High Speed Networks Need Proactive Congestion Control

Using Programmable Forwarding Planes!

vanya Jose, Steve Ibanez, Lisa Yan, Nick McKeown, Sachin Katti

Stanford University

Mohammad Alizadeh

MIT

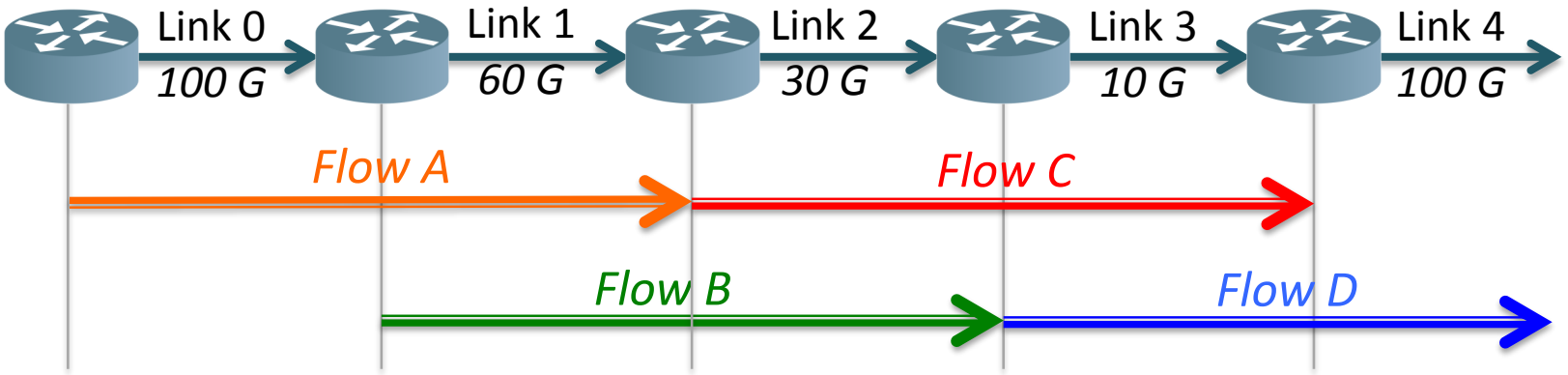
George Varghese

Microsoft Research

Outline

- At 100G speeds, we'll need much faster congestion control schemes
- Letting networks switches directly compute rates is a fast and scalable scheme
- We can realize such a scheme in 100G networks using programmable forwarding planes (stateful data planes)

The Congestion Control Problem

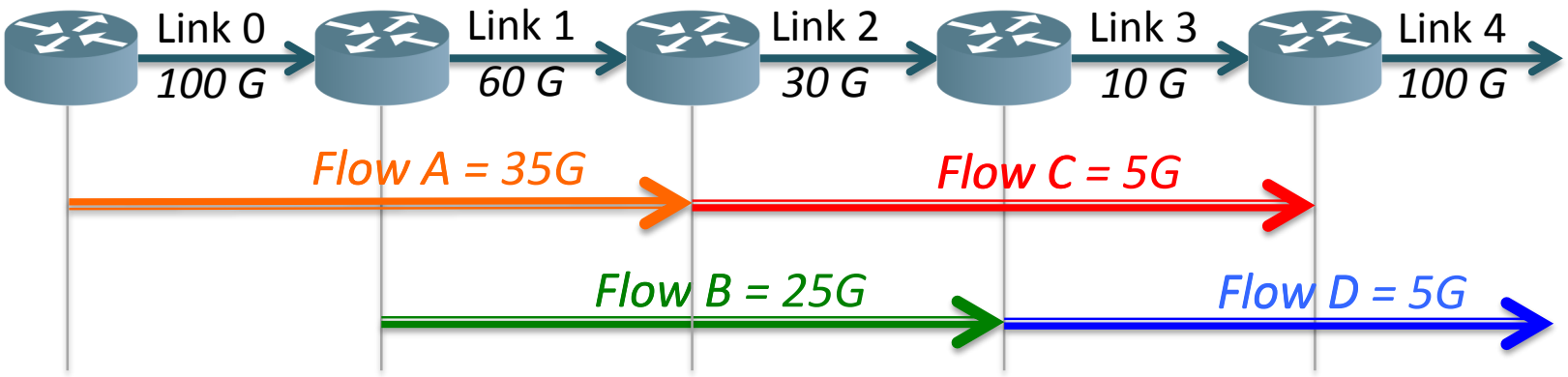


Ask an oracle.

	Link 0
Flow A	√
Flow B	
Flow C	
Flow D	

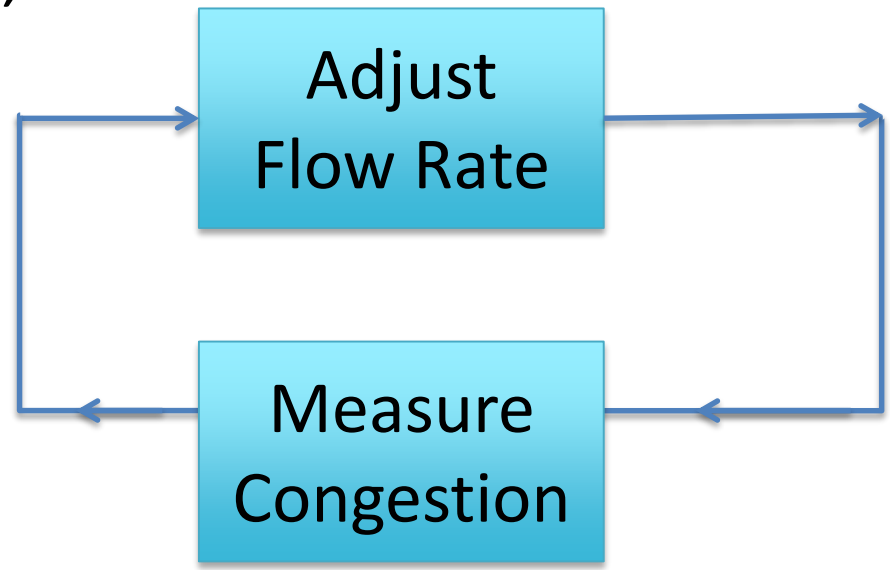
Link	Capacity
0	100
1	60
2	30
3	10
4	100

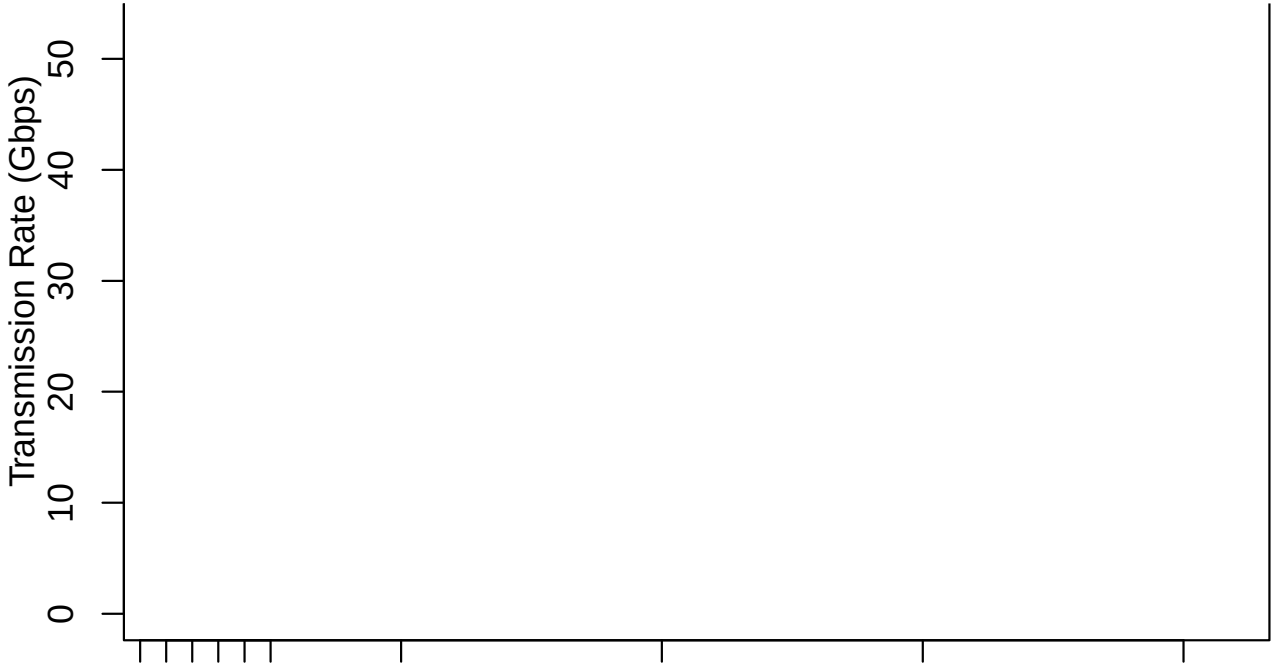
Flow	Rate
Flow A	35
Flow B	25
Flow C	5
Flow D	5

