Load balancing and traffic engineering: constructive interference

Nikhil Handigol, Ramesh Johari, Nick McKeown

Stanford University

April 12, 2011

Overview

- CDN operators ask:

  Given network conditions, what server will offer minimum latency to a given client?

- Network operators/ISPs ask:

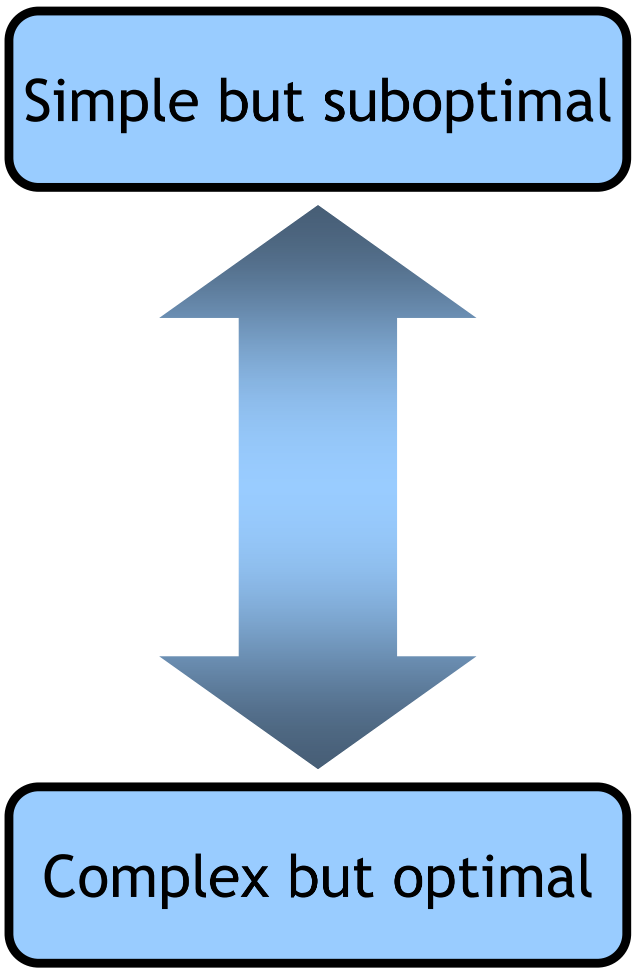
  Given traffic patterns, what paths should be used to route between sources and destinations?

  (Traffic engineering)

- We ask:

  Can these control loops "constructively interfere" with each other?

Some possible schemes



[Random+SP]

- CDN: random server selection
- TE: shortest path

[Disjoint]

- CDN: minimize server response time
- TE: minimize max link load

[Ideal]

- CDN + TE: Joint selection of optimal (path, server) pair

A first experiment

Our conjecture: “Ideal should be much better than the others.”

*Aster*x*: We implemented ideal load balancing *in the network* using OpenFlow.

The following demo illustrates this system:

http://yuba.stanford.edu/~nikhilh/Asterix-embed.mp4

Outline

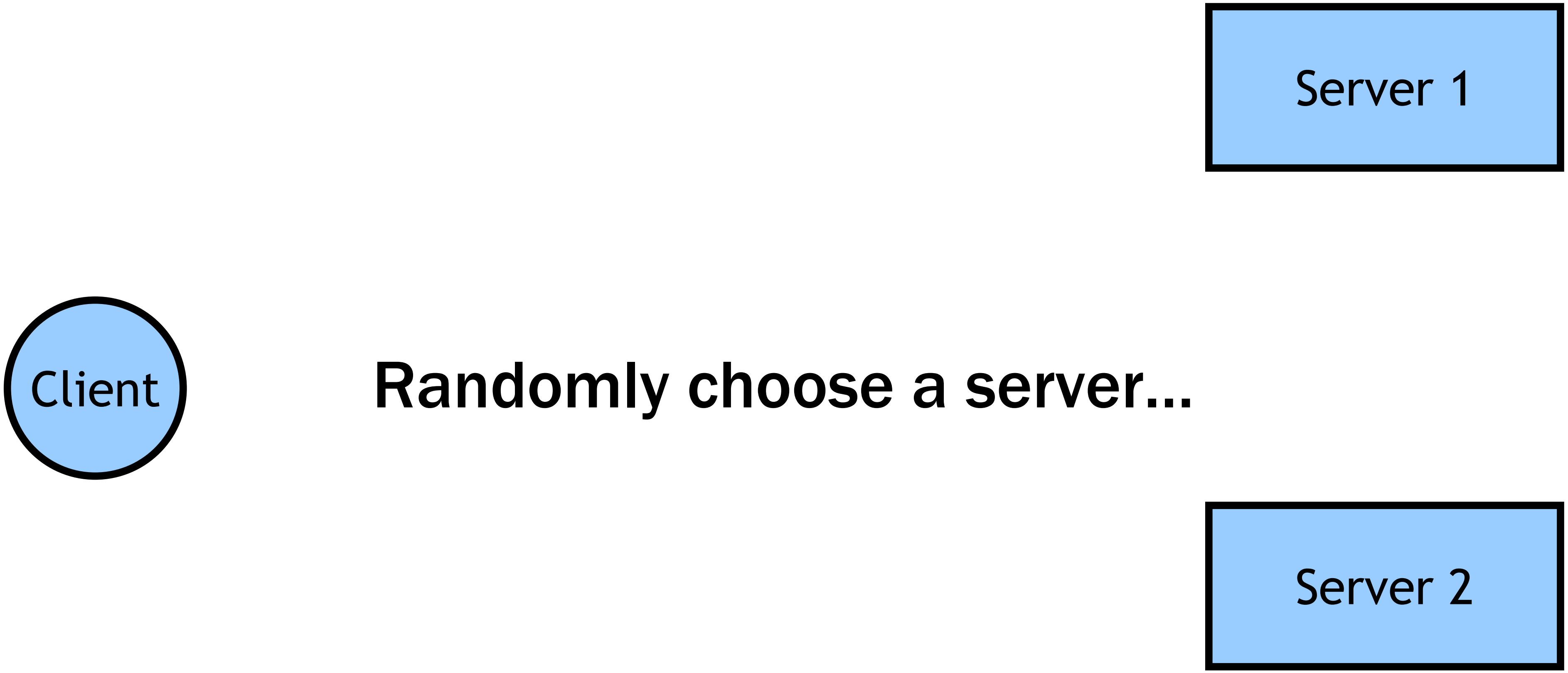
The experiment suggests Ideal is signficantly better than Random+SP.

