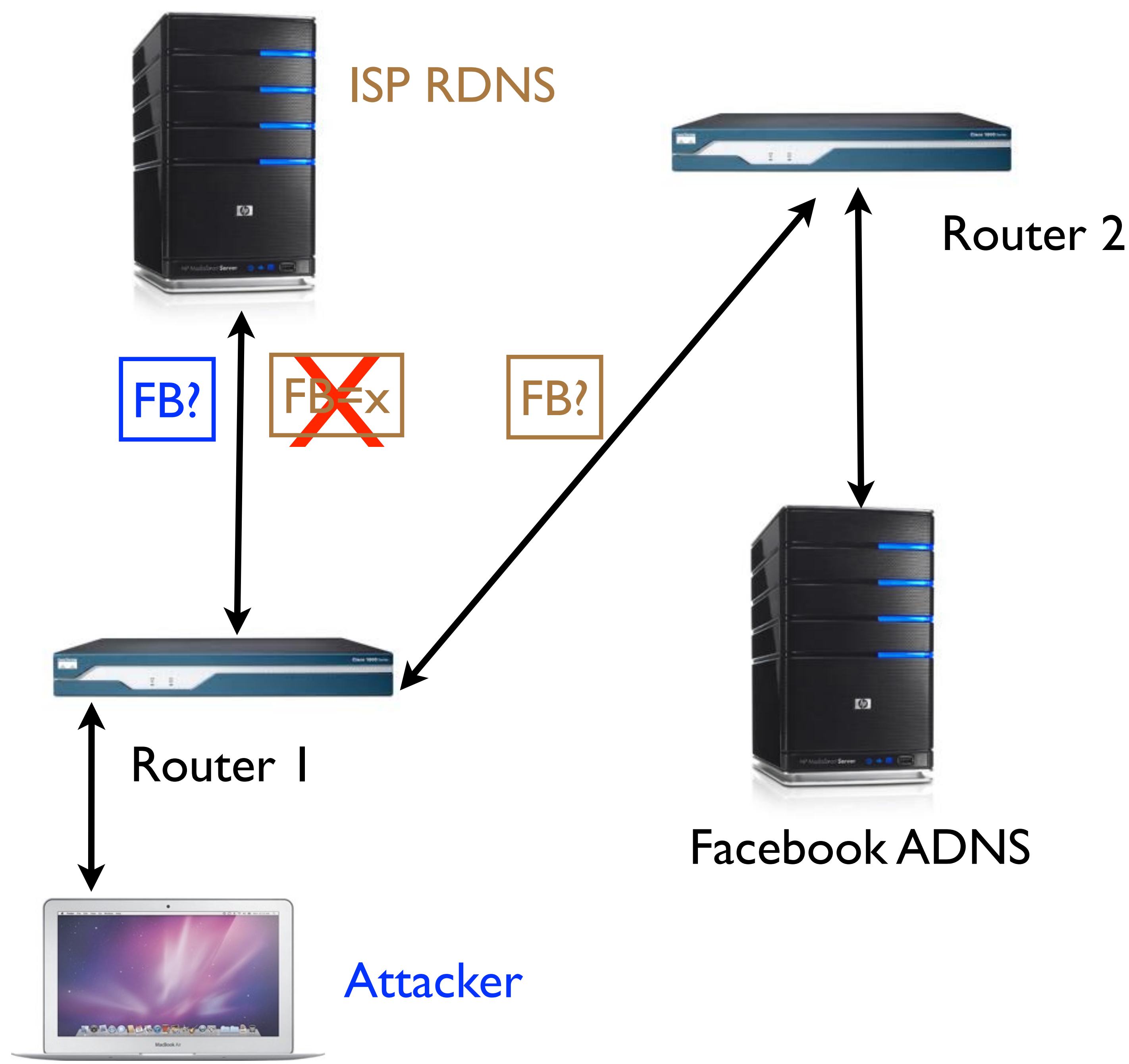
local IP: RDNS
remote IP: FB ADNS
local port: ???
remote port: 53
txID: ???
query: www.facebook.com



What Can Go Wrong? (4)

local IP: Client IP
remote IP: RDNS
local port: X
remote port: 53
txID:Y
query: www.facebook.com

local IP: RDNS
remote IP: FB ADNS
local port: ???
remote port: 53
txID: ???
query: www.facebook.com



Kaminsky Attack

Kaminsky Attack

- Theoretical space of unknowns: 4B
  (16 bit source port & 16 bit transaction ID)

Kaminsky Attack

- Theoretical space of unknowns: 4B
(16 bit source port & 16 bit transaction ID)

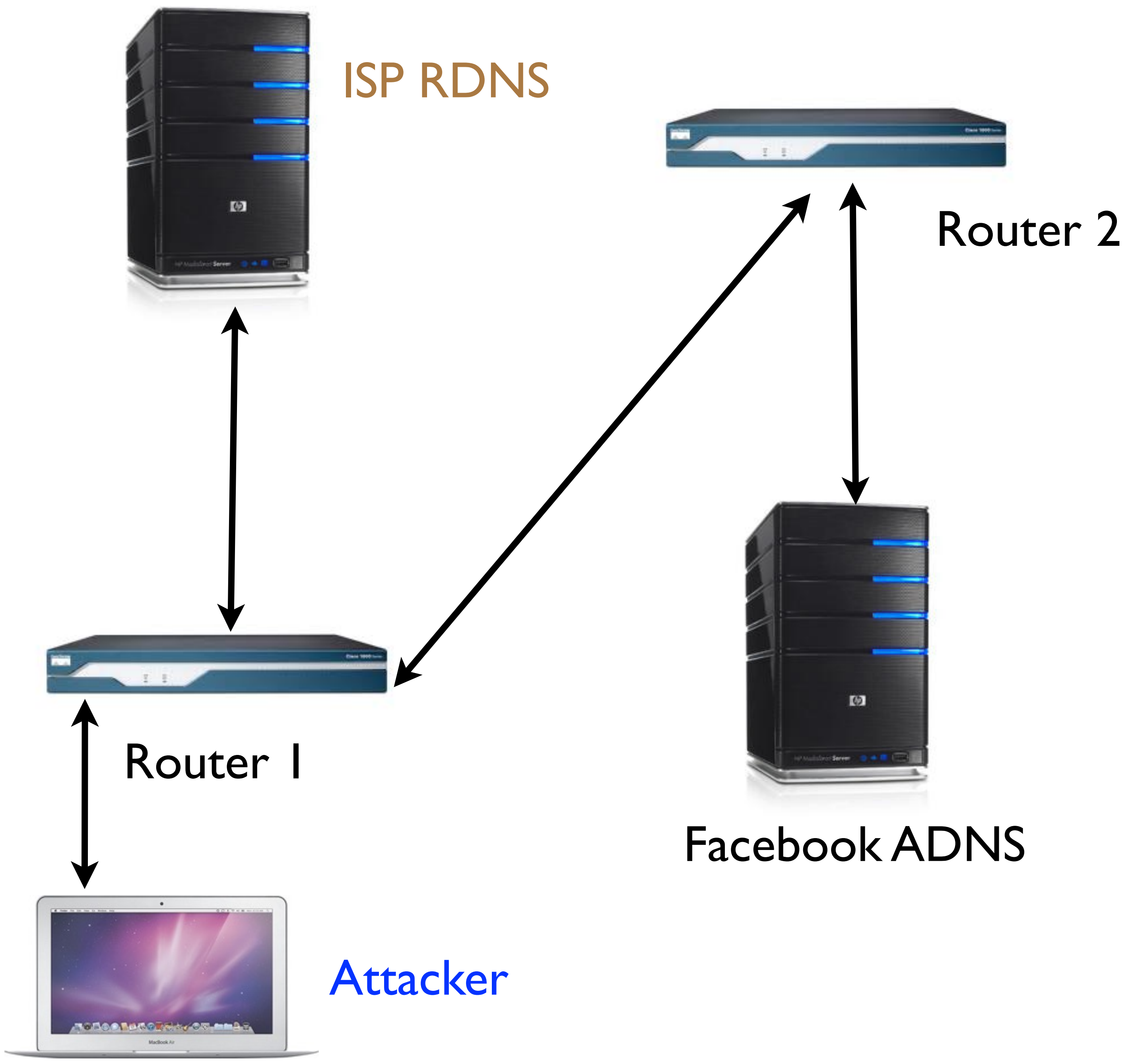
- 1 guess, P(success) = 0.0000000002

Kaminsky Attack

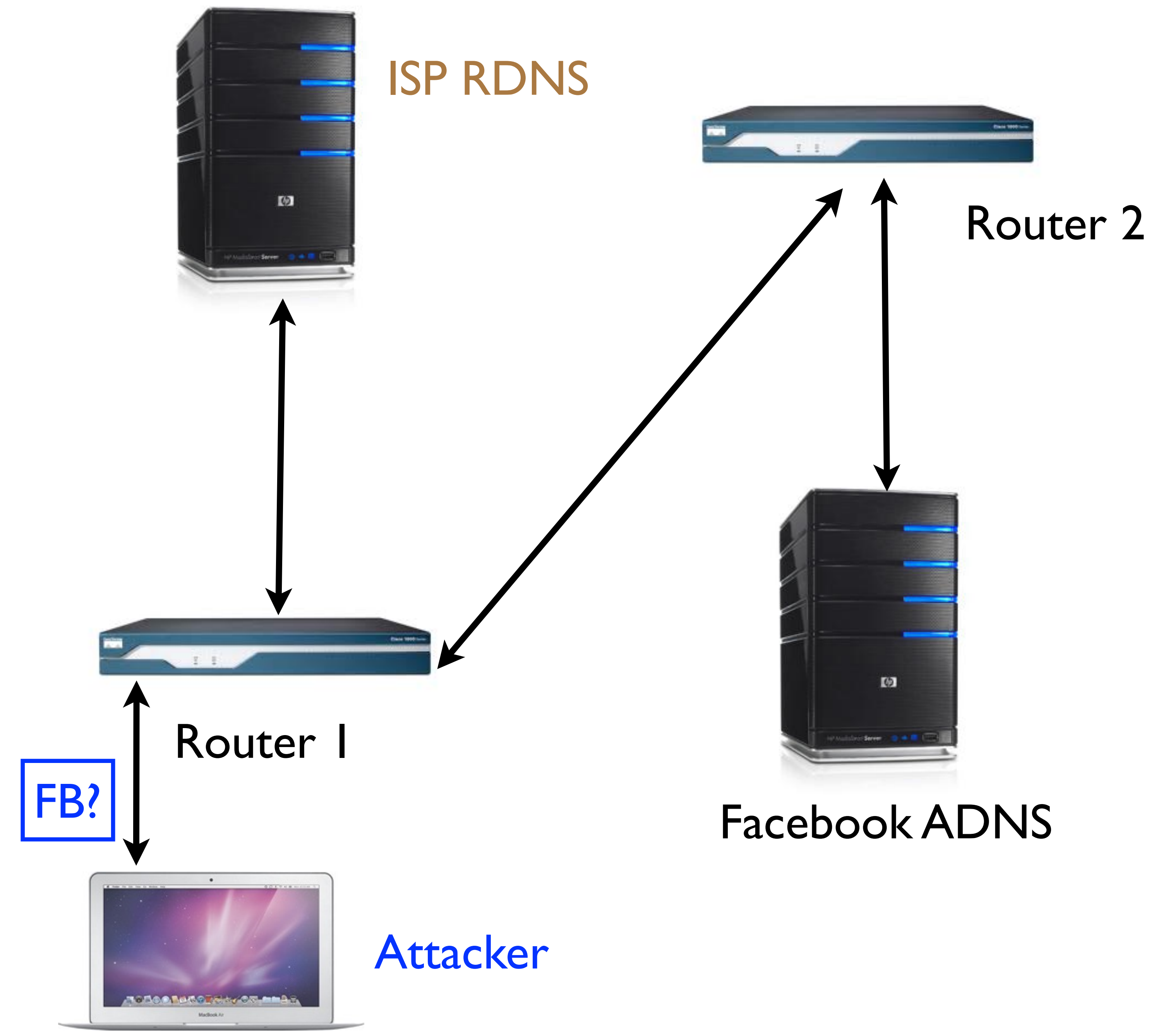
- Theoretical space of unknowns: 4B
  (16 bit source port & 16 bit transaction ID)