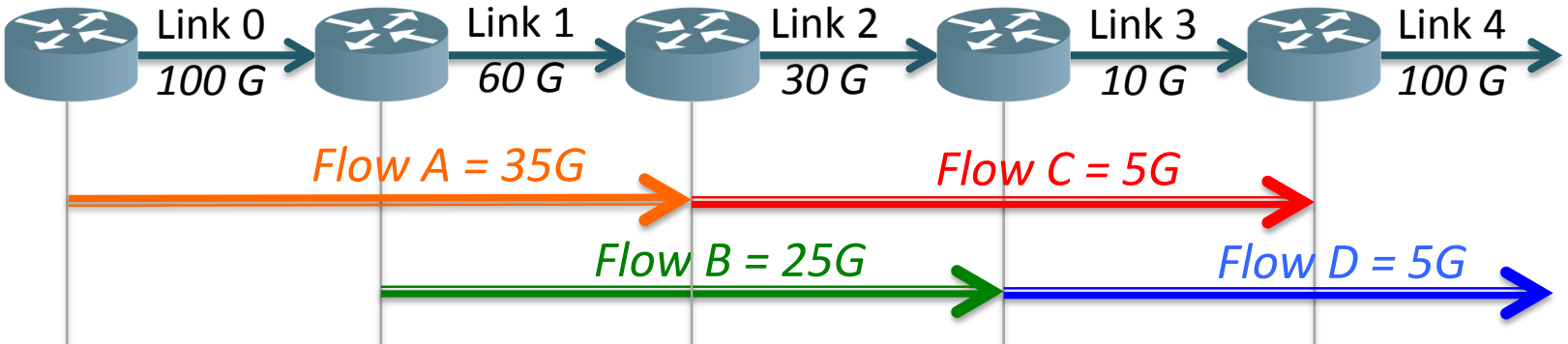


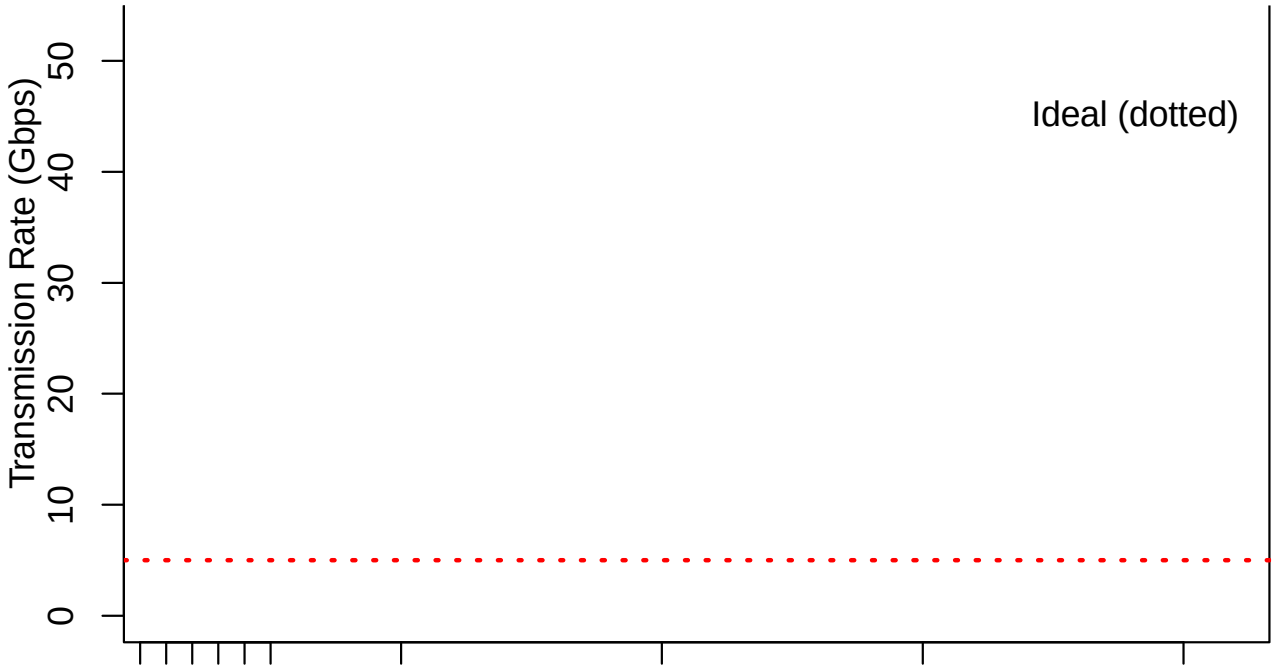
Traditional congestion control

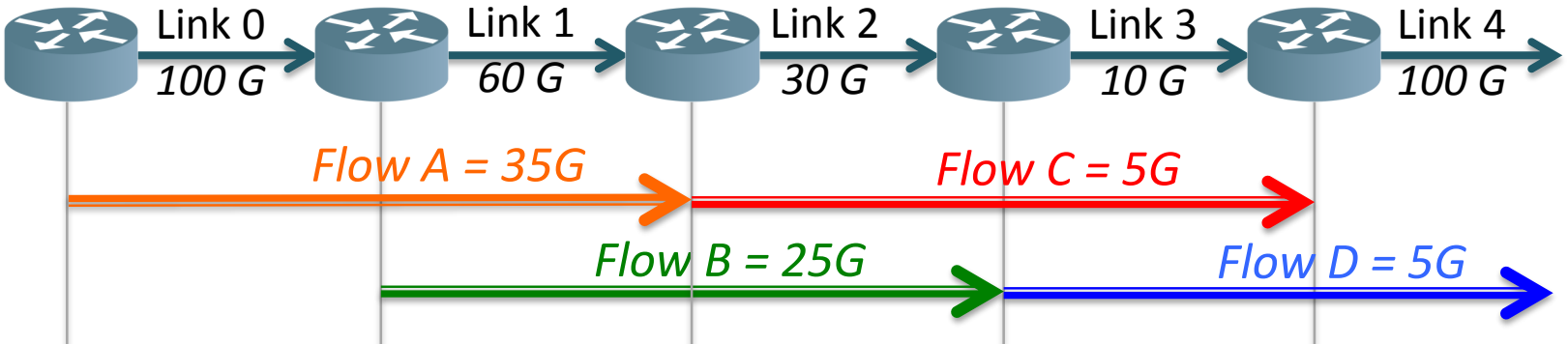
- No explicit information about traffic matrix
- Measure congestion signals, then react by adjusting rate after measurement delay
- Gradual, can't jump to right rates, know direction
- "Reactive Algorithms"