Is this "generally" true?

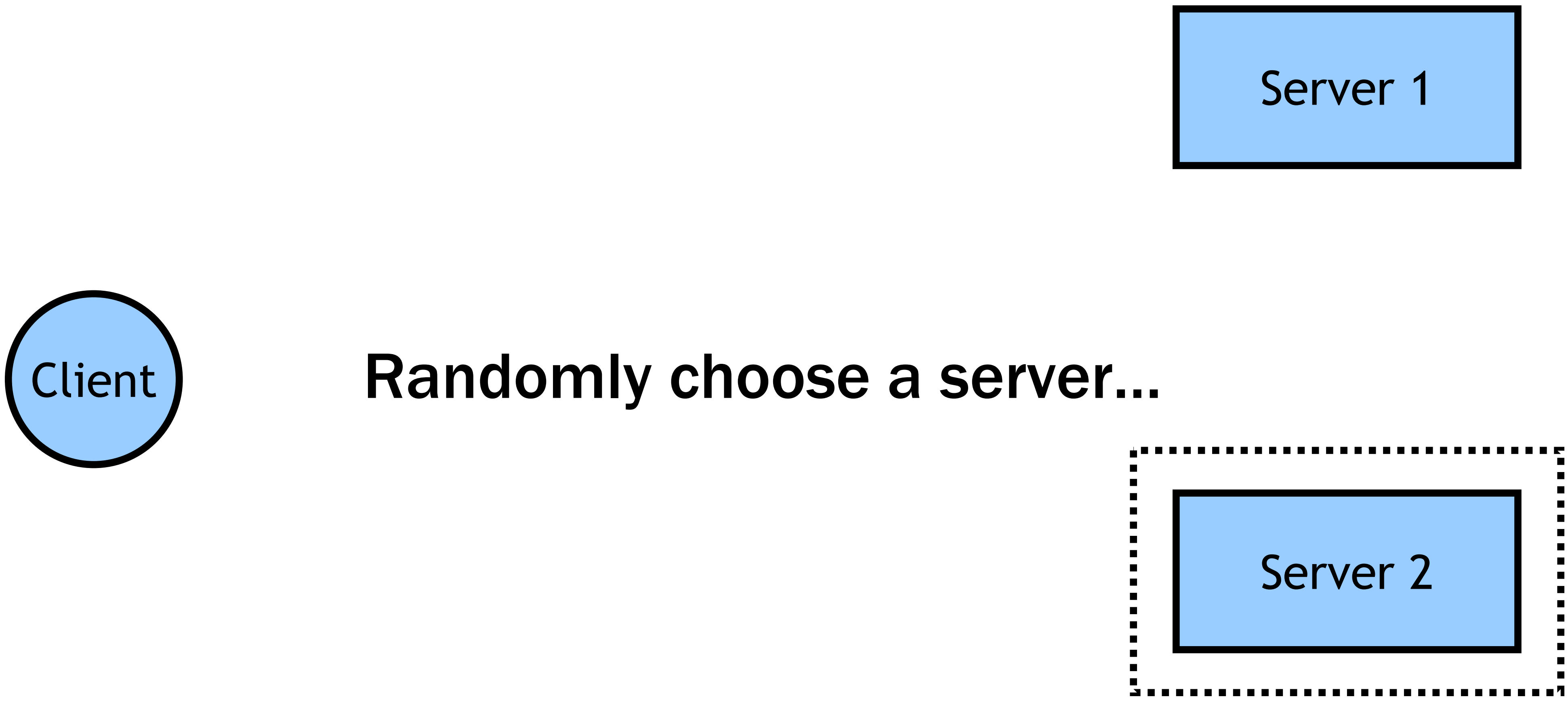
Is it also true when we compare to Disjoint?

We'll discuss these questions and close with open questions for the future.