- 1 guess, P(success) = 0.0000000002
- 15 guesses, P(success) = 0.000000003

Kaminsky Attack

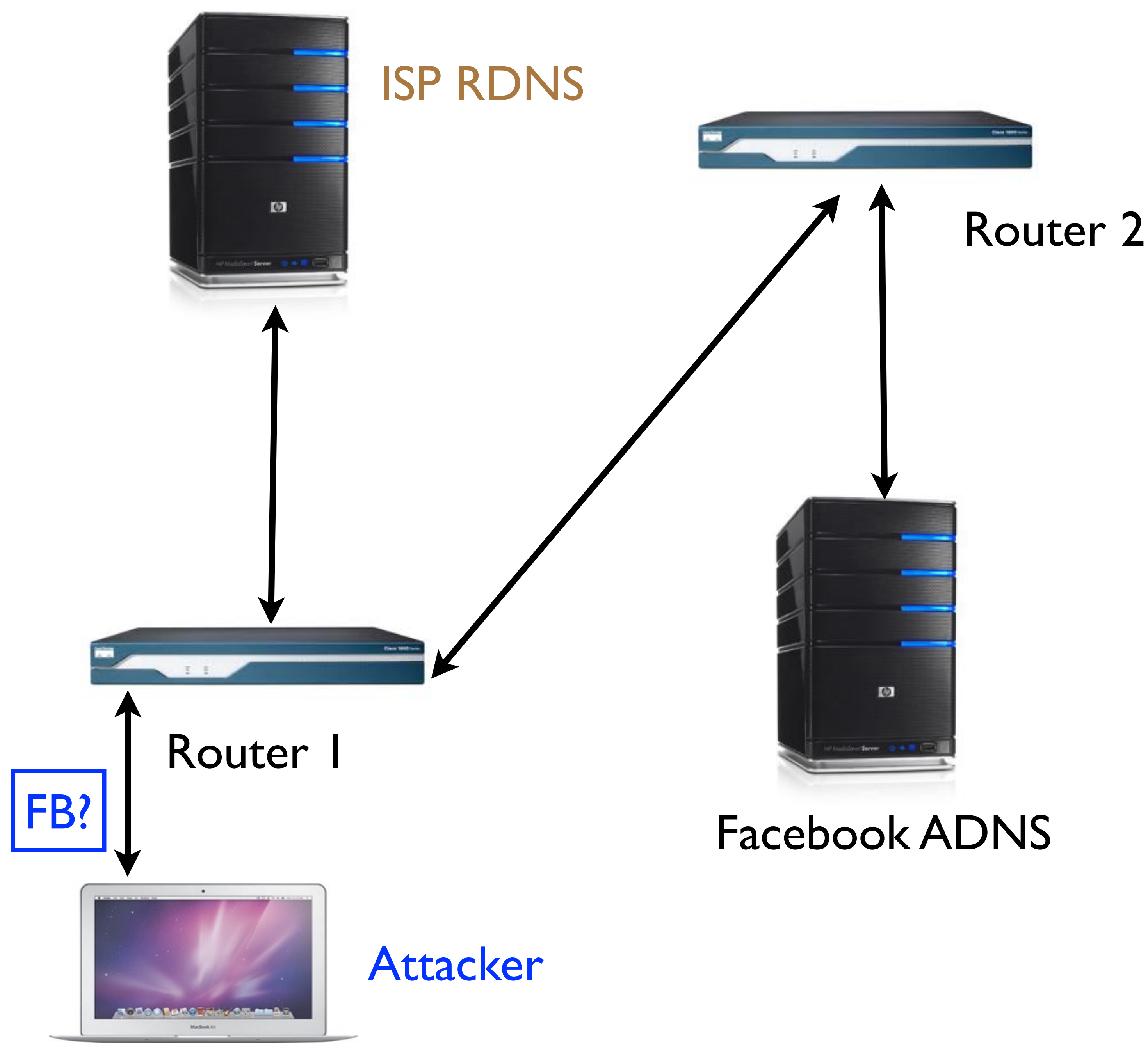


Kaminsky Attack



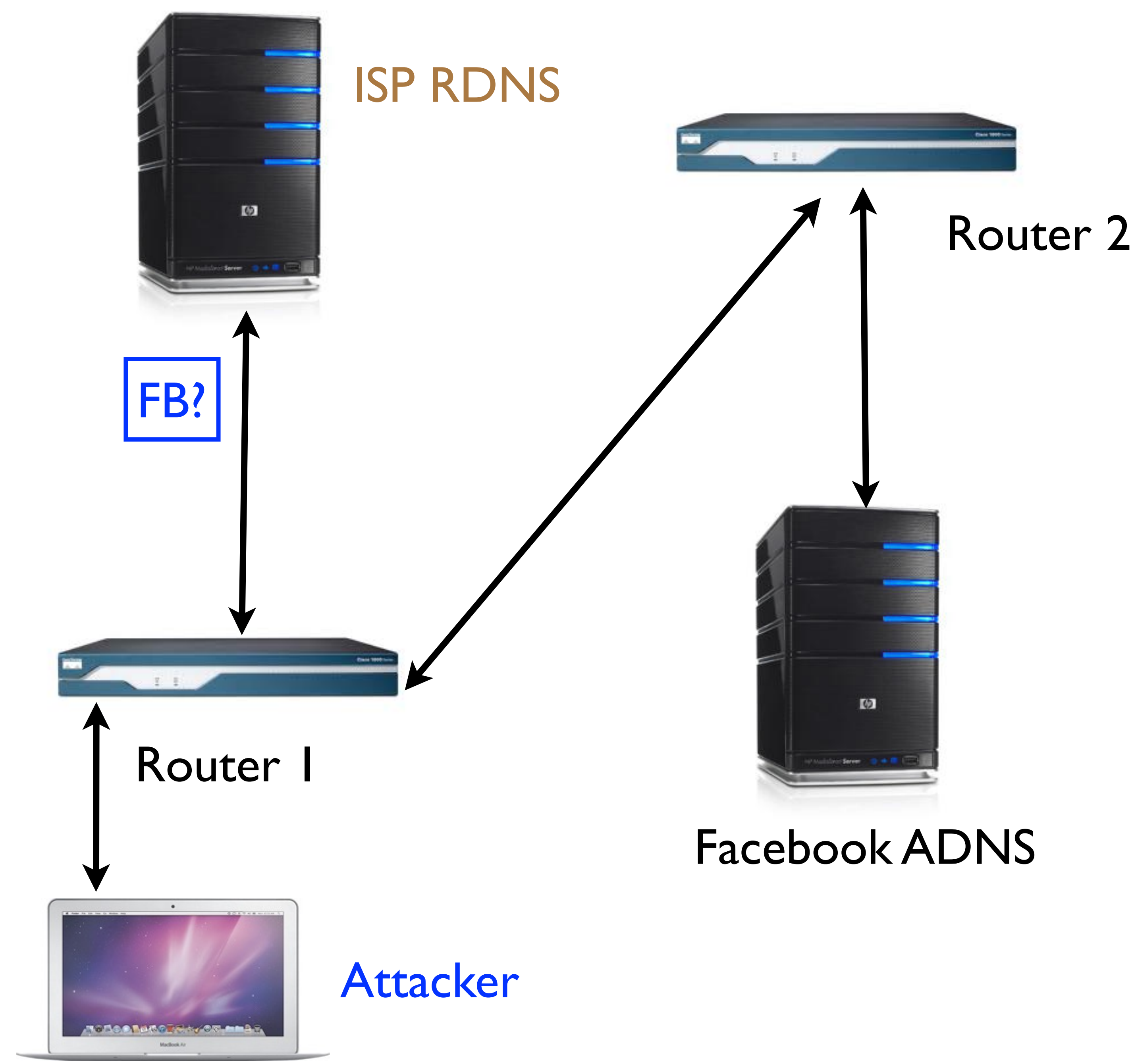
Kaminsky Attack

local IP: Client IP
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: 1.facebook.com



Kaminsky Attack

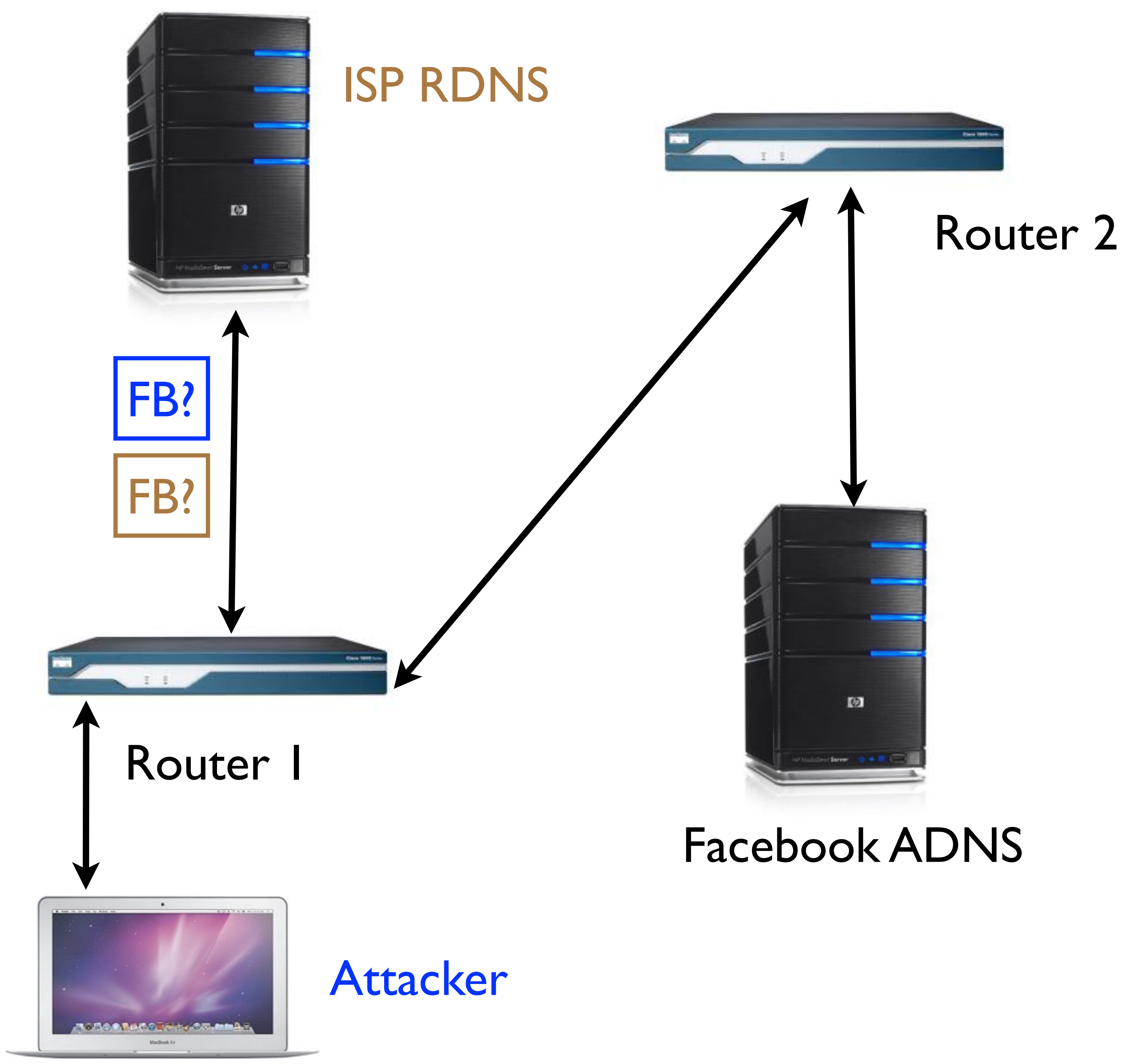
local IP: Client IP
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: 1.facebook.com