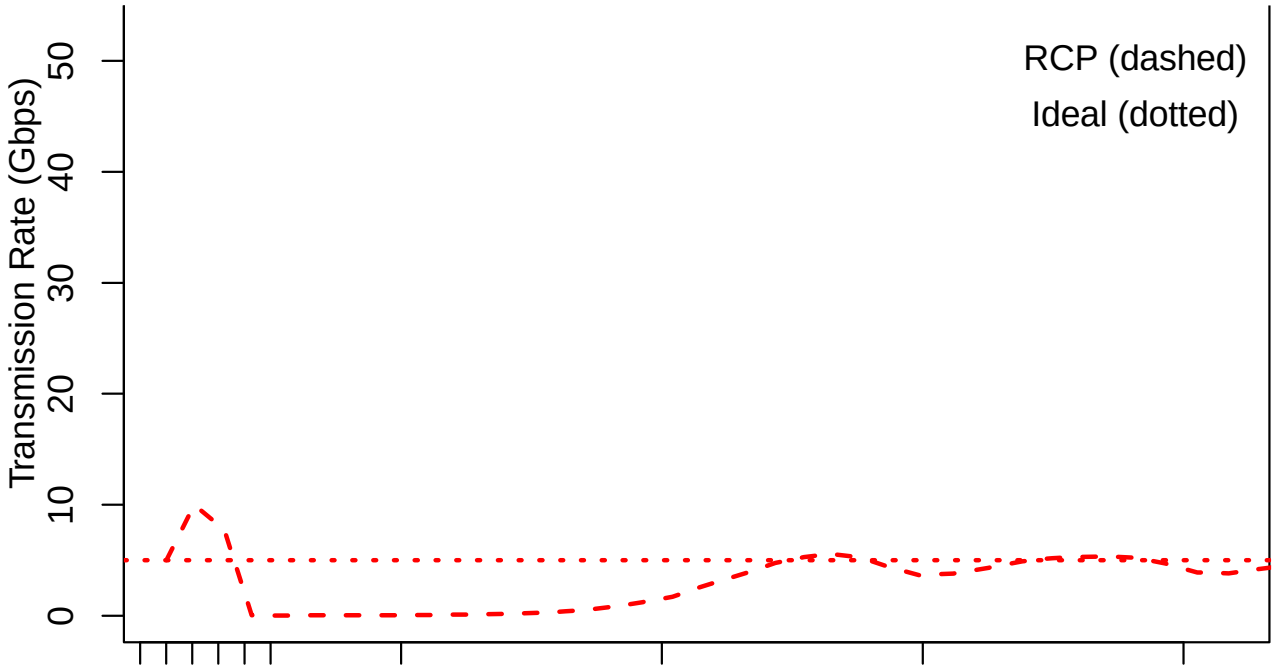


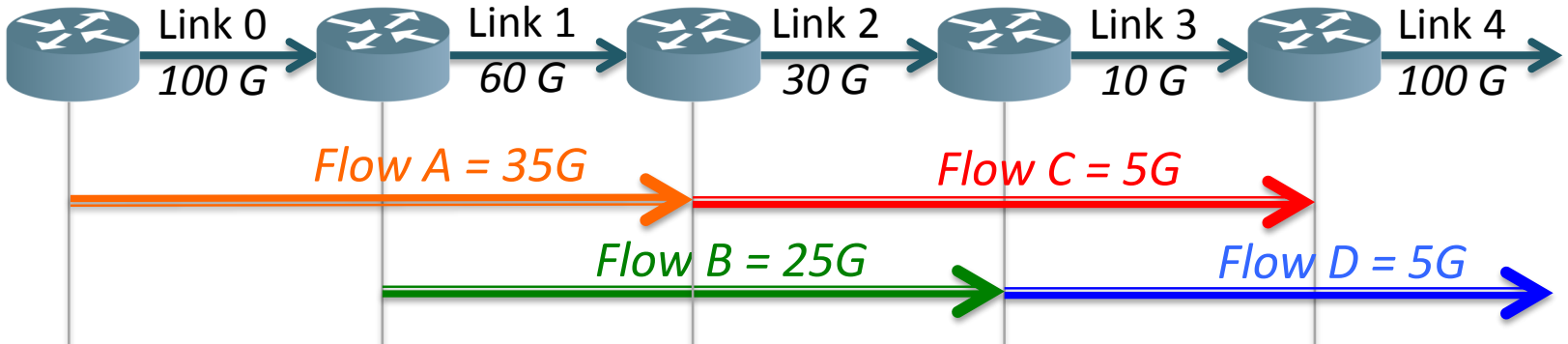

Transmission Rate (Gbps)
0
10
20
30
40
50

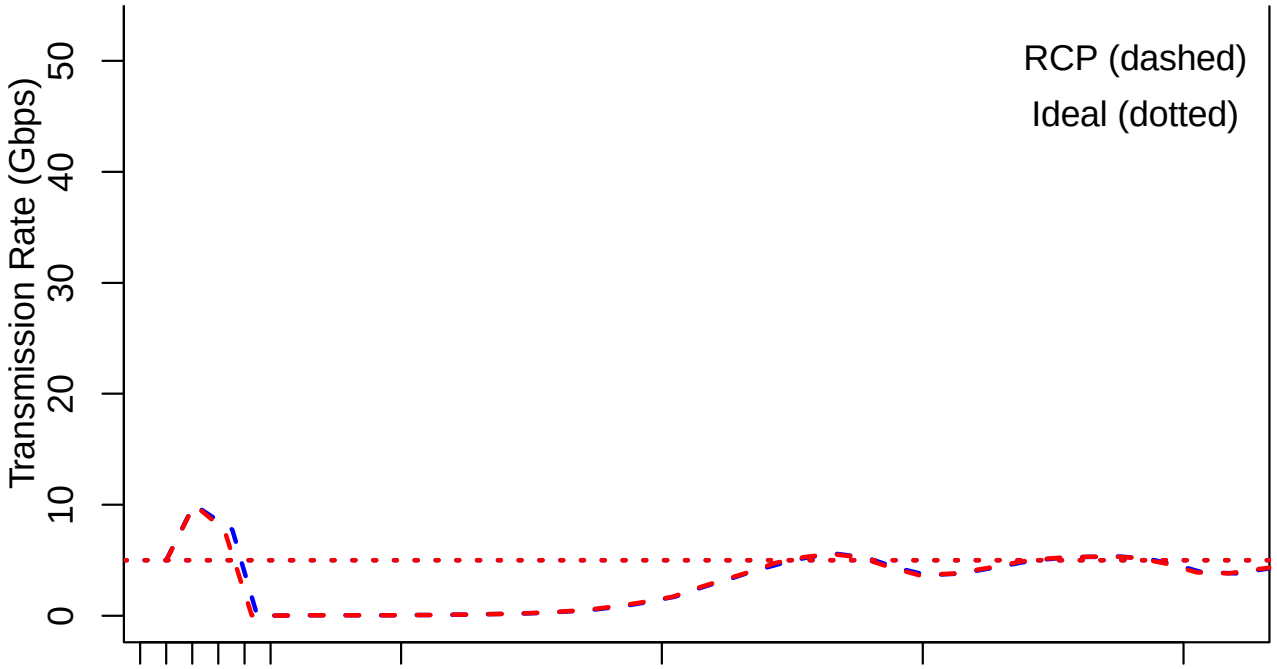

Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G

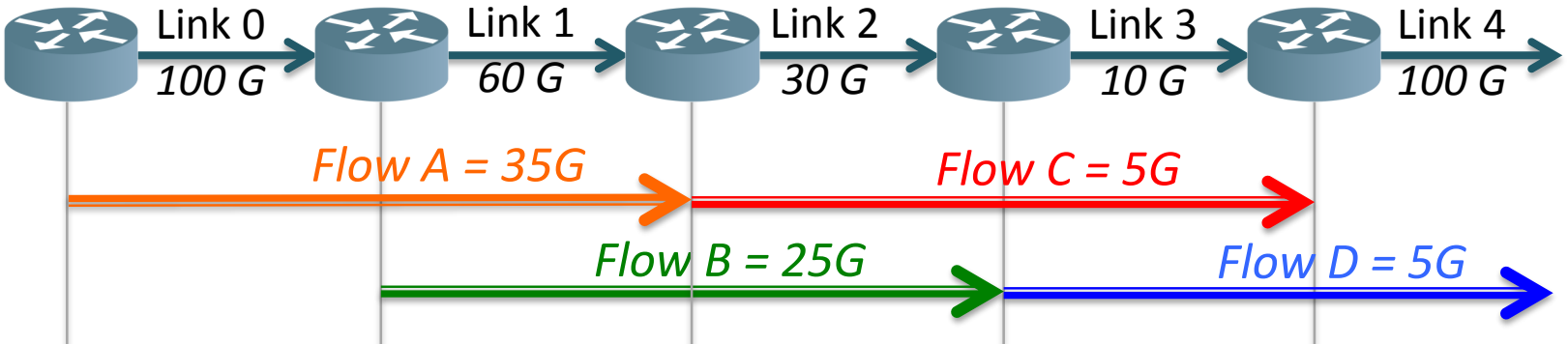

Transmission Rate (Gbps)
0
10
20
30
40
50
Ideal (dotted)