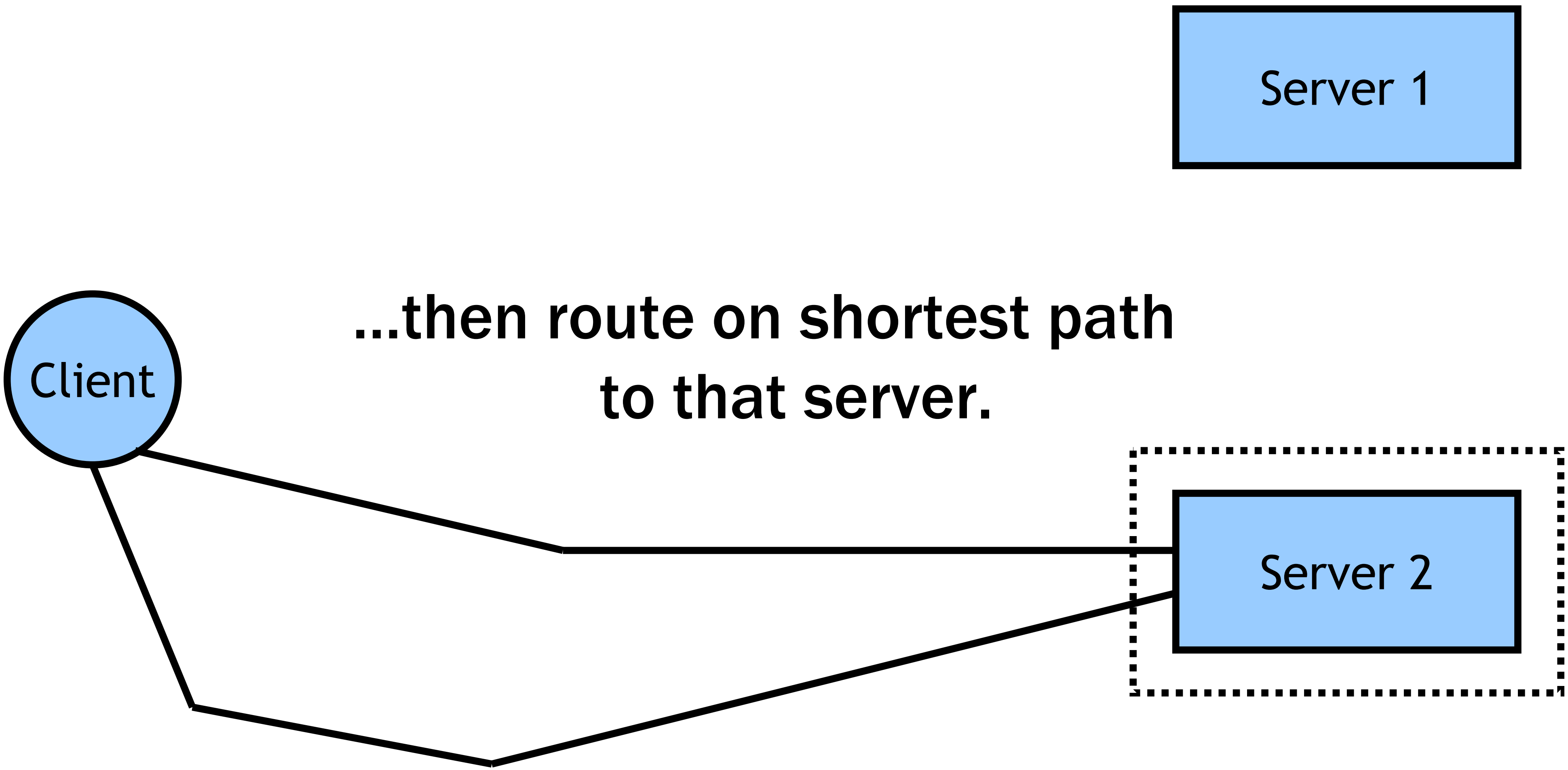
Random+SP



Random+SP

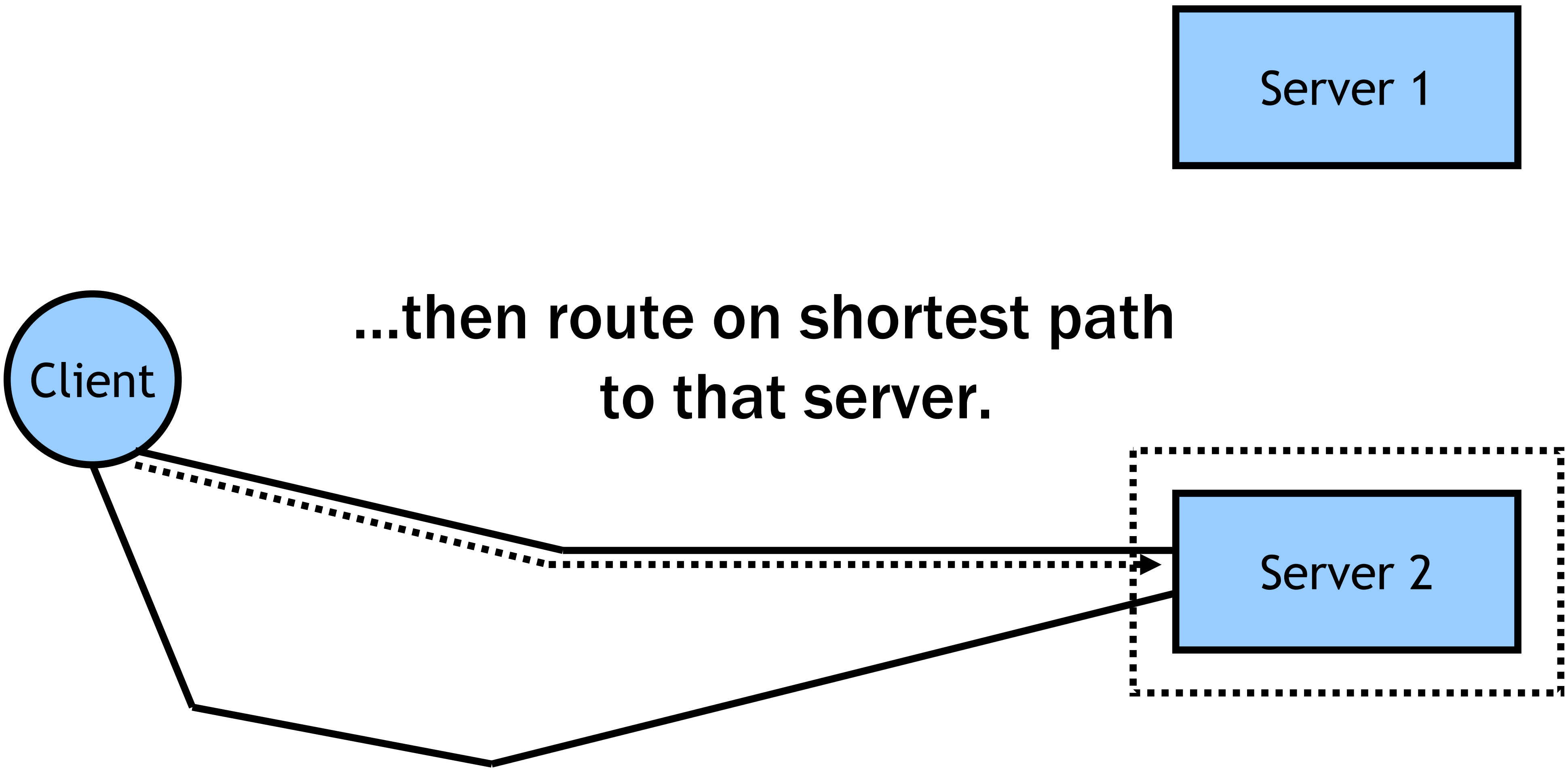


Random+SP



...then route on shortest path to that server.

Random+SP



Server 1
Client
...then route on shortest path
to that server.
Server 2

Disjoint

Divide server response time into:

Retrieve: Time to fetch first byte

Deliver: Time to complete streaming of request

Compute using moving averages.