Kaminsky Attack

local IP: Client IP
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: 1.facebook.com

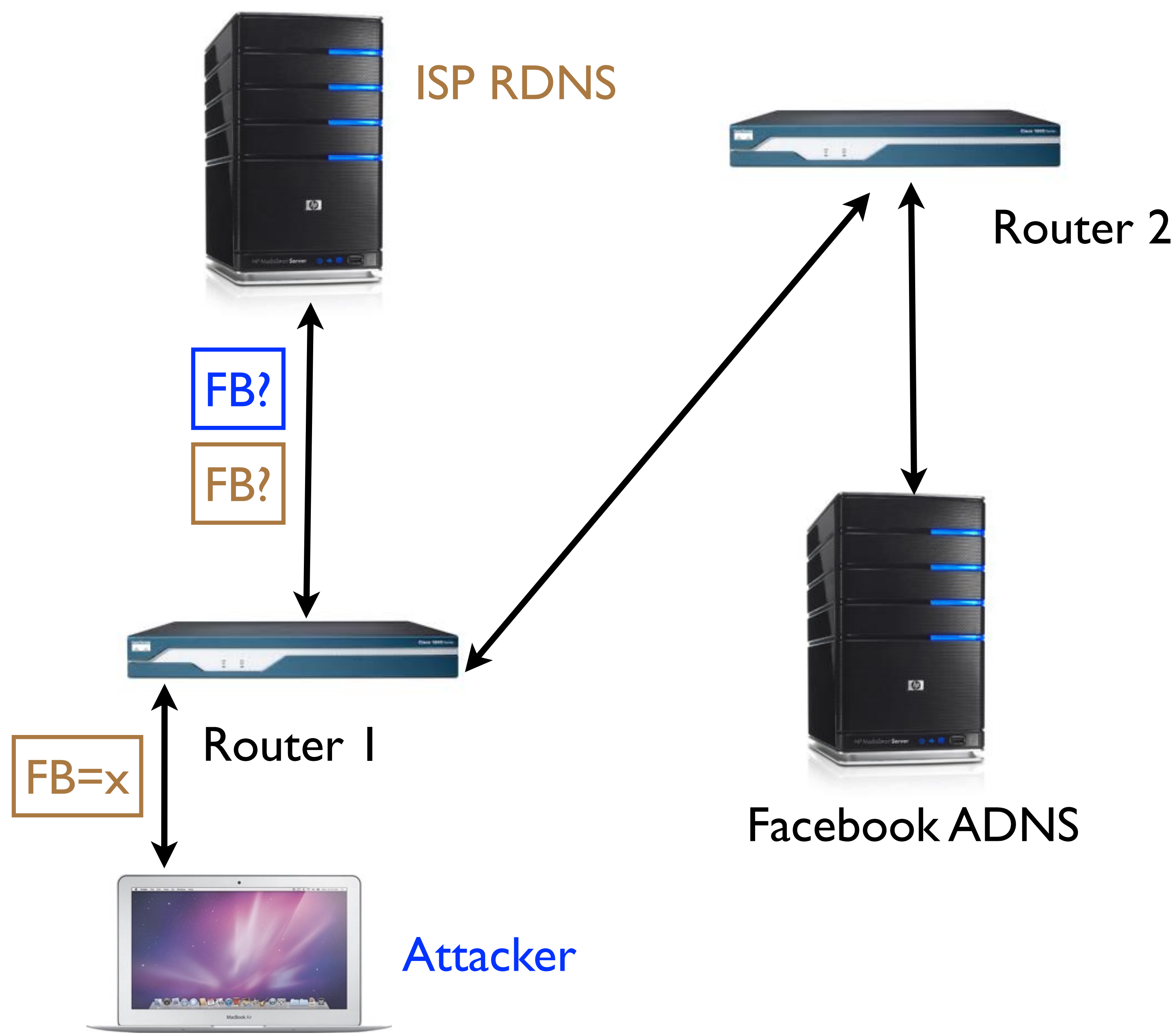
srcIP: RDNS
dstIP: Facebook ADNS
srcPort: ???
dstPort: 53
txID: ???
query: 1.facebook.com



Kaminsky Attack

local IP: Client IP
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: 1.facebook.com

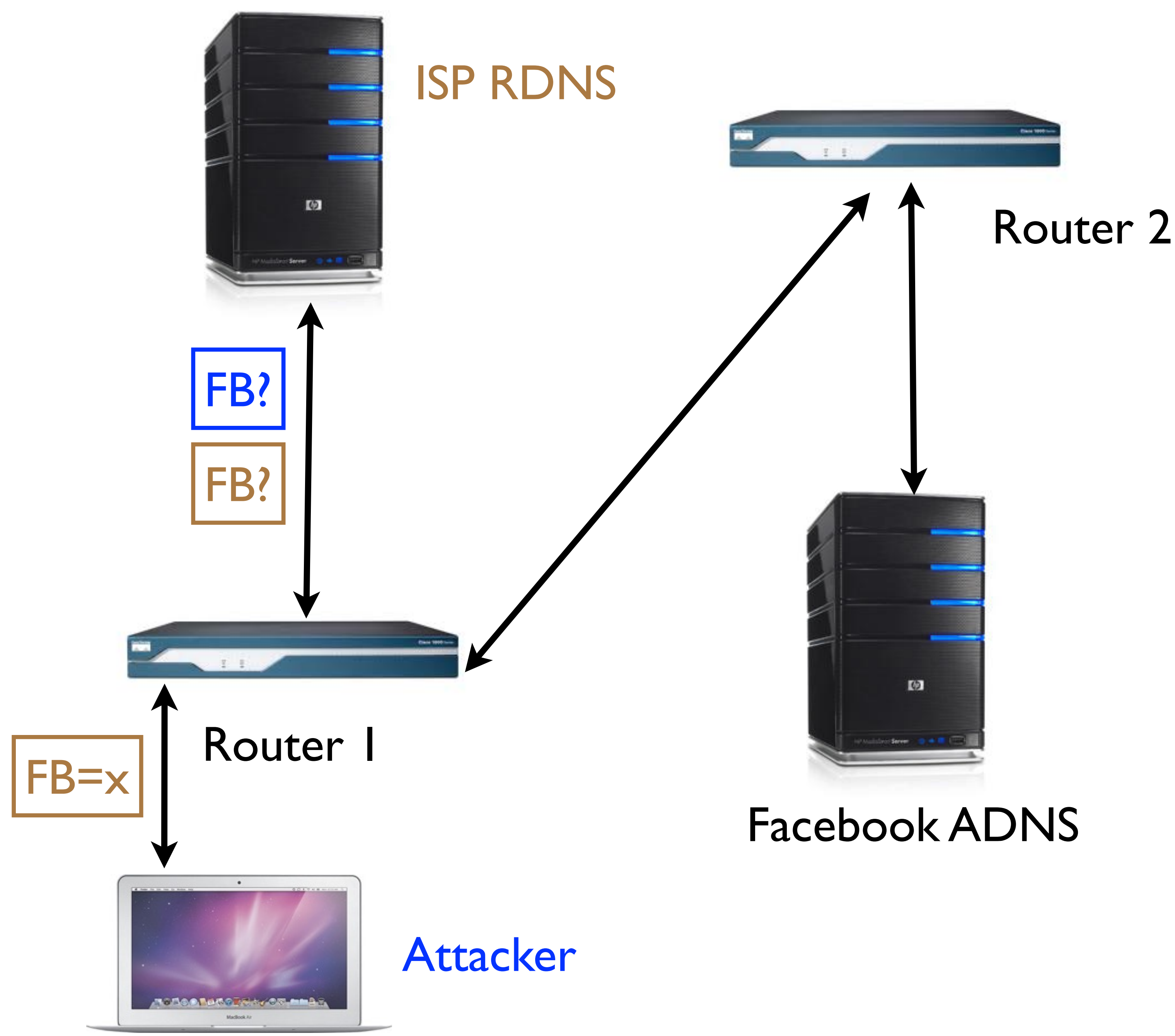
local IP: RDNS
remote IP: FB ADNS
local Port: 1 (guess)
remote Port: 53
txID: 1 (guess)
query: 1.facebook.com



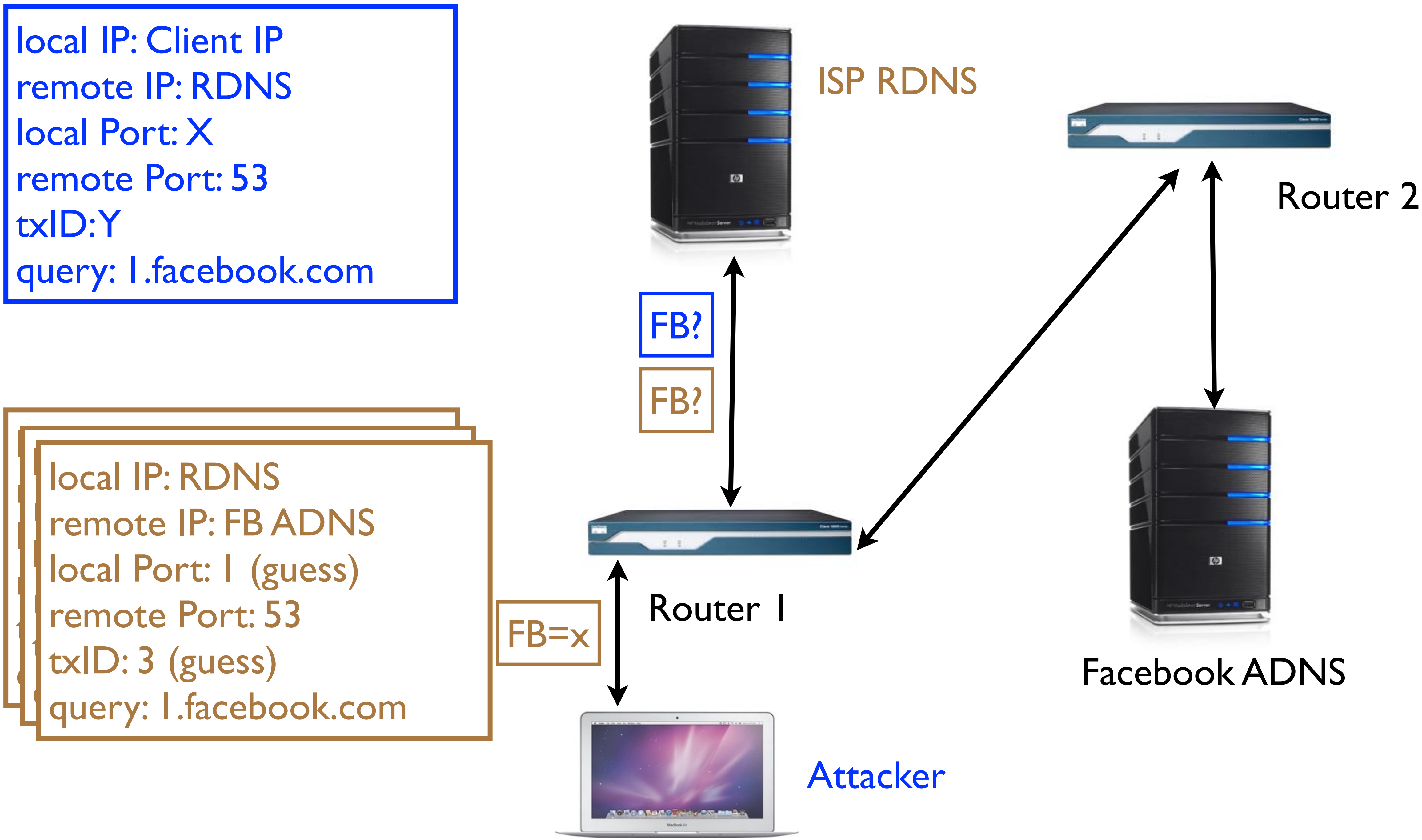
Kaminsky Attack

local IP: Client IP
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: 1.facebook.com

local IP: RDNS
remote IP: FB ADNS
local Port: 1 (guess)
remote Port: 53
txID: 2 (guess)
query: 1.facebook.com



Kaminsky Attack



Kaminsky Attack

Kaminsky Attack

- But, who cares about "1.facebook.com"?