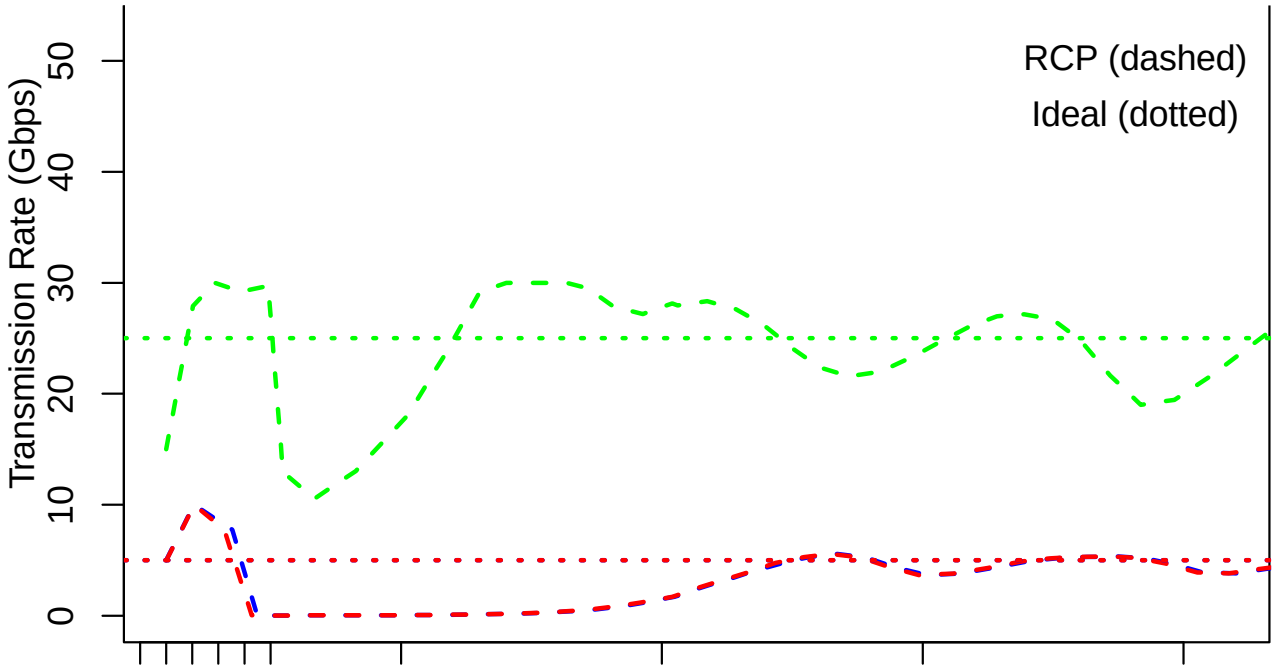

Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G

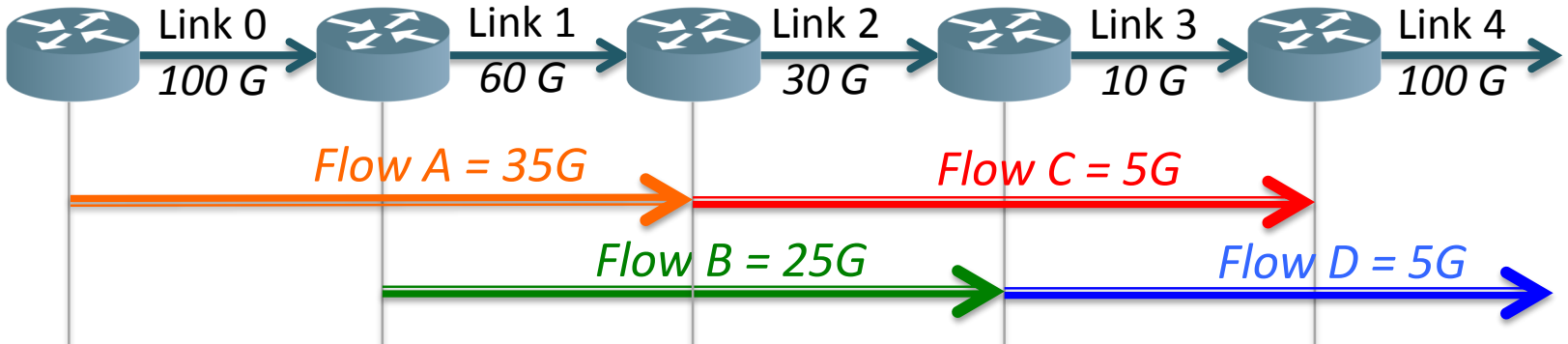

Transmission Rate (Gbps)
0
10
20
30
40
50
RCP (dashed)
Ideal (dotted)