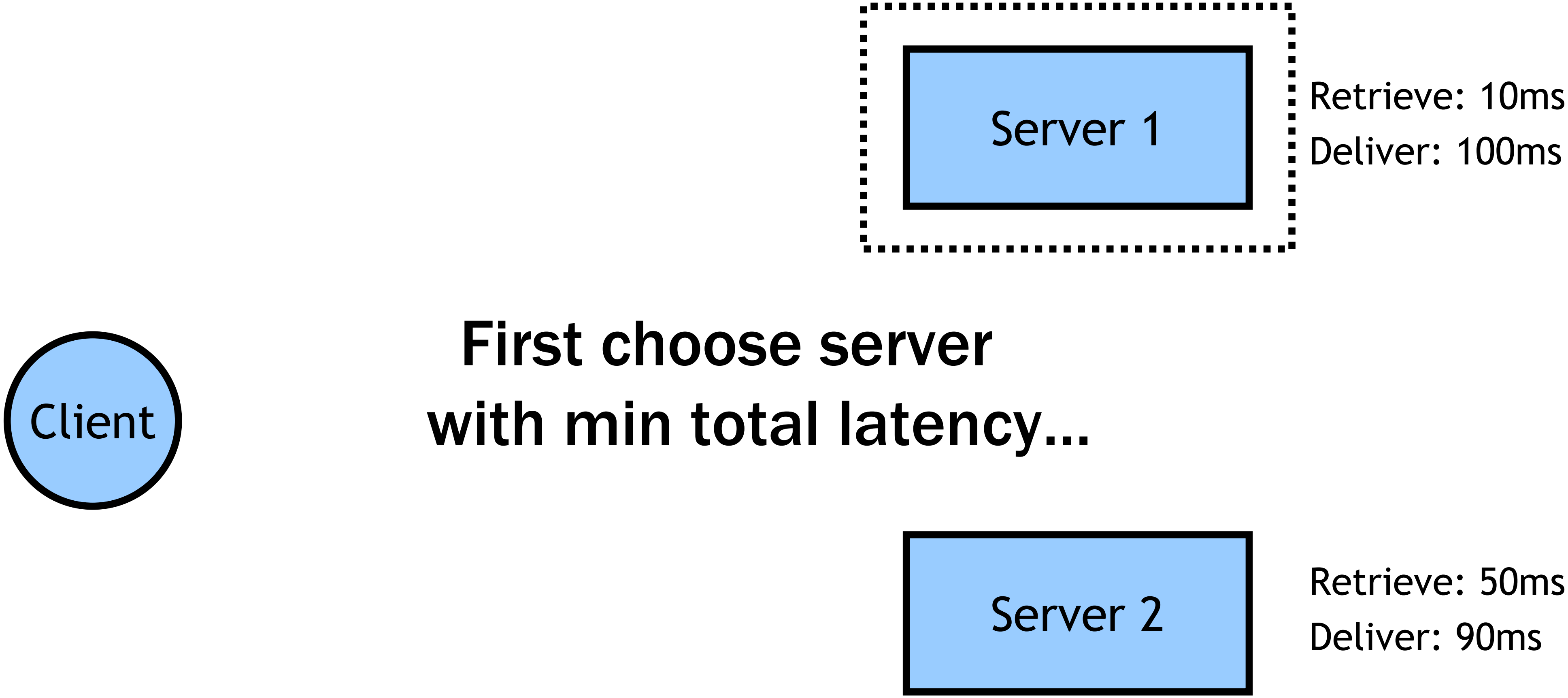
Server 1

Retrieve: 10ms
Deliver: 100ms

Server 2

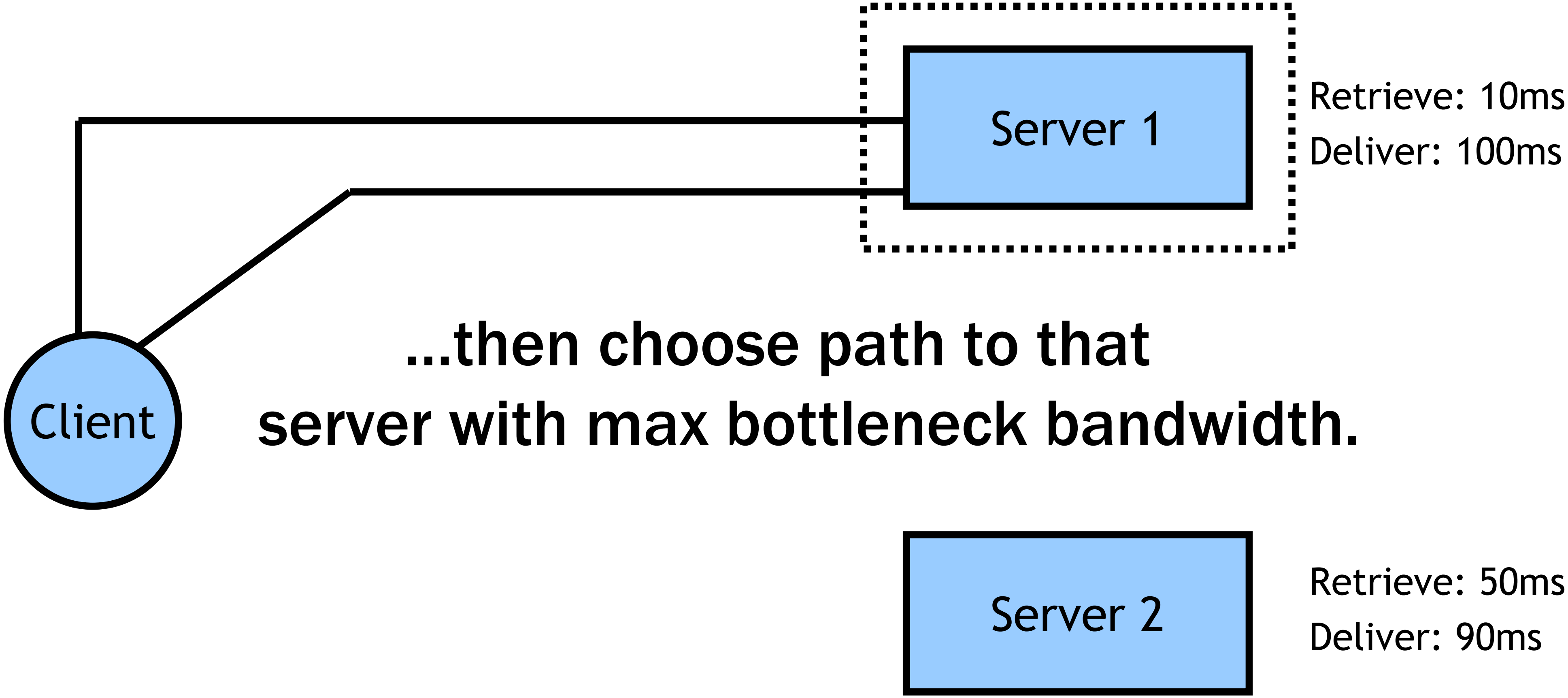
Retrieve: 50ms
Deliver: 90ms

Disjoint



First choose server with min total latency…