Kaminsky Attack

- But, who cares about "1.facebook.com"?
- In each answer we include an "additional record" for "www.facebook.com"
  - brute force a valid response for an unused name
  - .... and "www.facebook.com" is cached too!

Kaminsky Attack

- But, who cares about "1.facebook.com"?
- In each answer we include an "additional record" for "www.facebook.com"
  - brute force a valid response for an unused name
  - .... and "www.facebook.com" is cached too!
- So, we get as many tries as we need!

Kaminsky Attack

Kaminsky Attack

- 1Mb/sec ≈ 300 guesses / sec

Kaminsky Attack

- 1Mb/sec ≈ 300 guesses / sec
  - After 1 sec, P(success) ≈ 0.00000006
  - After 1 min, P(success) ≈ 0.000004
  - After 1 day, P(success) ≈ 0.006
  - After 166 days, P(success) ≈ 1

Kaminsky Attack

- 1Mb/sec $\approx$ 300 guesses / sec
  - After 1 sec, P(success) $\approx$ 0.00000006
  - After 1 min, P(success) $\approx$ 0.000004
  - After 1 day, P(success) $\approx$ 0.006
  - After 166 days, P(success) $\approx$ 1
- Or, about 4 hours at 1Gb/sec

Kaminsky Attack

- 1Mb/sec $\approx$ 300 guesses / sec
  - After 1 sec, P(success) $\approx$ 0.00000006
  - After 1 min, P(success) $\approx$ 0.000004
  - After 1 day, P(success) $\approx$ 0.006
  - After 166 days, P(success) $\approx$ 1
- Or, about 4 hours at 1Gb/sec
- But, in reality it took 10min to mount the attack

Kaminsky: What To Do?

Kaminsky: What To Do?

- In the limit, nothing

Kaminsky: What To Do?

- In the limit, nothing
- We can make the attack more difficult by using all of the entropy *currently available*
  - i.e., entire port and transaction ID space

Kaminsky: What To Do?

- In the limit, nothing
- We can make the attack more difficult by using all of the entropy *currently available*
  - i.e., entire port and transaction ID space
- We can make the attack more difficult by finding new sources of entropy
  - e.g., 0x20 encoding

Kaminsky: Status

Kaminsky: Status

- Nearly all RDNS employ a complex (probably random) method for setting the transaction ID

Kaminsky: Status

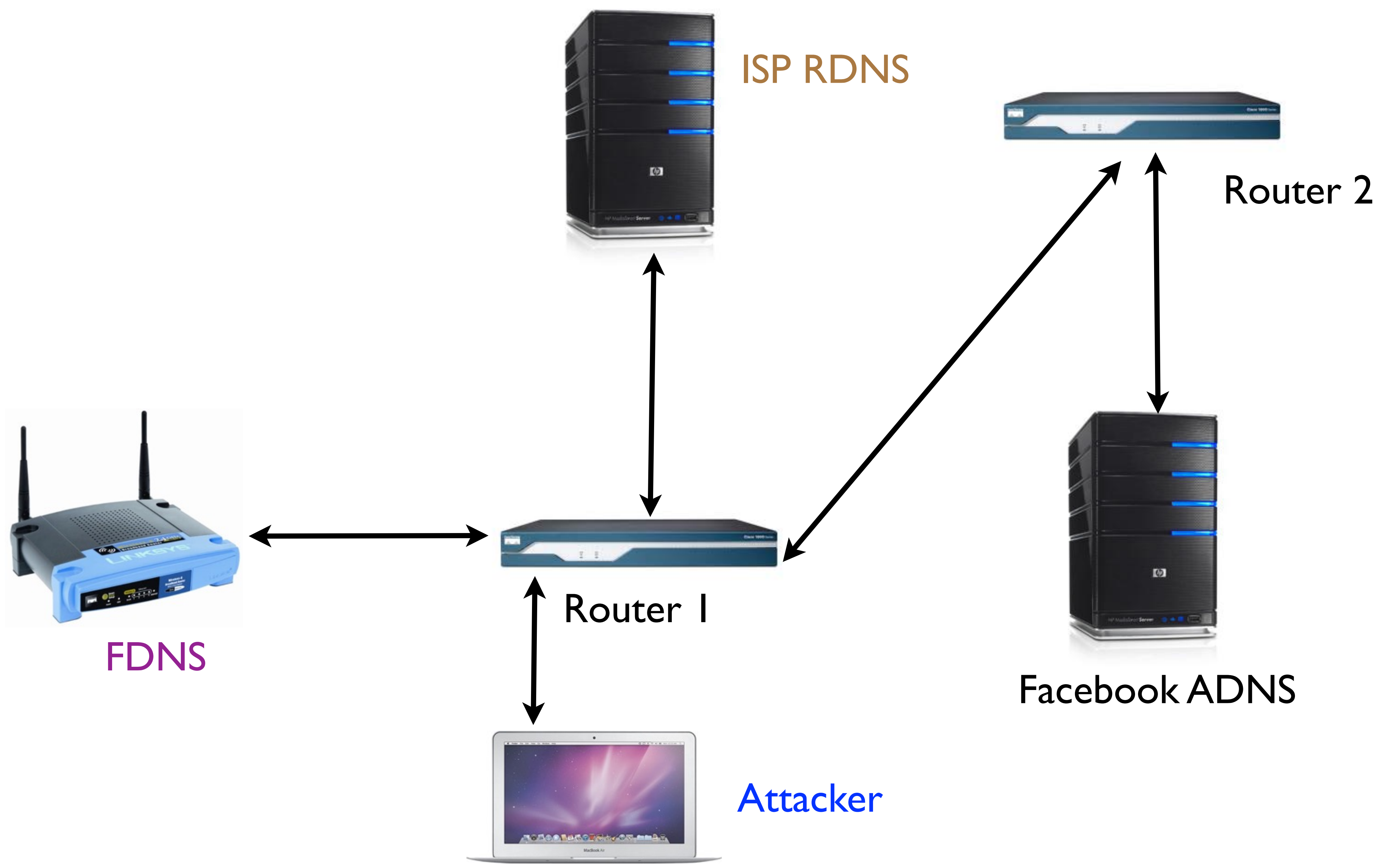
- Nearly all RDNS employ a complex (probably random) method for setting the transaction ID

- 84% of RDNS vary the ephemeral port
- 16% of RDNS *use a static ephemeral port!*
  - across 37% of the ASes

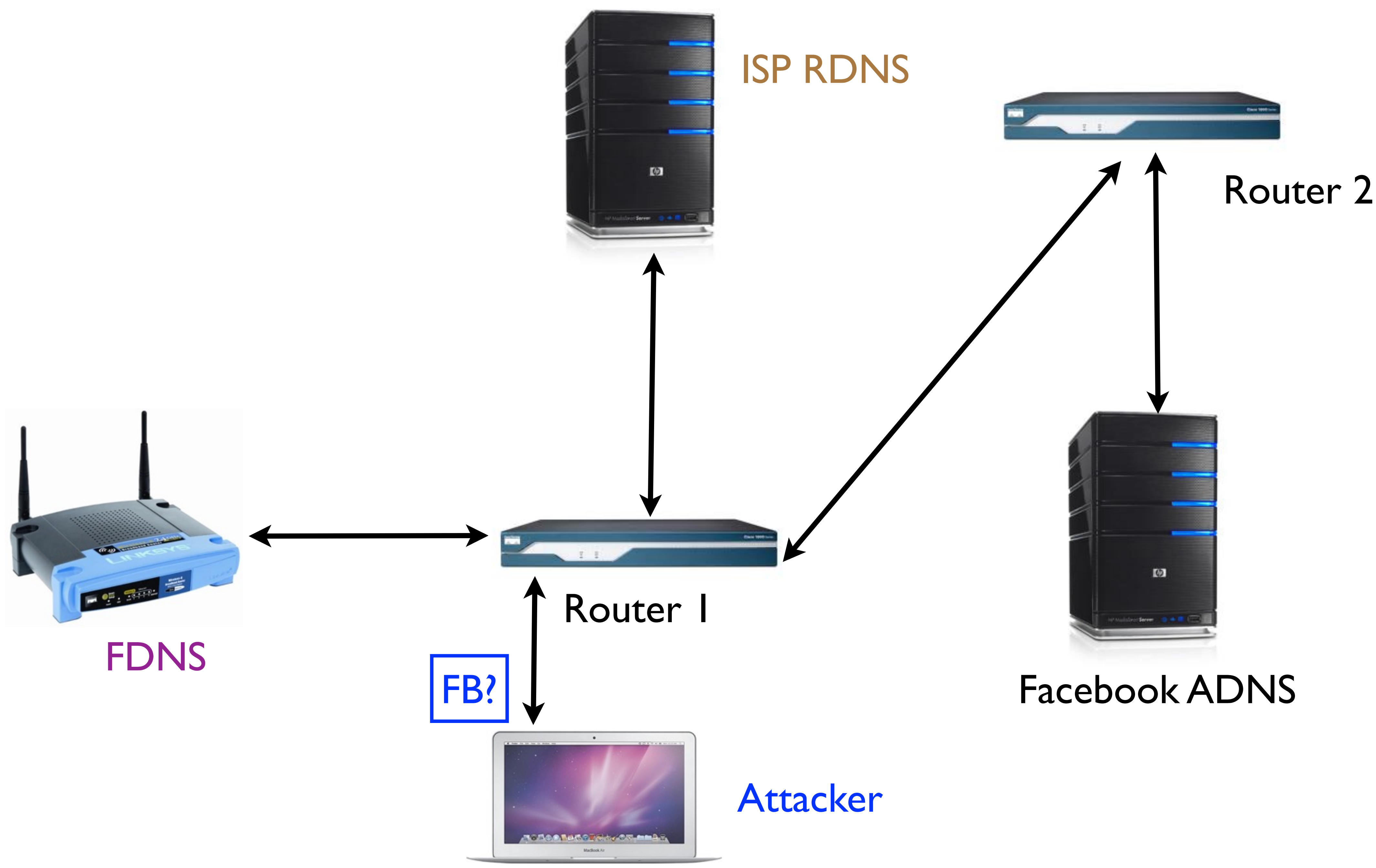
Kaminsky: Status

- Nearly all RDNS employ a complex (probably random) method for setting the transaction ID
- 84% of RDNS vary the ephemeral port
- 16% of RDNS *use a static ephemeral port!*
  - across 37% of the ASes
- 0.3% of RDNS use 0x20 encoding (lower bound ...)