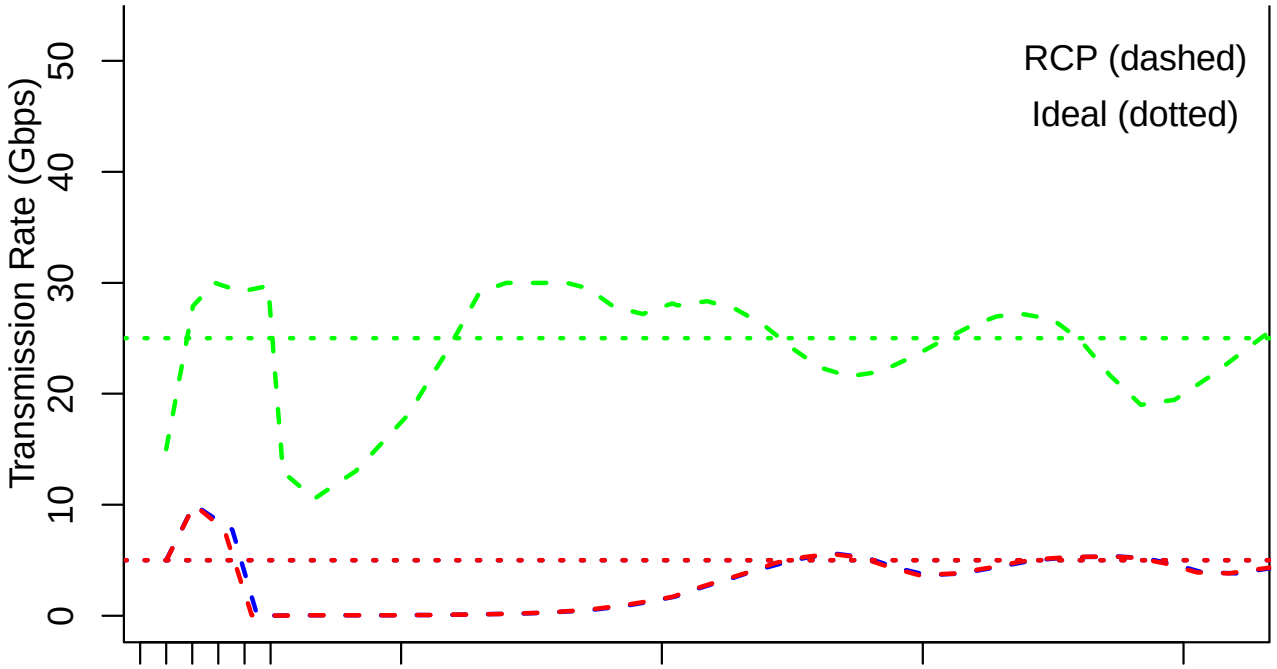

Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G

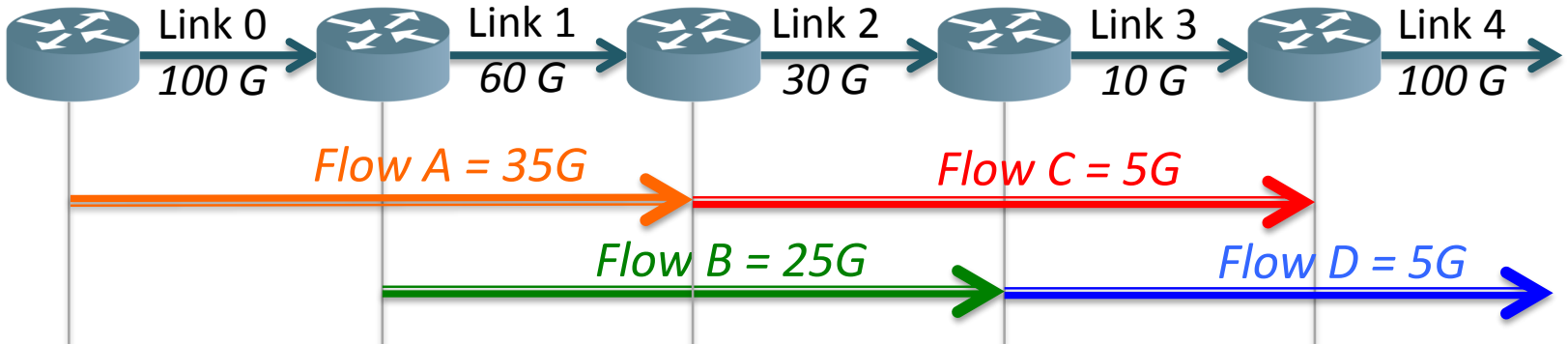

RCP (dashed)
Ideal (dotted)
Transmission Rate (Gbps)
0
10
20
30
40
50

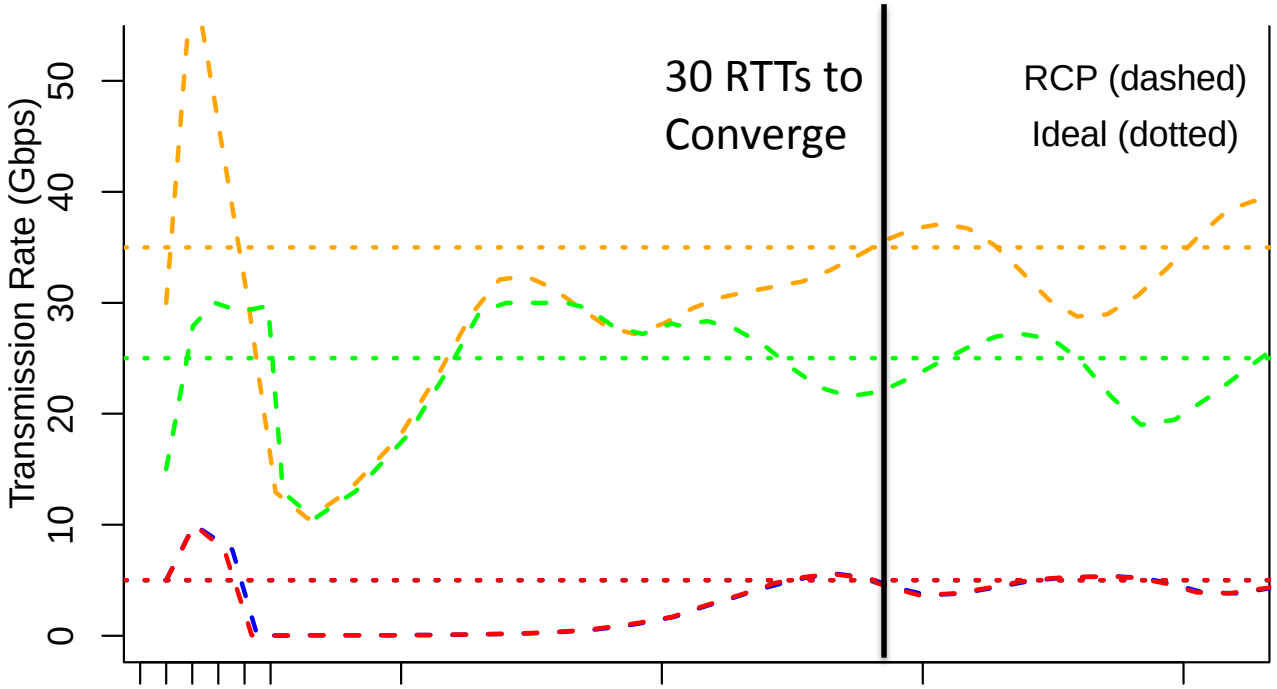

Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G

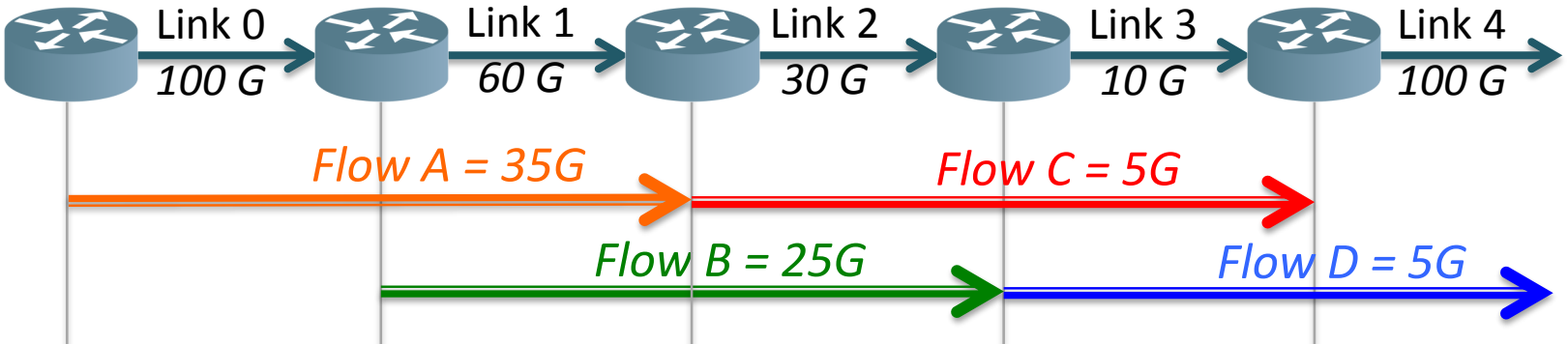

RCP (dashed)
Ideal (dotted)
Transmission Rate (Gbps)
0
10
20
30
40
50


Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G


RCP (dashed)
Ideal (dotted)
Transmission Rate (Gbps)
0
10
20
30
40
50

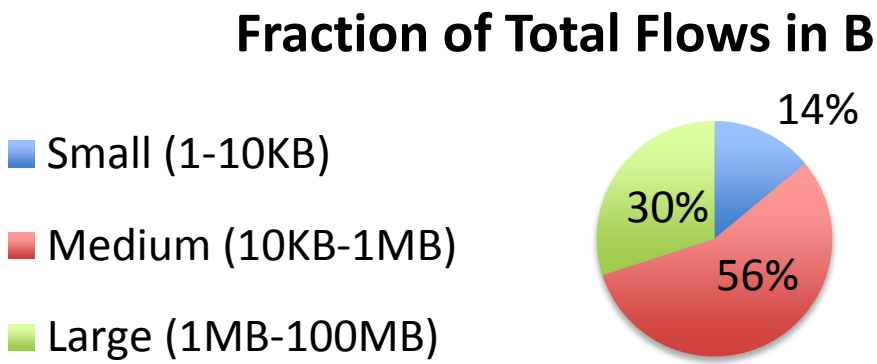

Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G


Transmission Rate (Gbps)
0
10
20
30
40
50
30 RTTs to Converge
RCP (dashed)
Ideal (dotted)

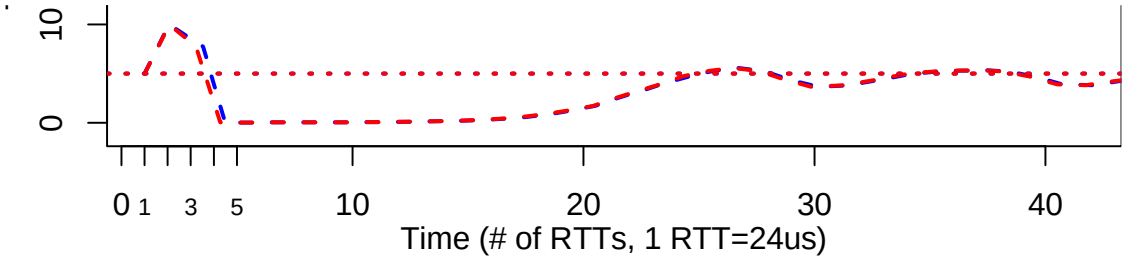

Link 0
100 G
Link 1
60 G
Link 2
30 G
Link 3
10 G
Link 4
100 G
Flow A = 35G
Flow C = 5G
Flow B = 25G
Flow D = 5G

Reconverge Time is Low for 100G

At 100G, a typical flow in a search workload is < 7 RTTs long.