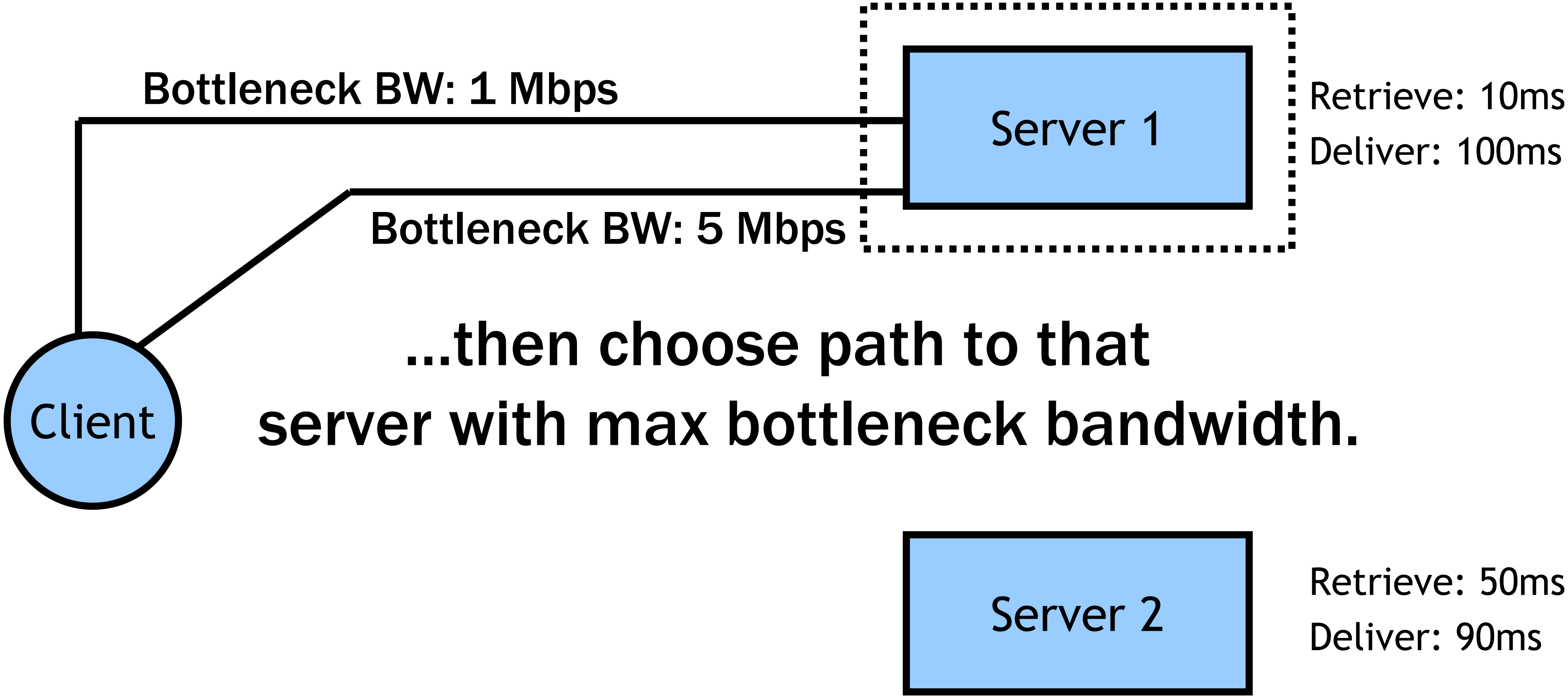
Disjoint



...then choose path to that server with max bottleneck bandwidth.

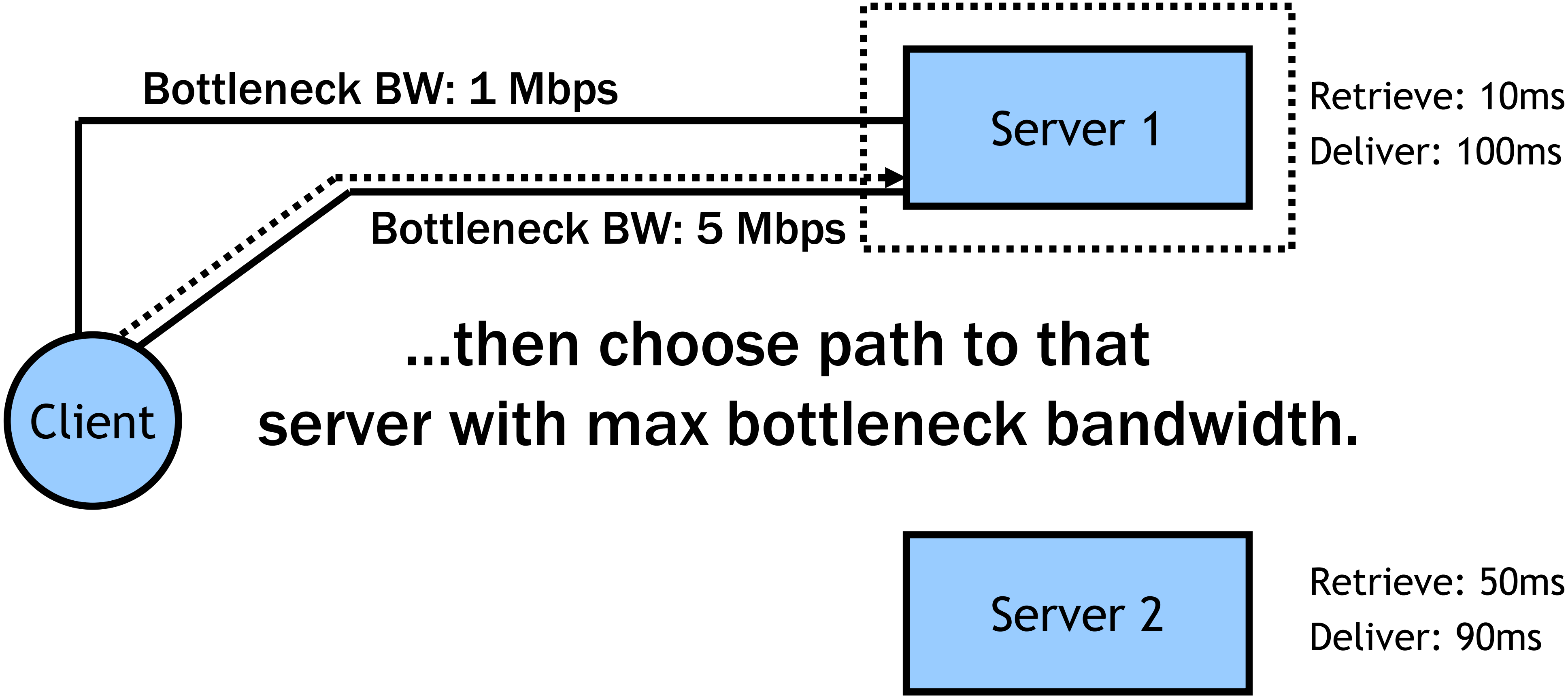
(This is a form of traffic engineering.)