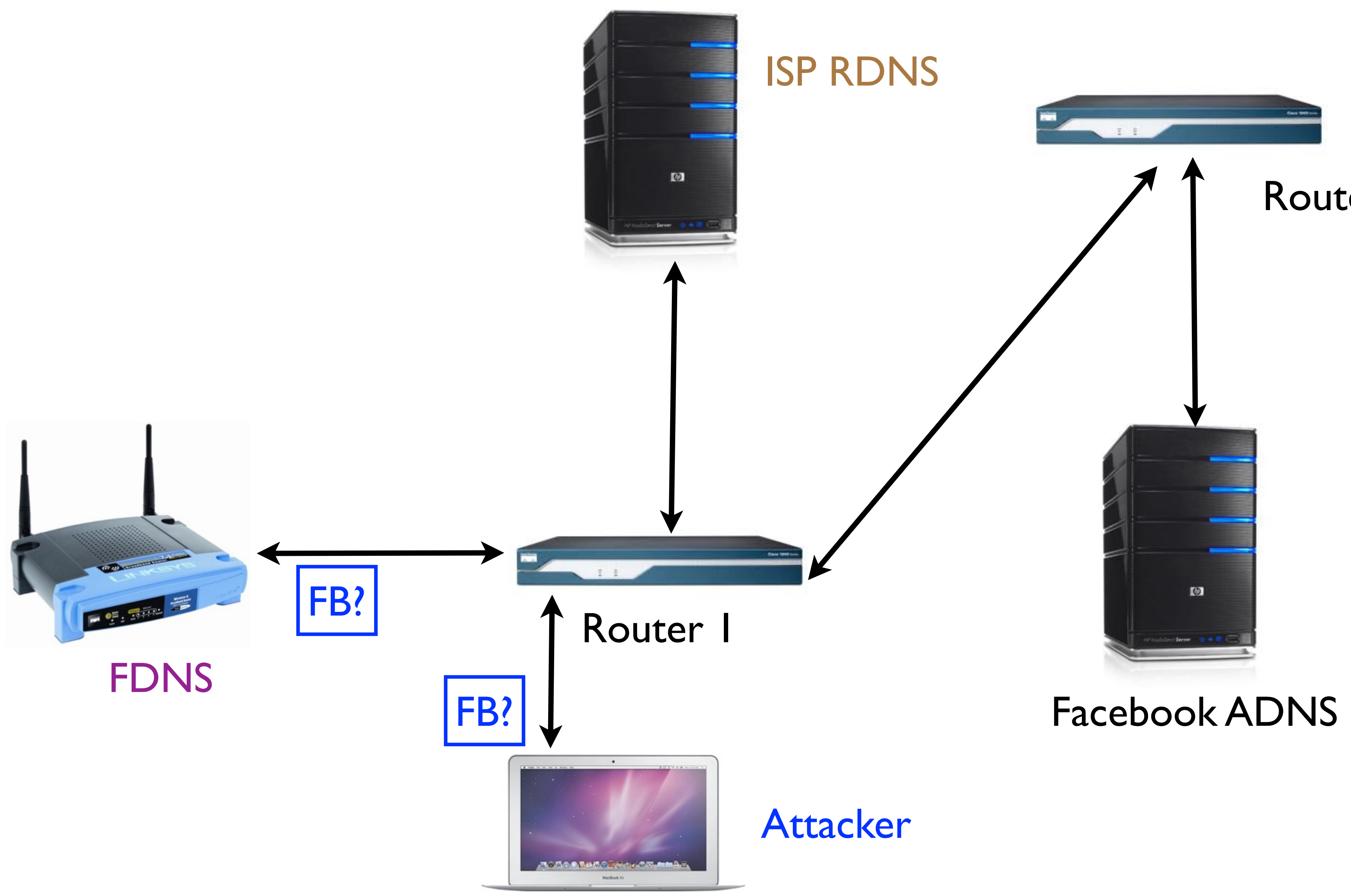
What Can Go Wrong? (5)



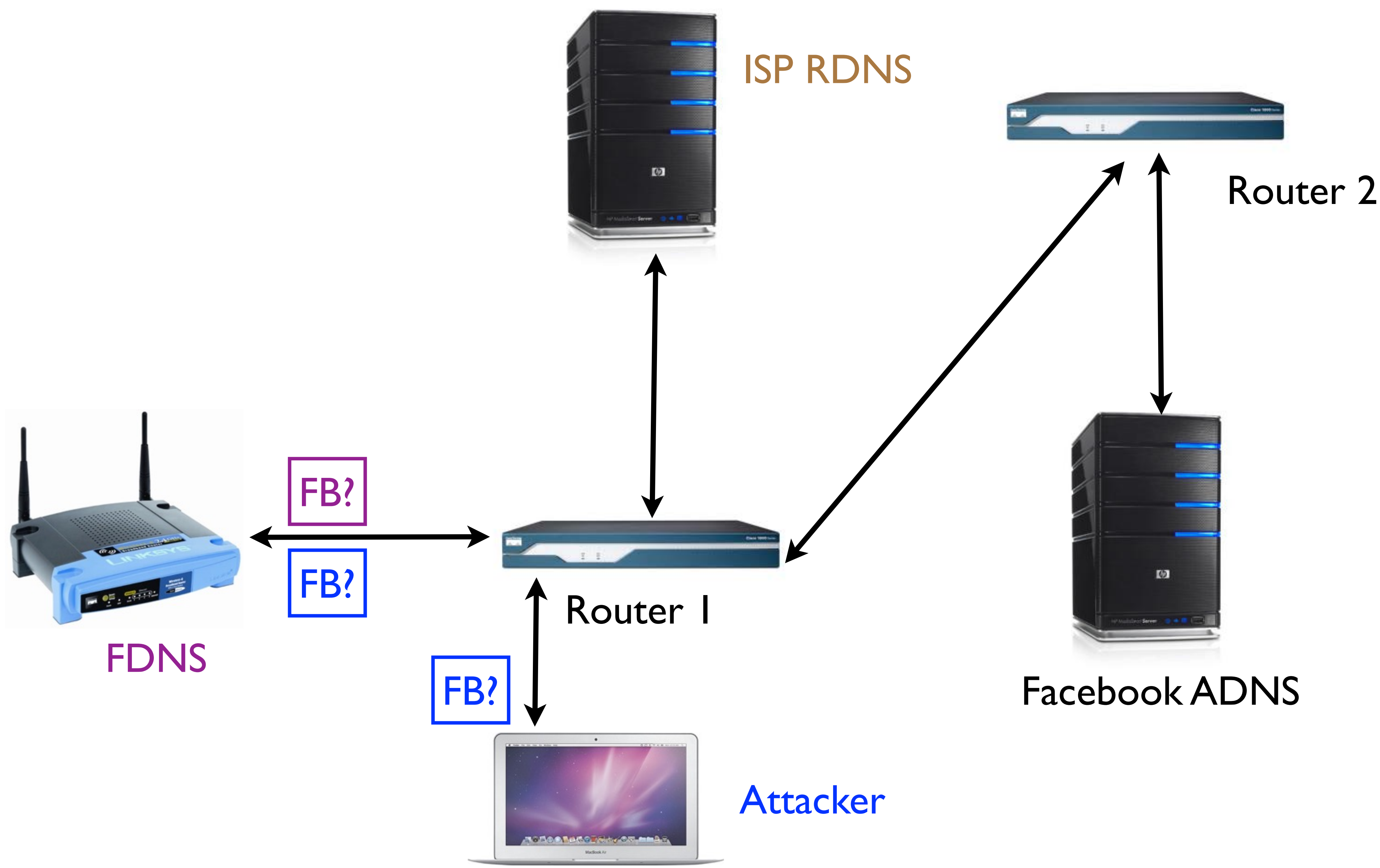
What Can Go Wrong? (5)



What Can Go Wrong? (5)

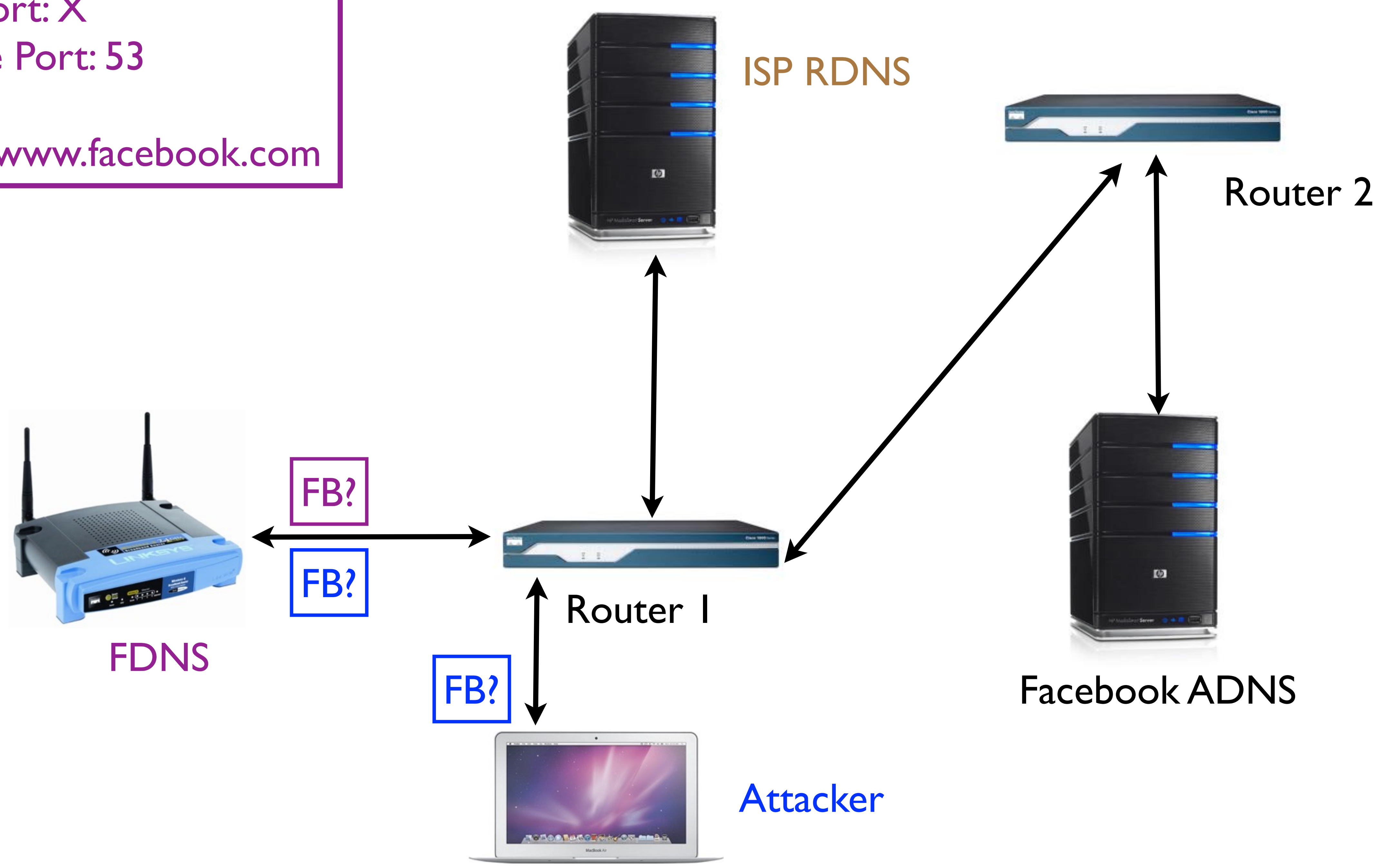


What Can Go Wrong? (5)