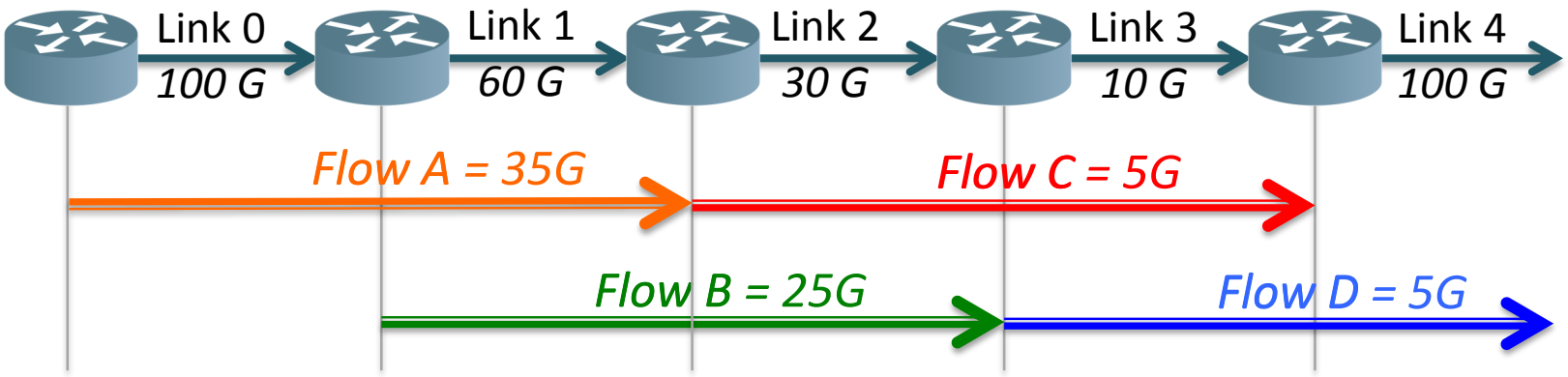
1MB / 100 Gb/s = 80 µs



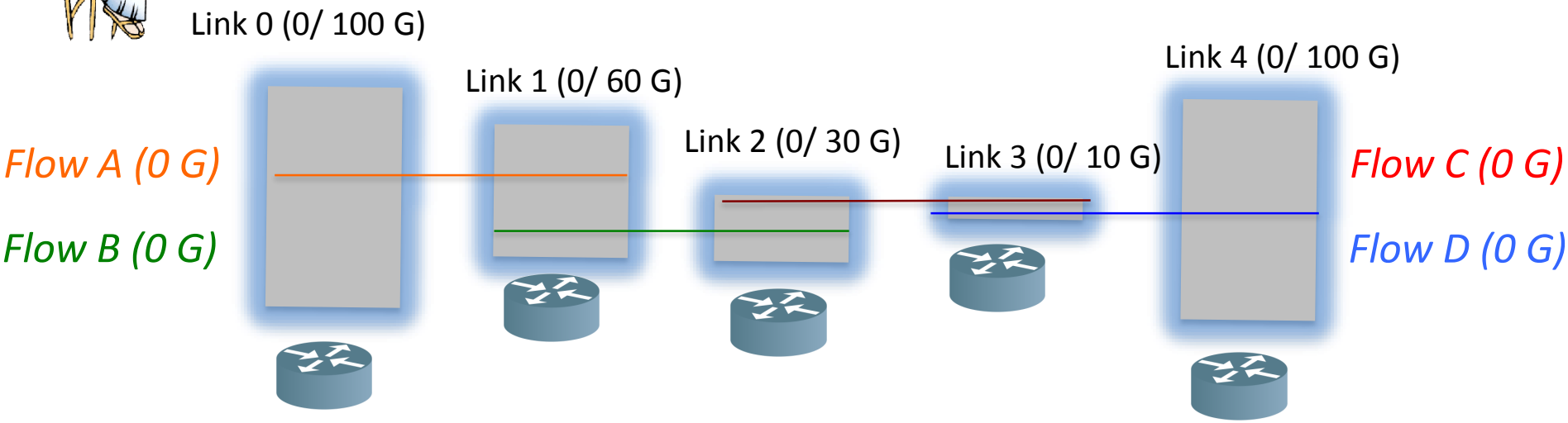
Reactive algorithms trade off *explicit flow information* for *long convergence times*

- Can we use explicit flow information
- and get shorter convergence times?

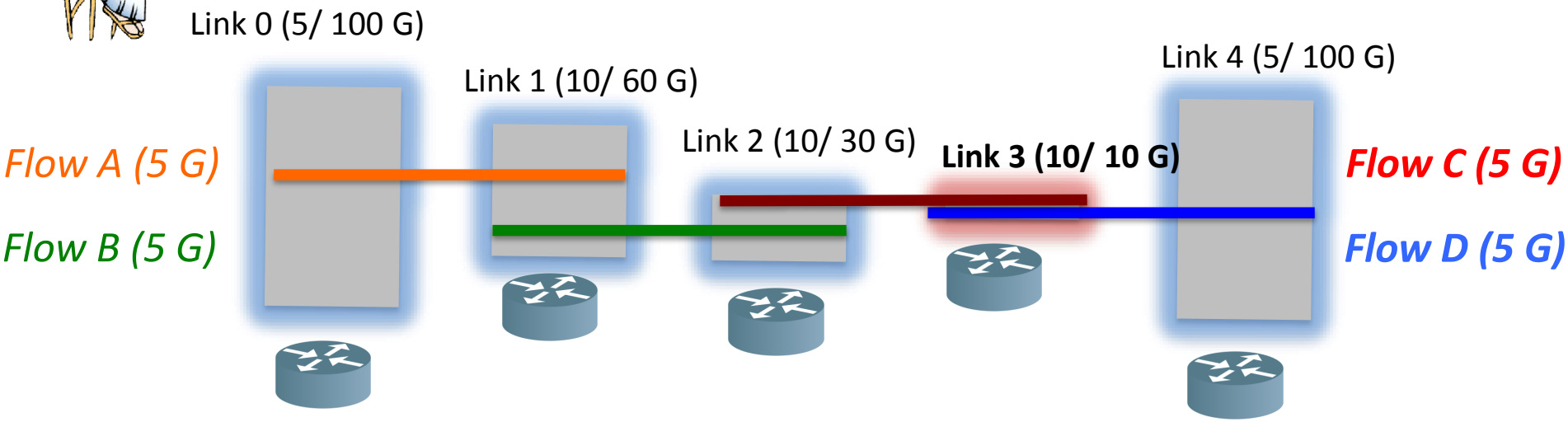
Back to the oracle, how did she use traffic matrix to compute rates?