Disjoint



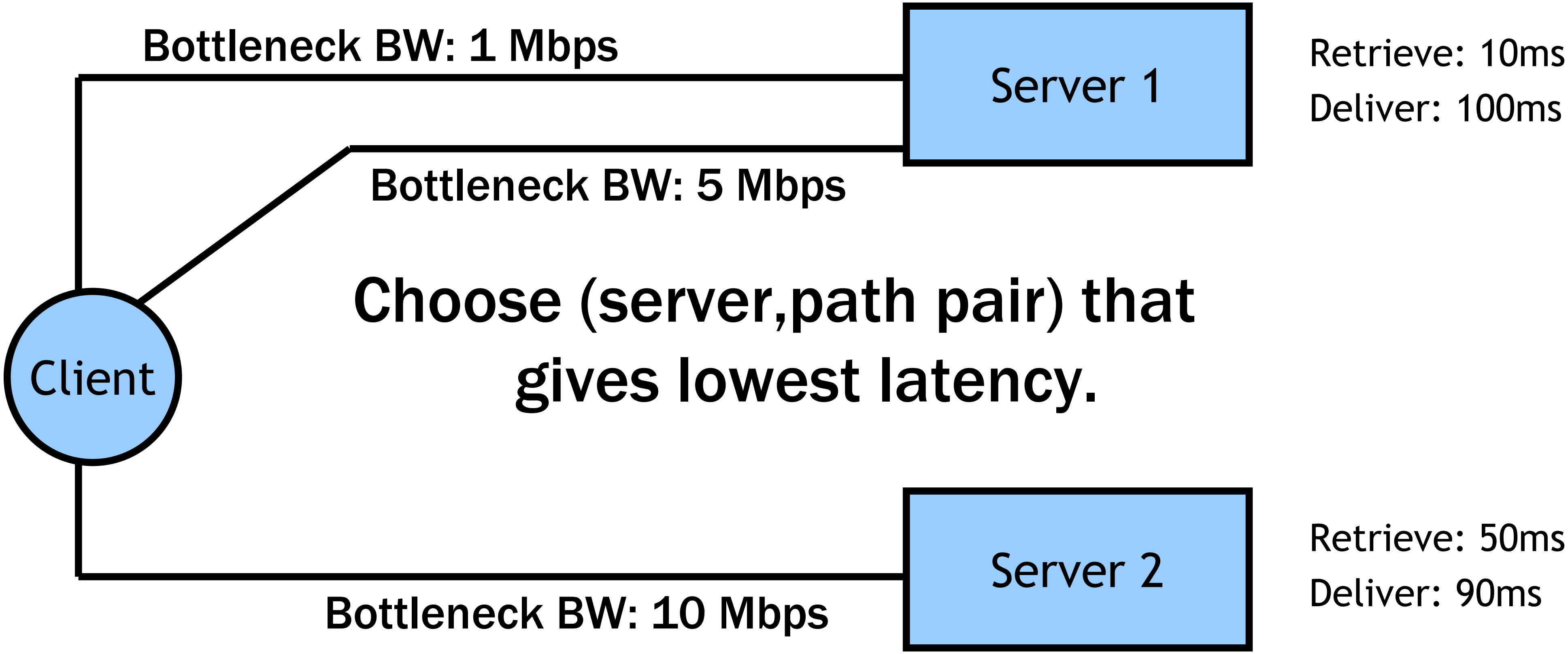
...then choose path to that server with max bottleneck bandwidth.

Disjoint



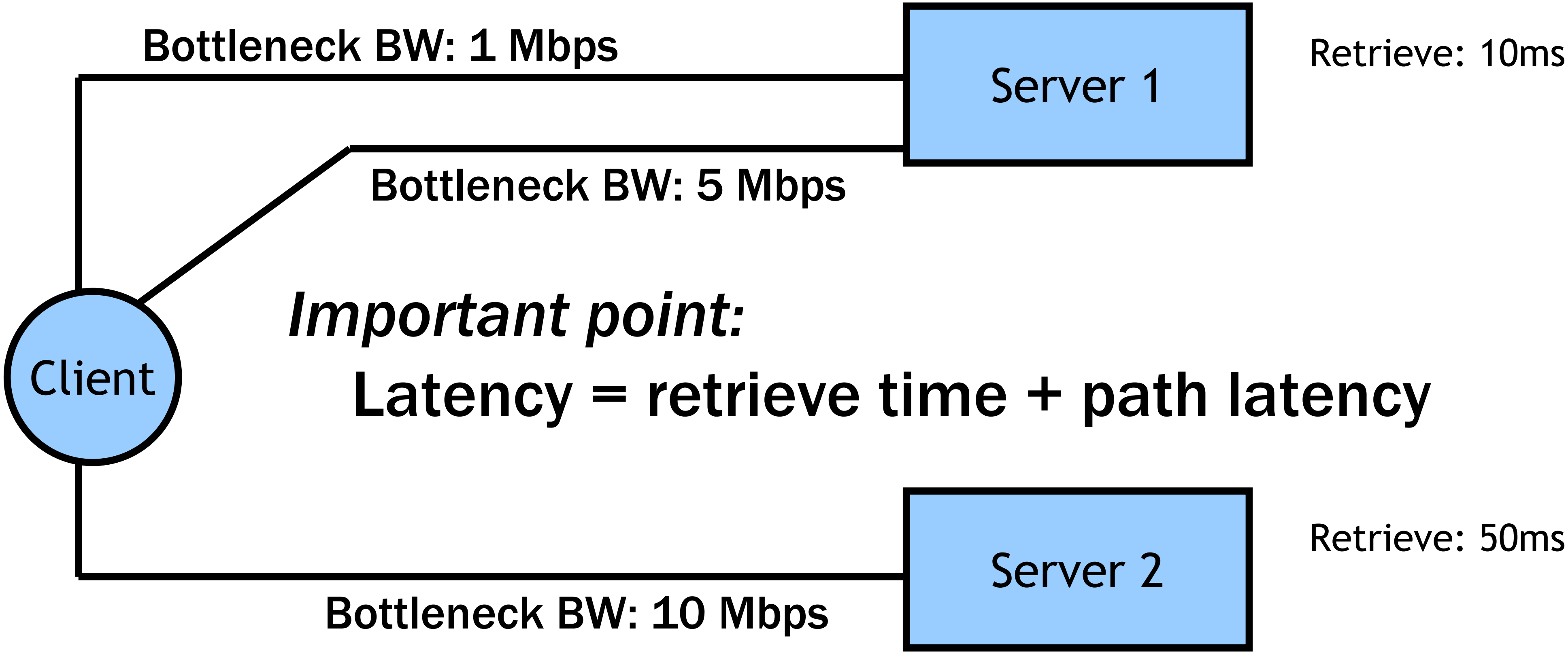
...then choose path to that server with max bottleneck bandwidth.