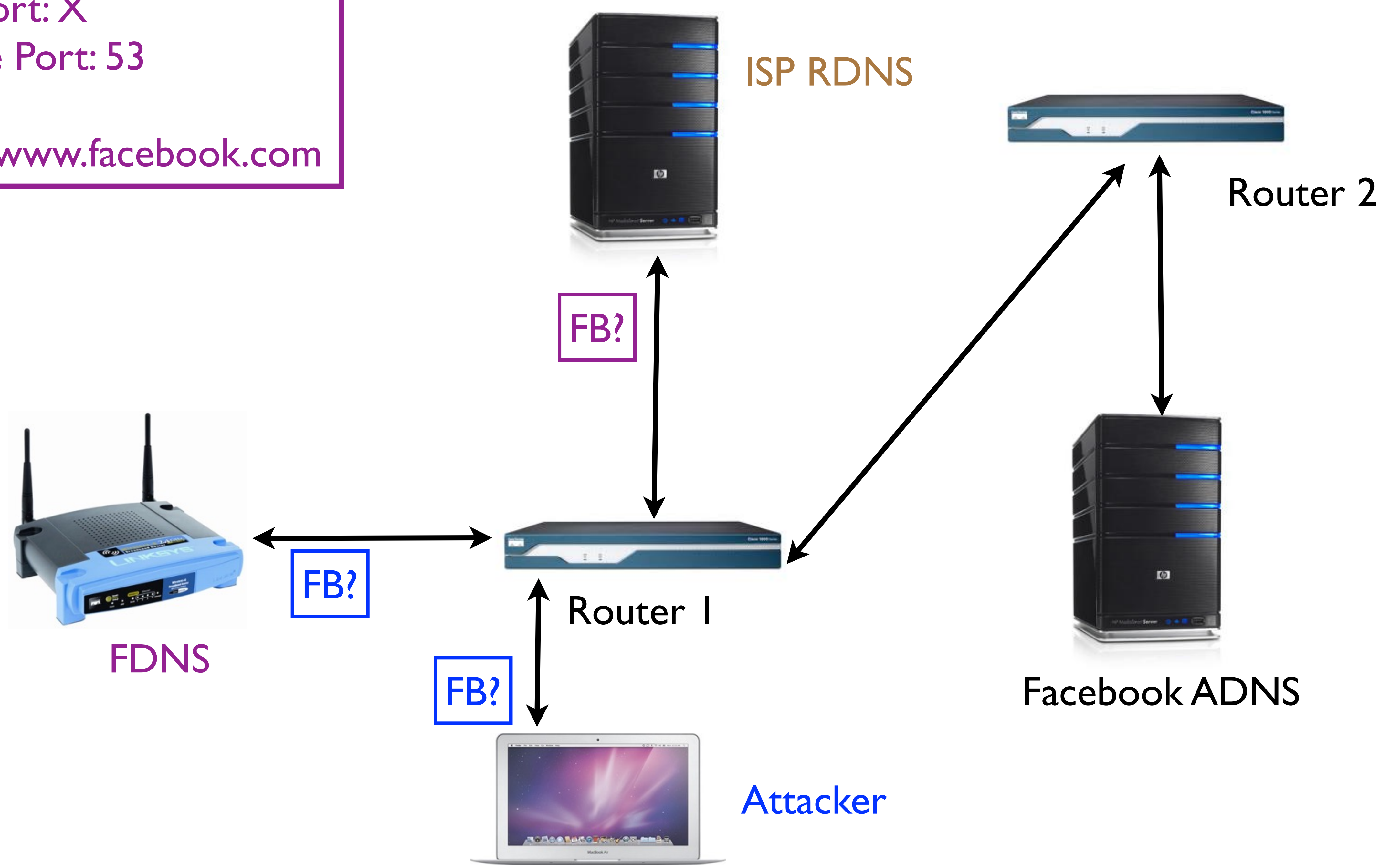
What Can Go Wrong? (5)

local IP: FDNS
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: www.facebook.com



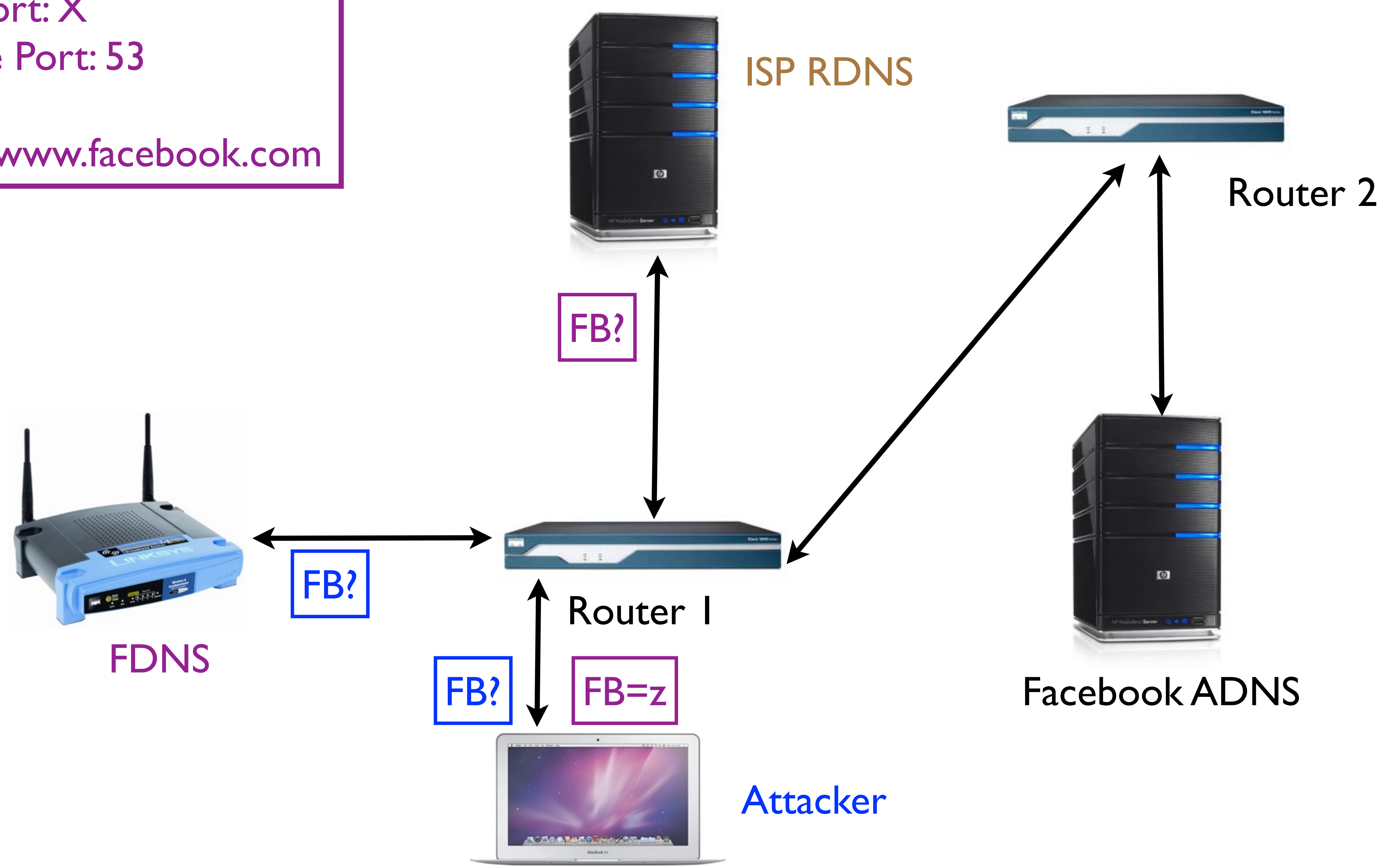
What Can Go Wrong? (5)

local IP: FDNS
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: www.facebook.com



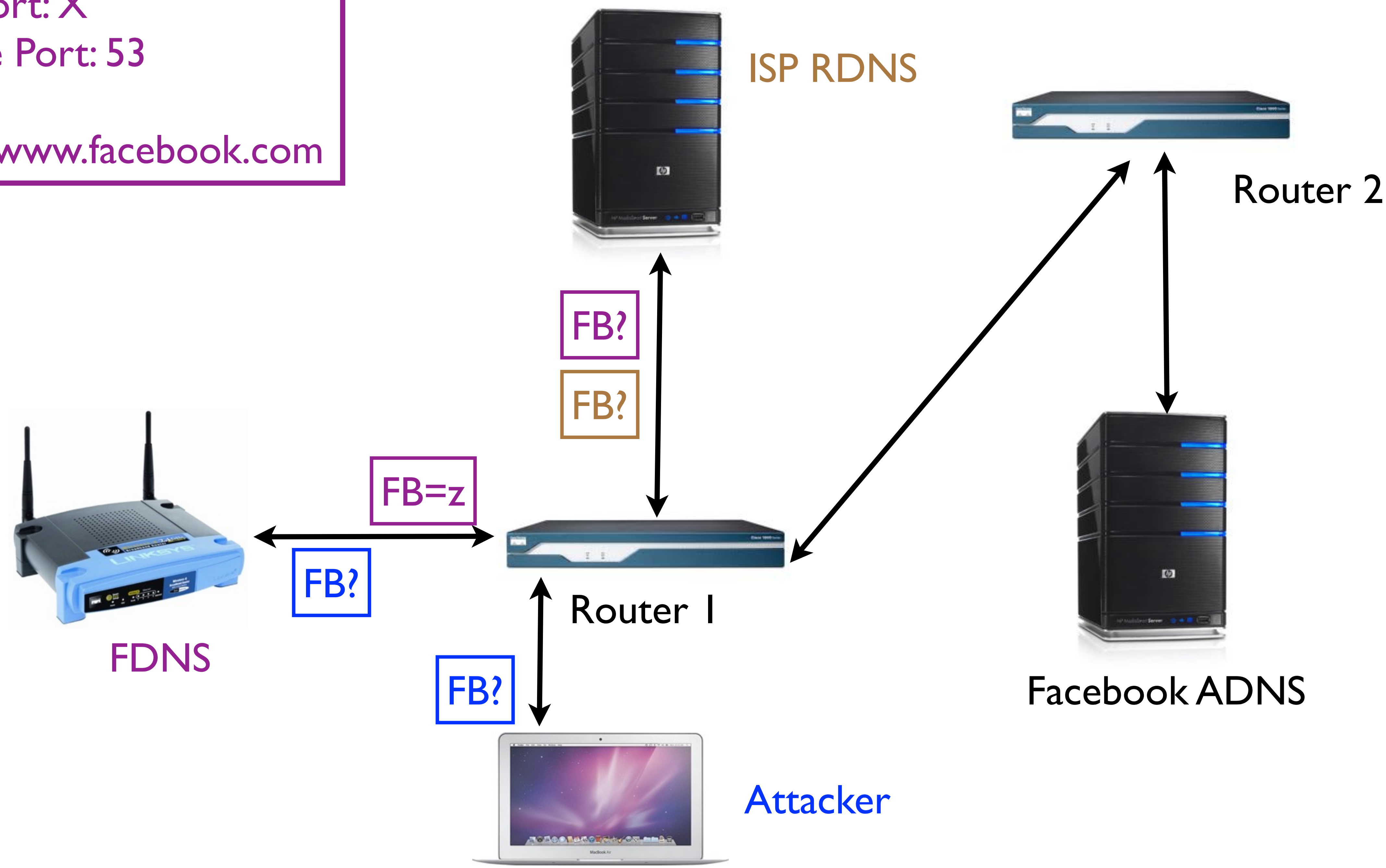
What Can Go Wrong? (5)

local IP: FDNS
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: www.facebook.com