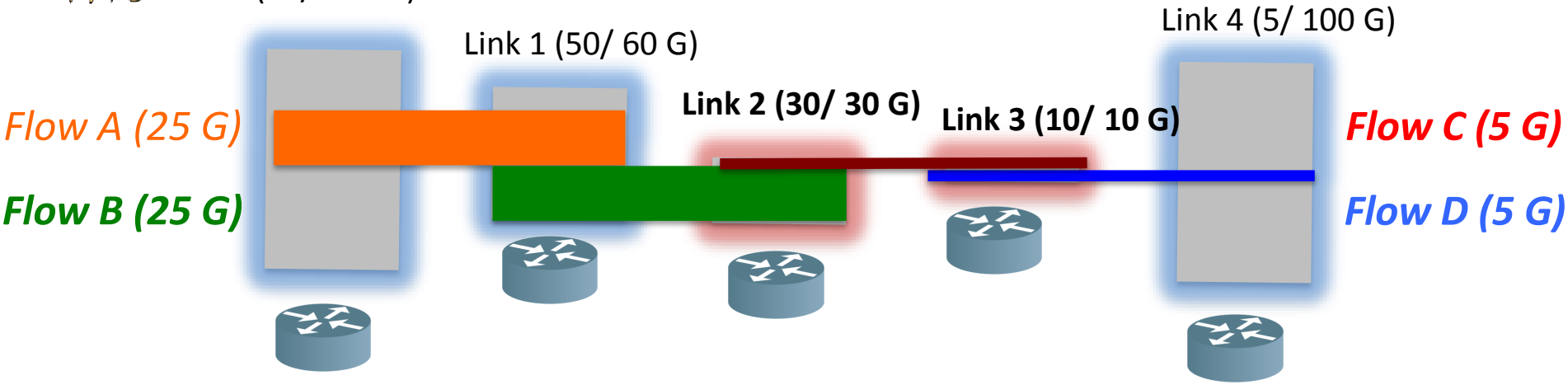
Waterfilling Algorithm



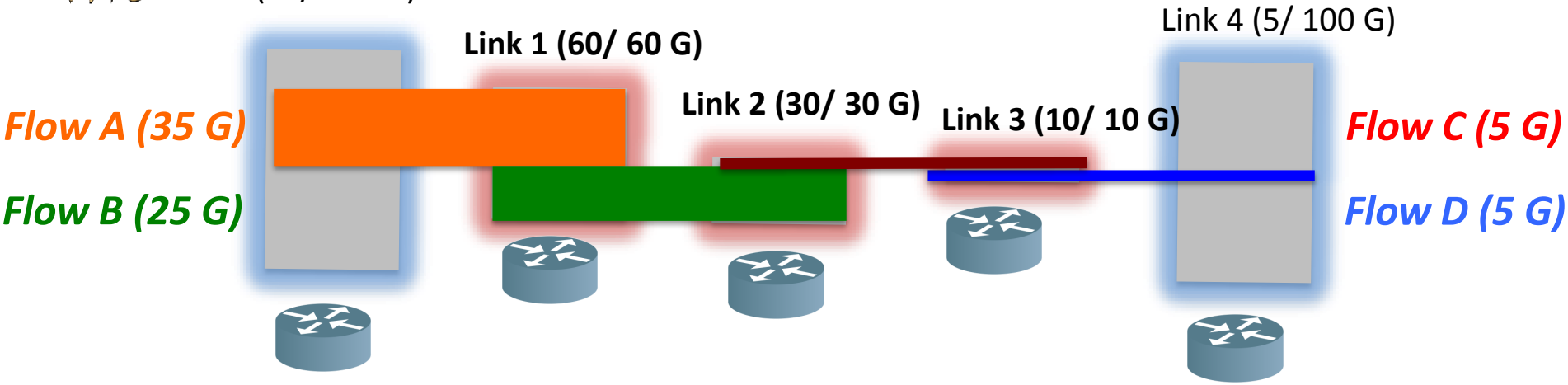
Waterfilling- 10 G link is fully used



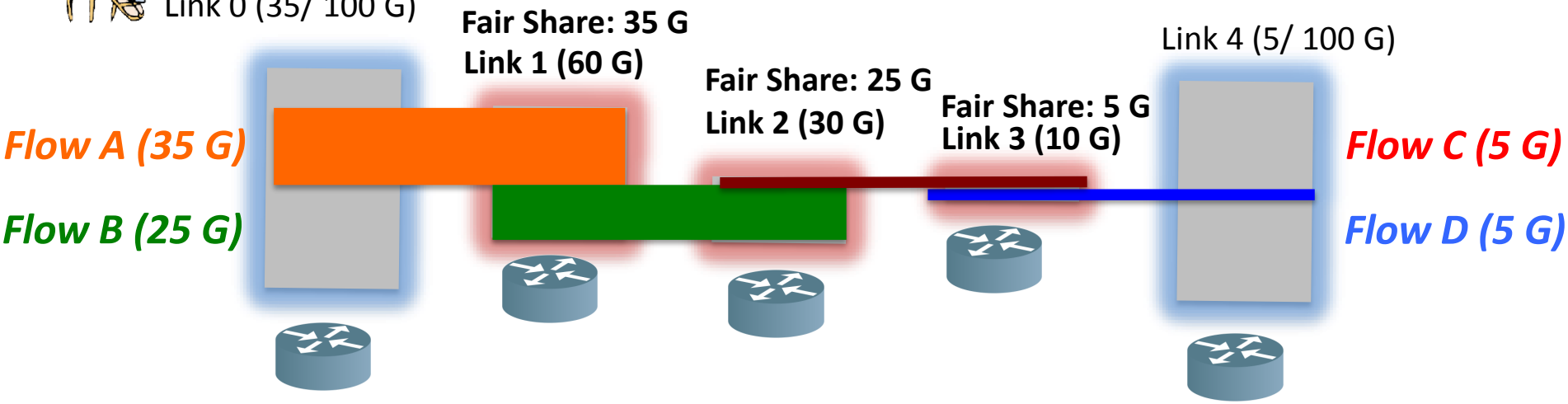
Waterfilling- 30 G link is fully used



Waterfilling- 60 G link is fully used



Fair Share of Bottlenecked Links



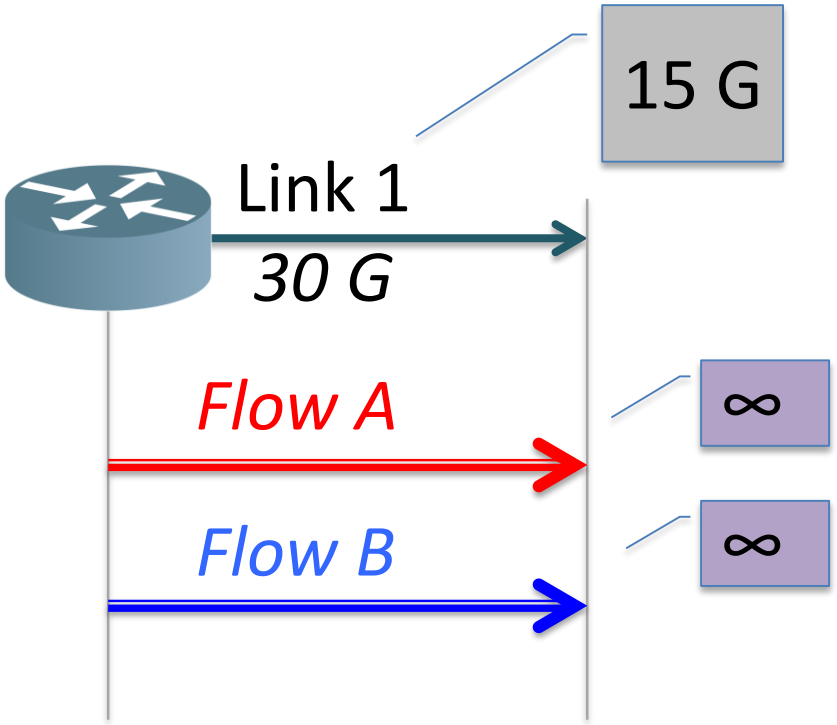
A *centralized* water-filling scheme may not scale.

Can we let the network figure out rates in a distributed fashion?