Ideal



Choose (server,path pair) that gives lowest latency.

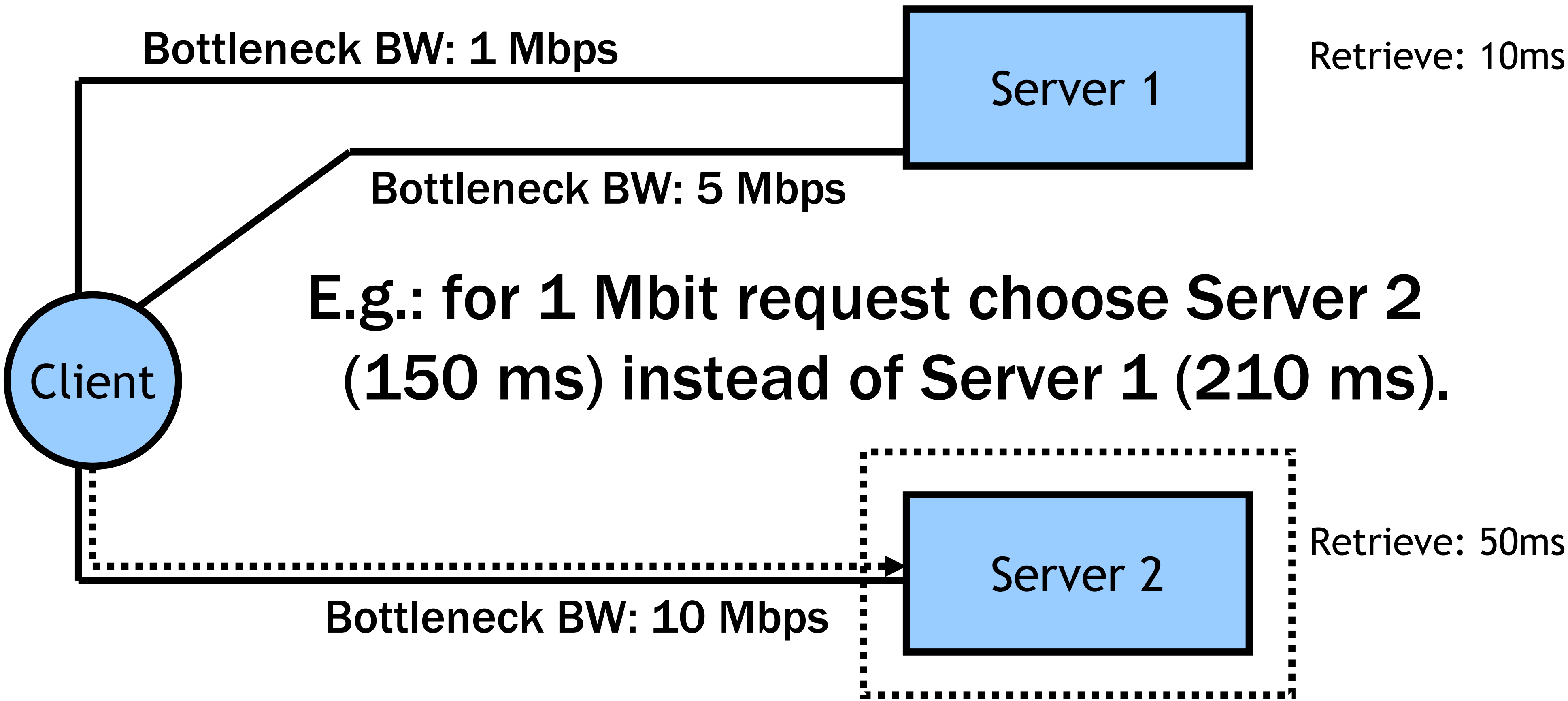
Ideal



Important point:

Latency = retrieve time + path latency

Ideal

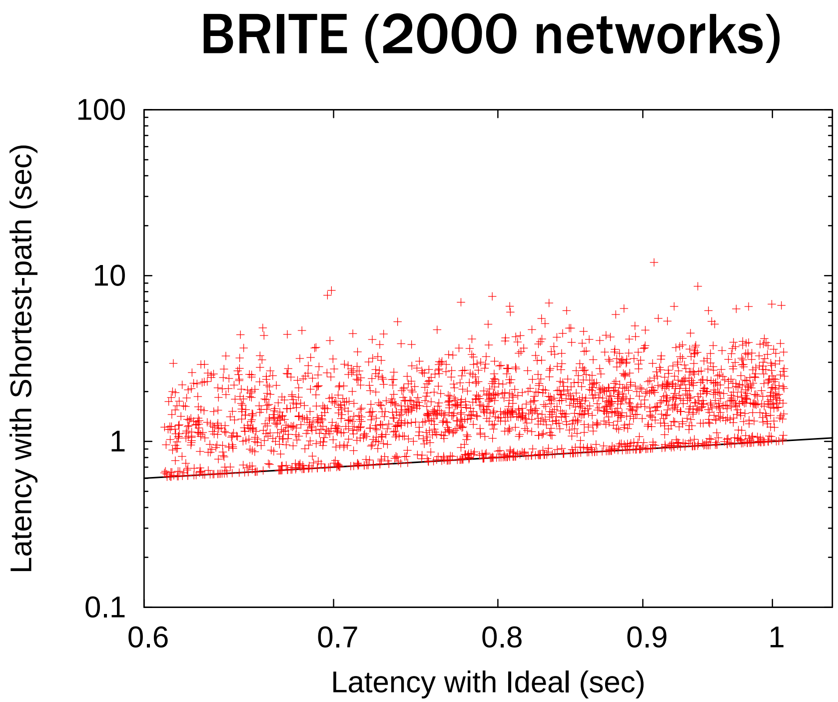


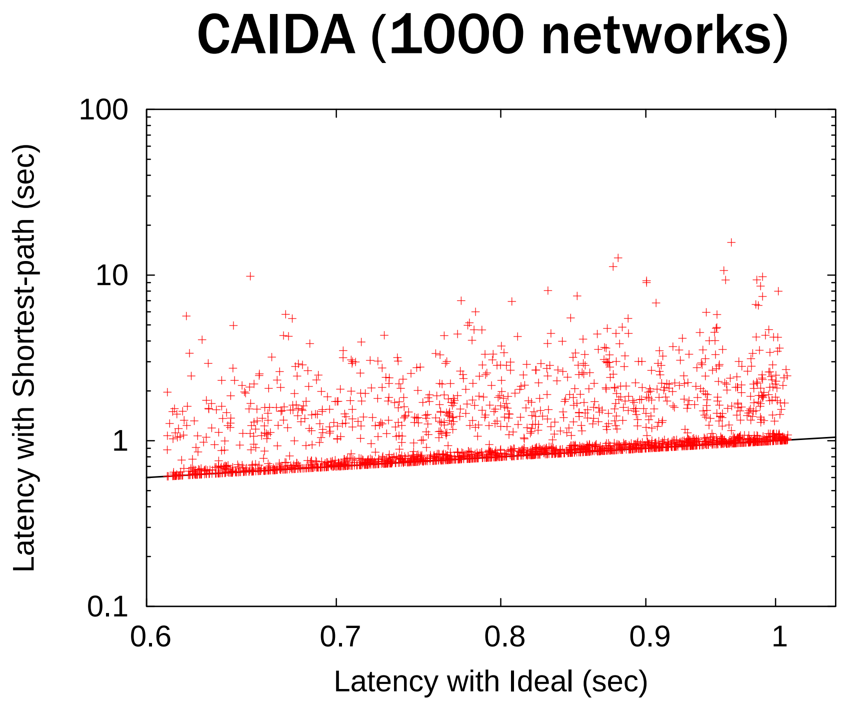
E.g.: for 1 Mbit request choose Server 2 (150 ms) instead of Server 1 (210 ms).

Testing methodology

- We use an *emulation environment* designed in-house: *MiniNet-RT*
- Two types of networks:
  1) BRITE (randomly generated) 40-node topologies
     (meant to "simulate" AS topologies)
  2) CAIDA 20-50 node topologies
     (actual intra-AS router-level topologies)
- All links fixed at 10Mbps
- Randomly place 1-3 clients, 5-20 servers
- 10 requests/sec, 1MB/request

Random+SP vs. Ideal

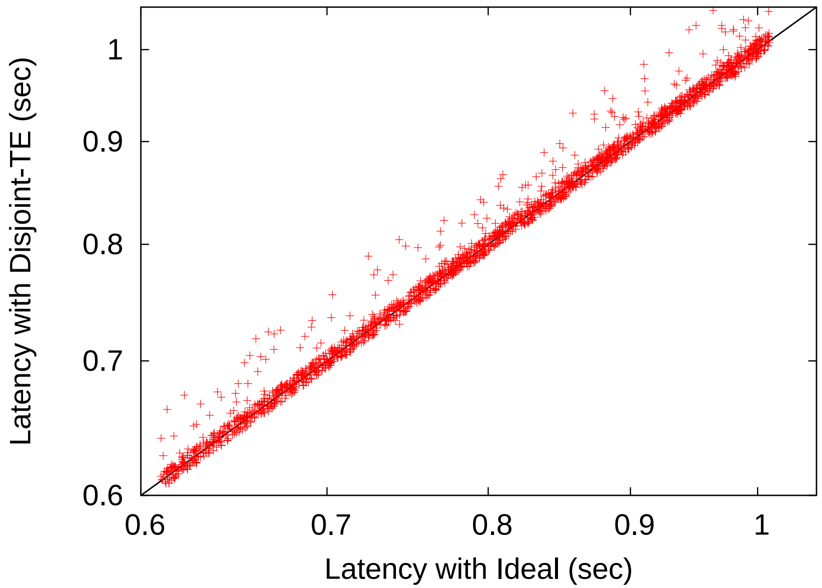




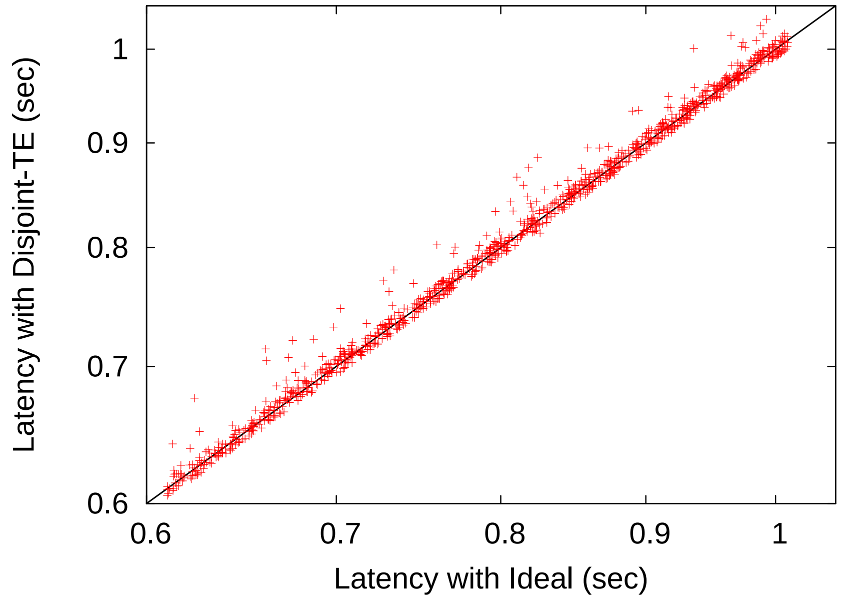
Random+SP achieves 50% of performance of Ideal in 50% (BRITE) to 85% (CAIDA) of topologies

Disjoint vs. Ideal

BRITE (2000 networks)



CAIDA (1000 networks)



Disjoint achieves 98% of performance of Ideal in over 90% of BRITE and CAIDA topologies!

Main observations

- Random+SP is bad, but not as bad as we may have initially thought (especially on *real* topologies).
  Question: Are networks designed to make this so?

- Disjoint performs almost as well as Ideal, despite *decoupling* of traffic engineering and server selection.
  Question: Why?

Disjoint vs. Ideal

Recall that in disjoint:

- Servers chosen based on minimum latency = *retrieve* time + *deliver* time.
- Paths chosen based on maximum bottleneck bandwidth.

Both push the system in the same direction:
servers with minimum latency eventually prove to be those with higher bottleneck bandwidth.

(We observe this empirically and justify it theoretically.)

Concluding questions

We want to know what *you* observe.

In real networks, performance results from the interaction of *design* and *operation.*

If you do observe adverse interactions of TE and server selection, is it poor operation or poor design?

If not, is it intelligent operation or planned design?