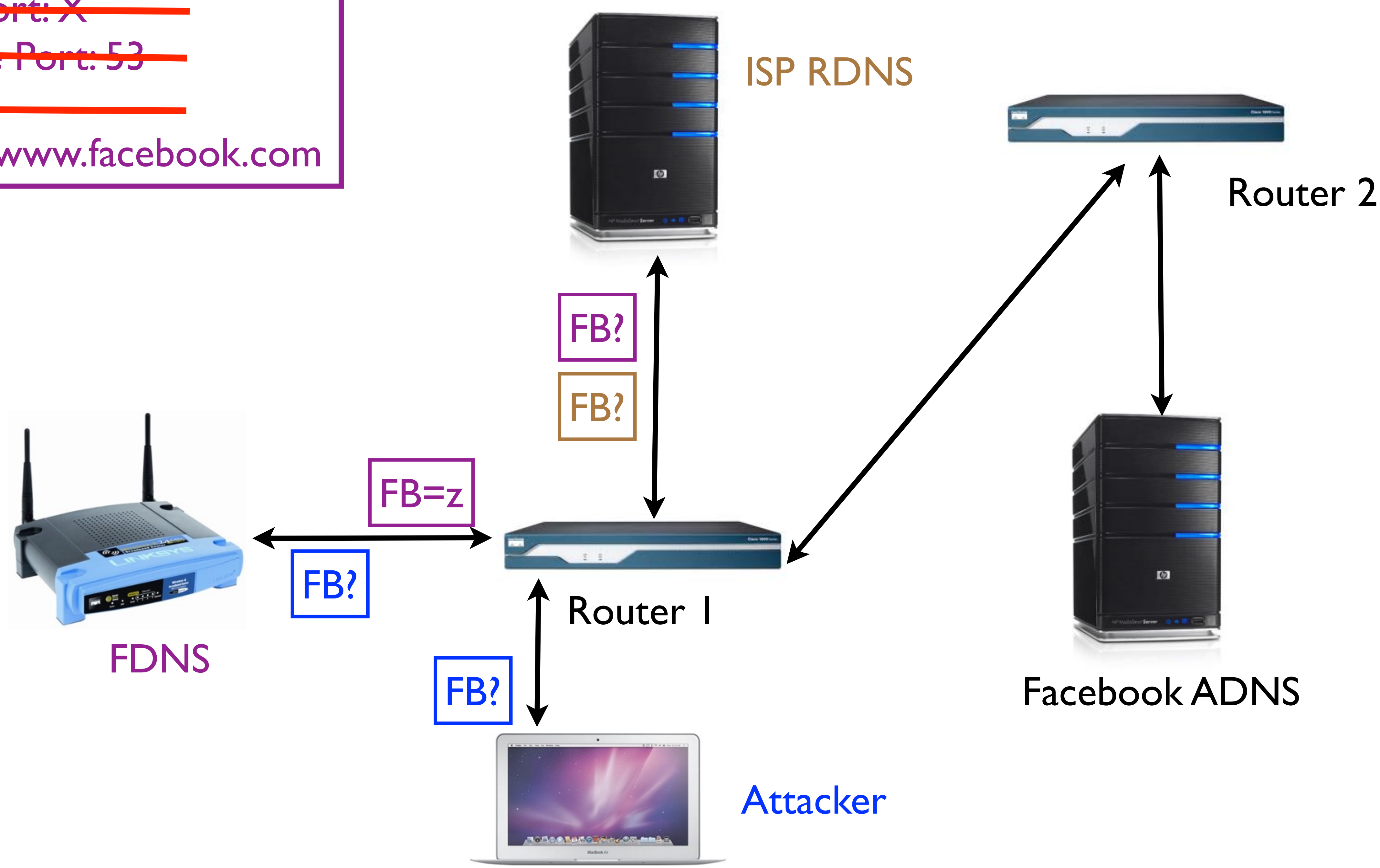
What Can Go Wrong? (5)

local IP: FDNS
remote IP: RDNS
local Port: X
remote Port: 53
txID:Y
query: www.facebook.com



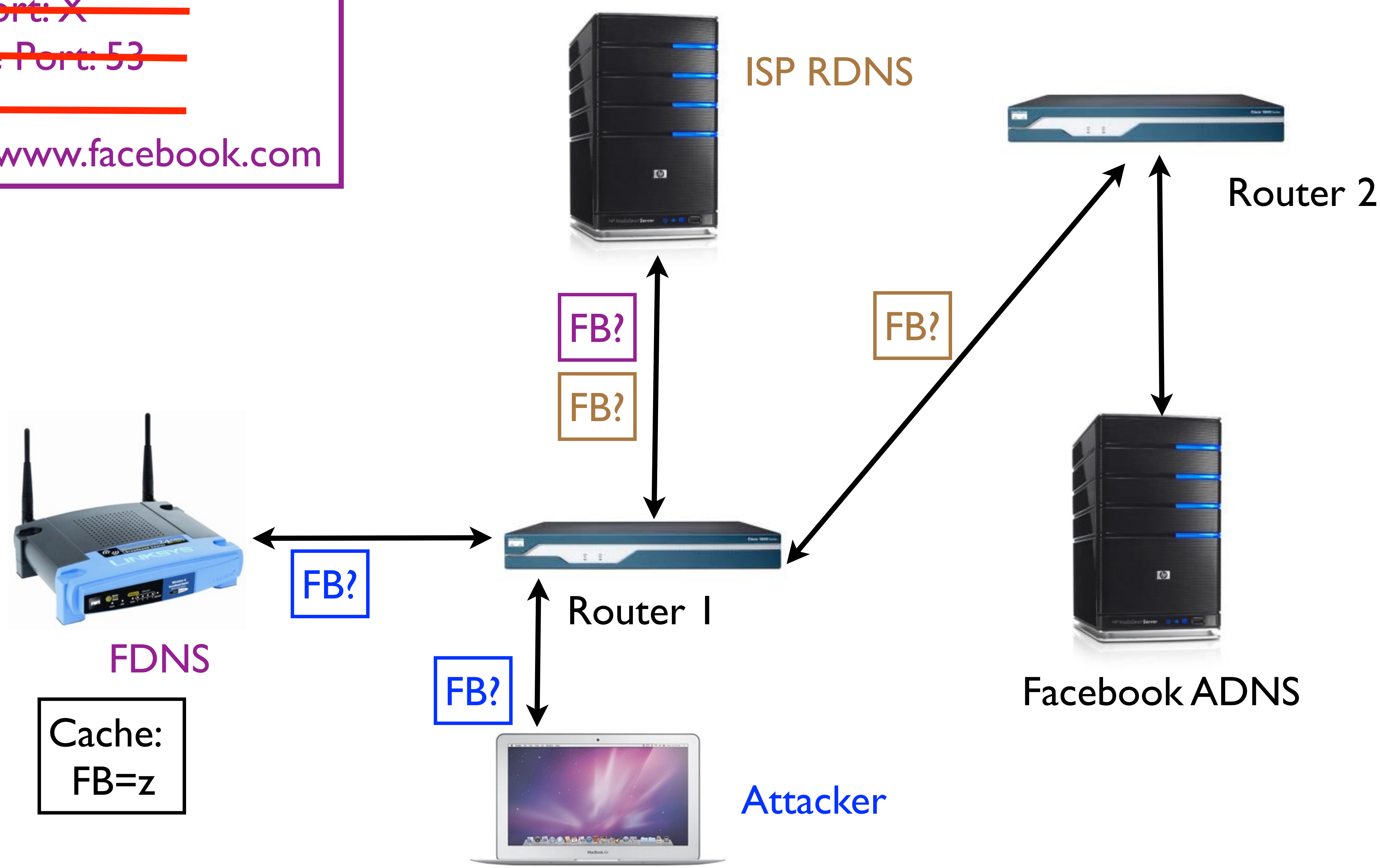
What Can Go Wrong? (5)

local IP: FDNS
~~remote IP: RDNS~~
~~local Port: X~~
~~remote Port: 53~~
~~txID:Y~~
query: www.facebook.com



What Can Go Wrong? (5)

local IP: FDNS
~~remote IP: RDNS~~
~~local Port: X~~
~~remote Port: 53~~
~~txID:Y~~
query: www.facebook.com



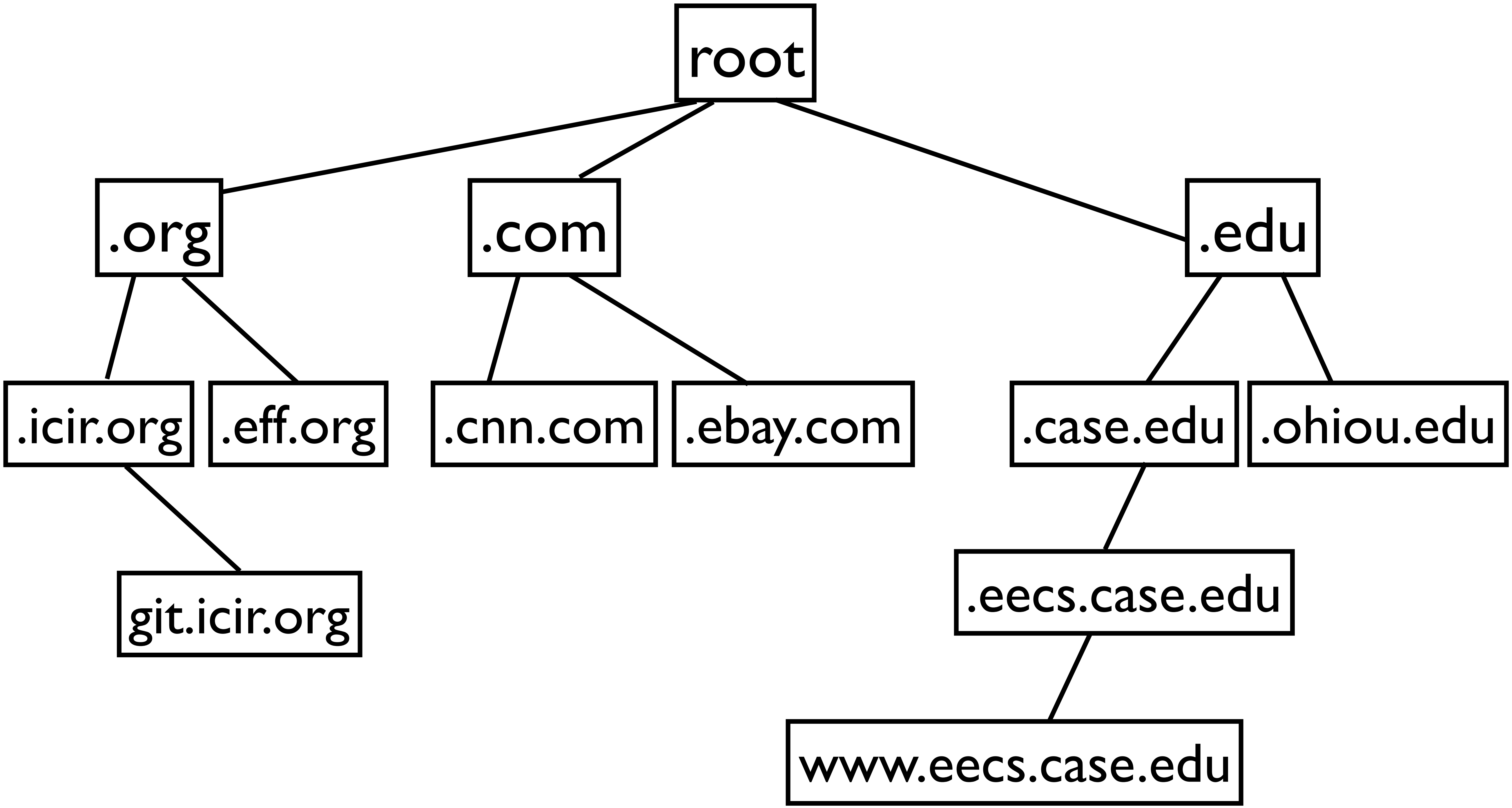
Preplay Attack

- We find 7--9% of open resolvers to be vulnerable to the preplay attack
  - i.e., about 2 million open resolvers