Fair Share for a Single Link

flow	demand
A	∞
B	∞

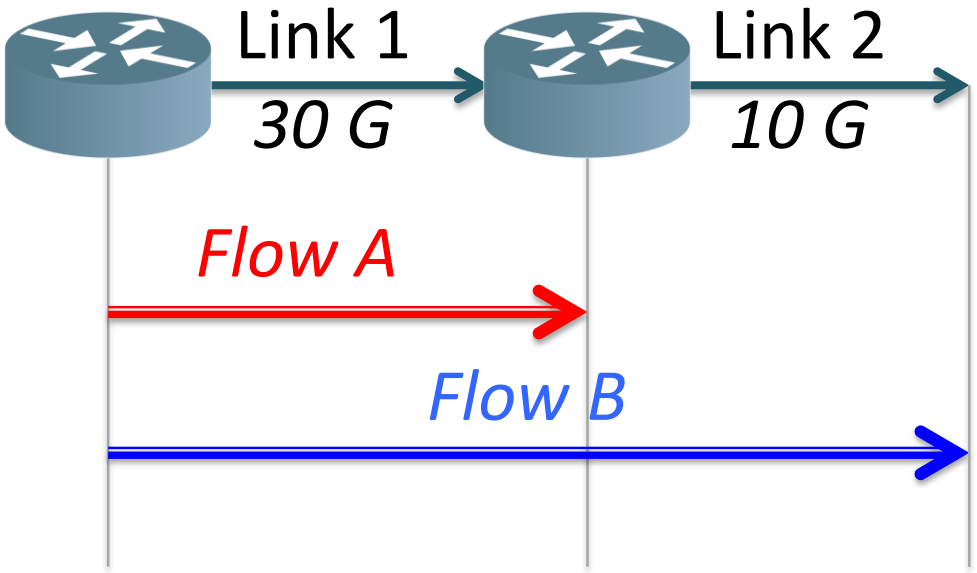
Capacity at Link 1: 30G
So Fair Share Rate: 30G/2 = 15G



A second link introduces a *dependency*

flow	demand
A	∞
B	~~10~~ 10 G

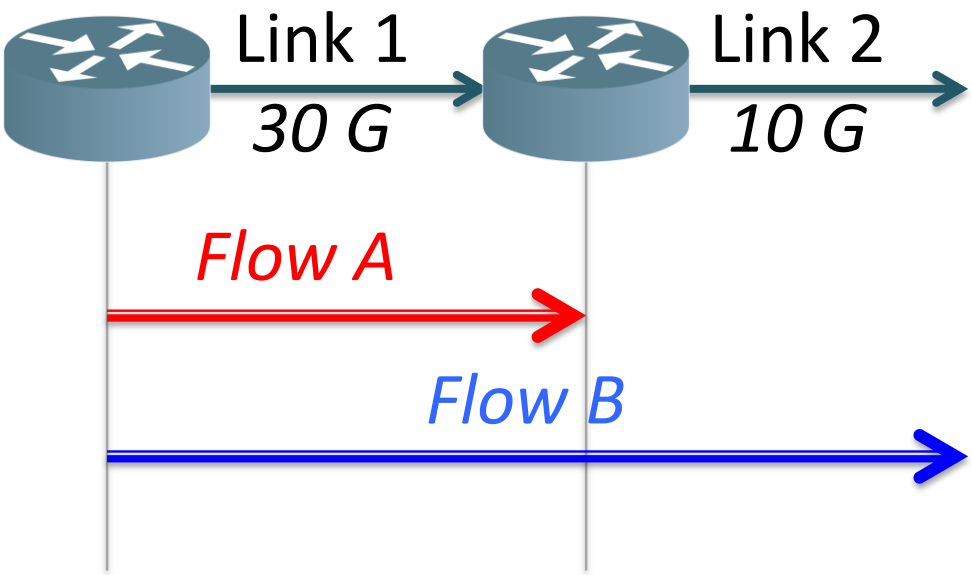
Capacity at Link 1: 30G
Demand of Flows restricted at other links: 10G
Number of unrestricted flows: 1
So Fair Share Rate: 30G-10G/1 = 20G



Proactive Explicit Rate Control (PERC)

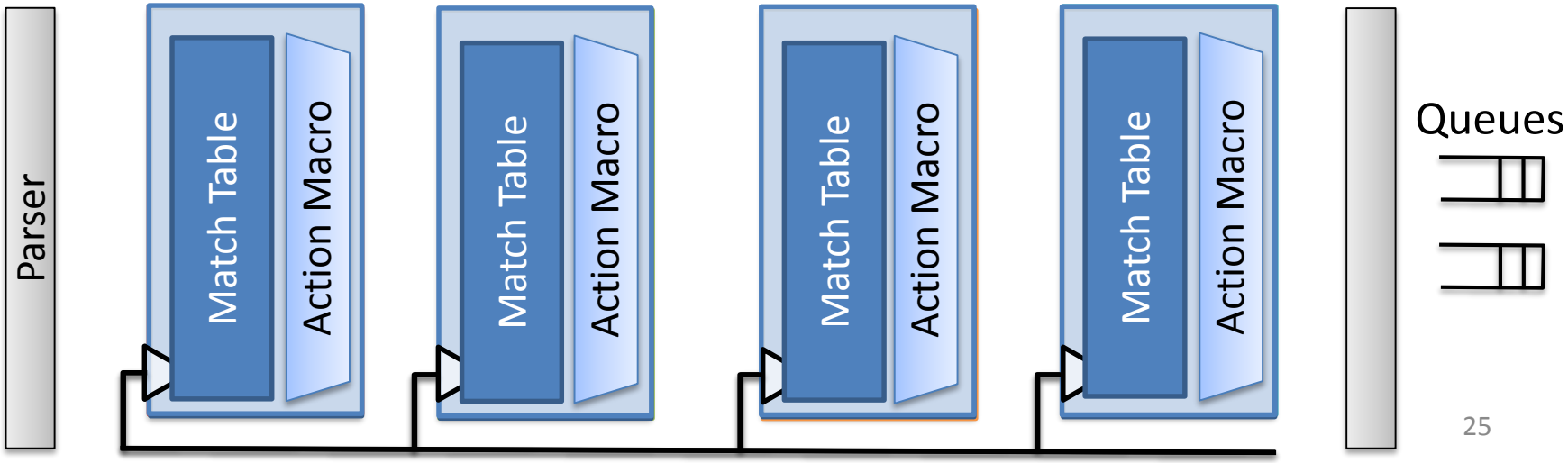
Control Packet For Flow B

d	∞	∞
f	?	?



Constraints of Programmable Forwarding Planes at 100 Gb/s

- Limited compute- action ~ ns, typically primitives like add/ compare etc.
- Limited info. that we can modify per packet.
- Limited area for state and look-up tables ~ MB, much of which is for L2/L3

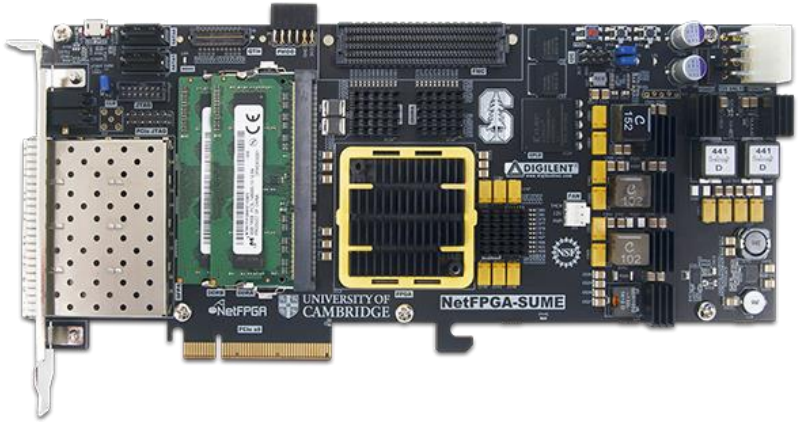


PERC in P4 → NetFPGA

P4 Front end

Xilinx SDNet Compilation

NetFPGA SUME Switch



Division of compute b/n end host & switch

flow	demand
A	∞
B	**10 G**

Capacity at Link 1: 30G

Demand of Flows restricted at other links: 10

Number of unrestricted flows: 1

Start Fair Share Rate: 30G/1 = 20G

Fair Share Rate: 30G-10G/1 = 20G

30G,10G,1

Interesting Questions

- Minimum time for a distributed scheme
- Minimum amount of state for provable convergence
- How many active flows in a max-min fair network?
- Imprecise demands → some reactive component