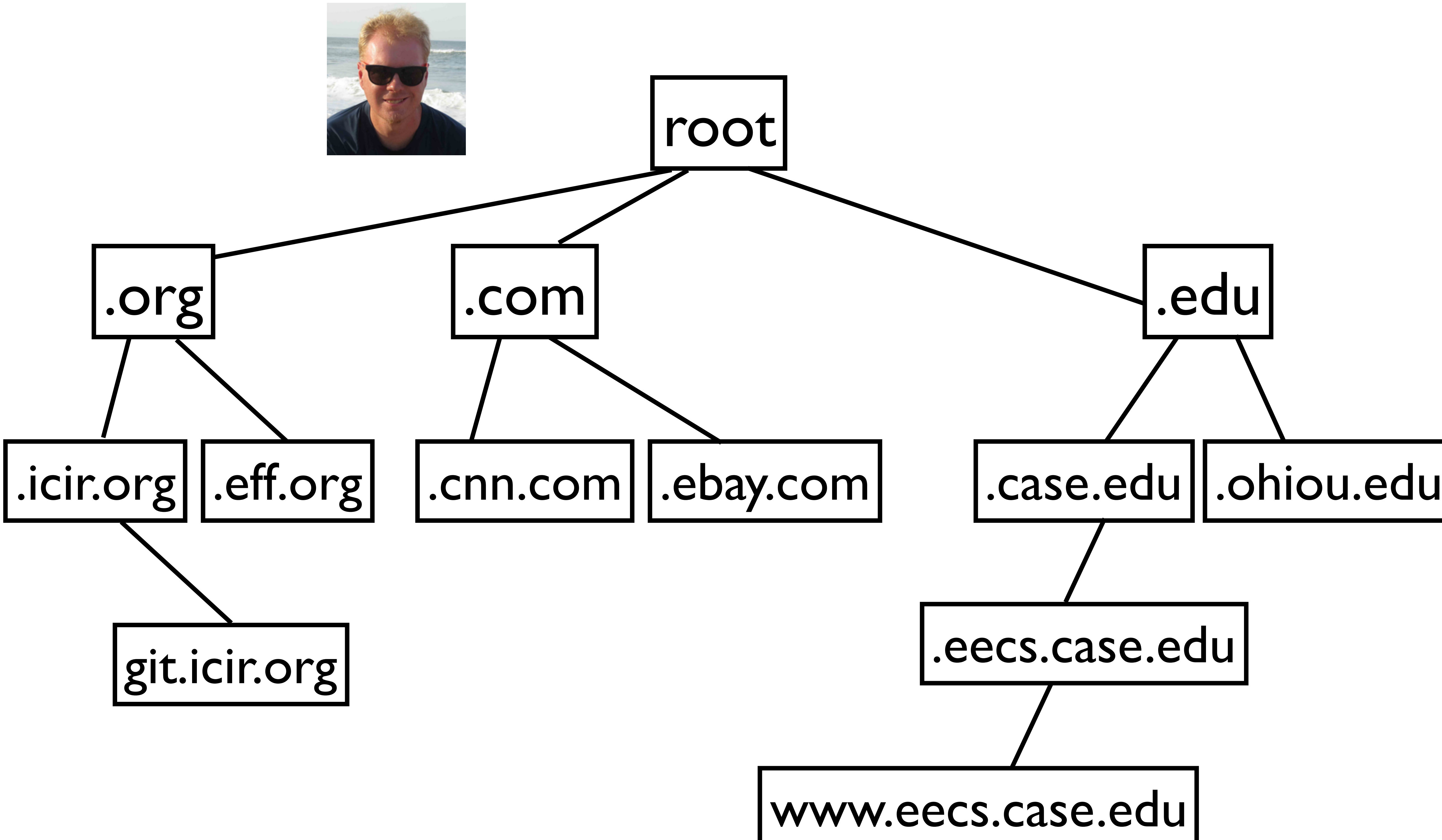
What Can Go Wrong? (6)

- Roots are the most crucial part of the DNS hierarchy
- And, have a *small footprint*
  - 13 logical servers
- Hence, an attacker that can usurp control of the roots or can manipulate traffic to/from the roots can have a tremendous impact

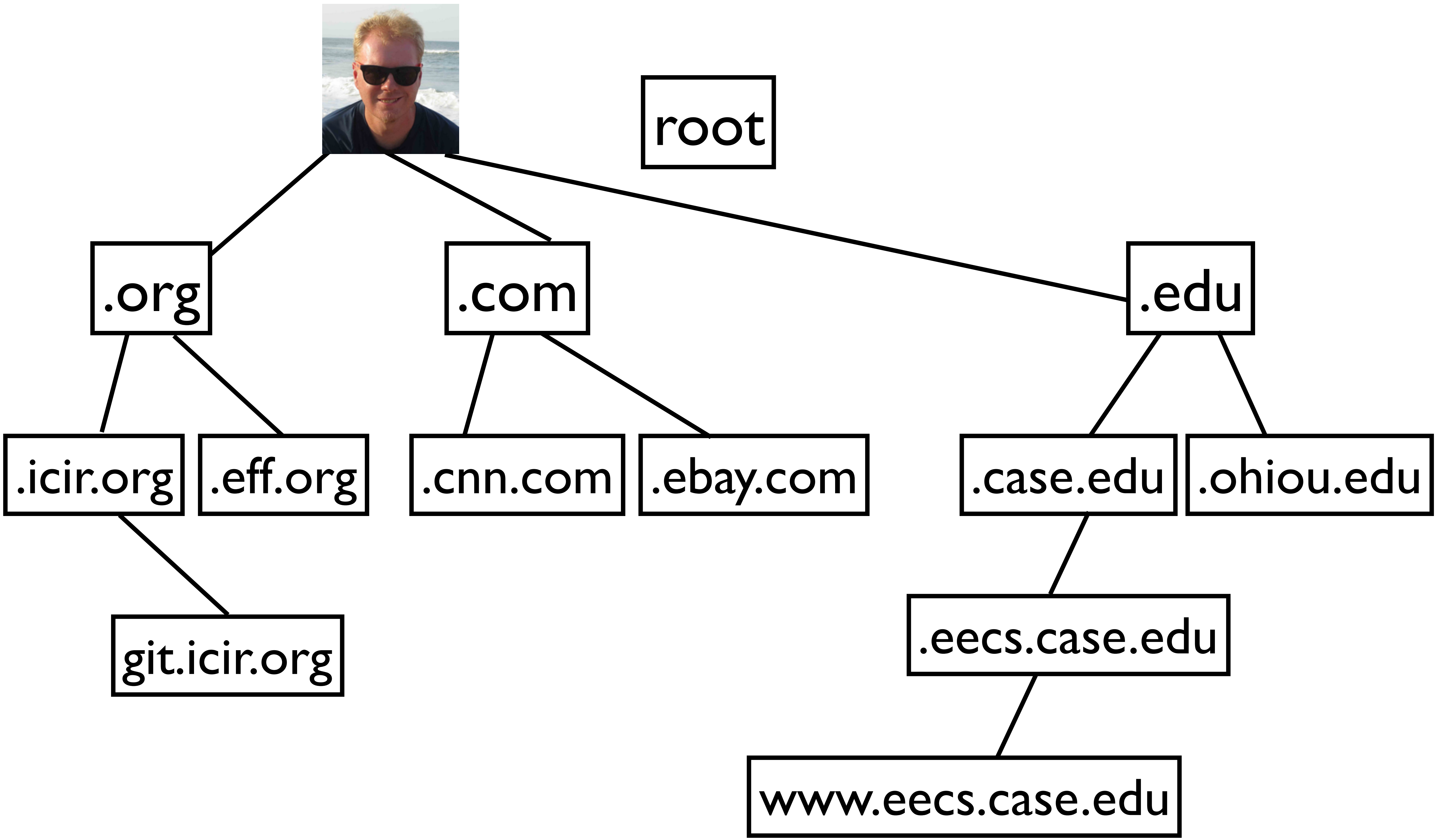
DNS Root Manipulation



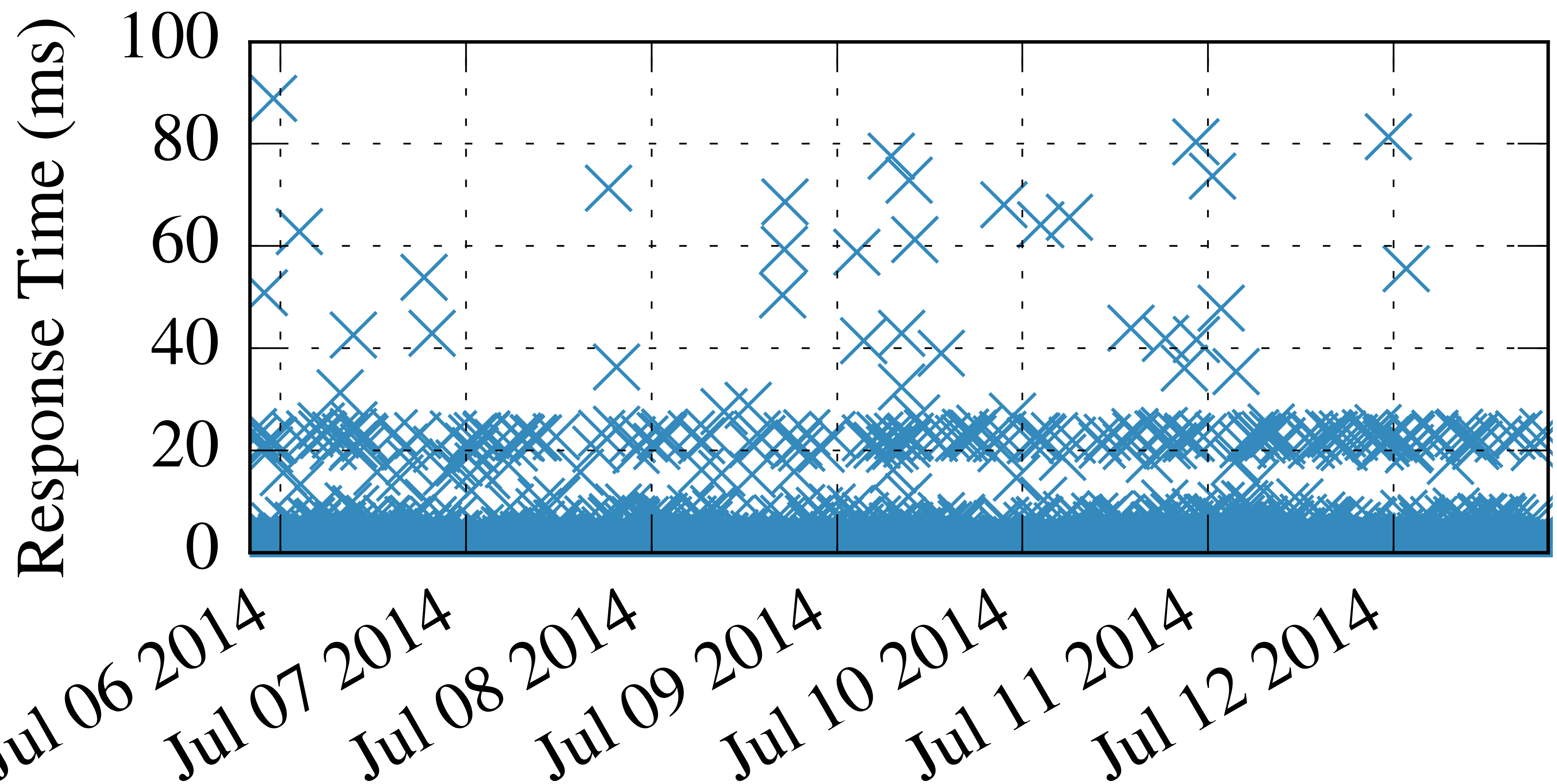
DNS Root Manipulation



DNS Root Manipulation



DNS Root Manipulation



- Impossibly low RTTs from China to B root (in Los Angeles)