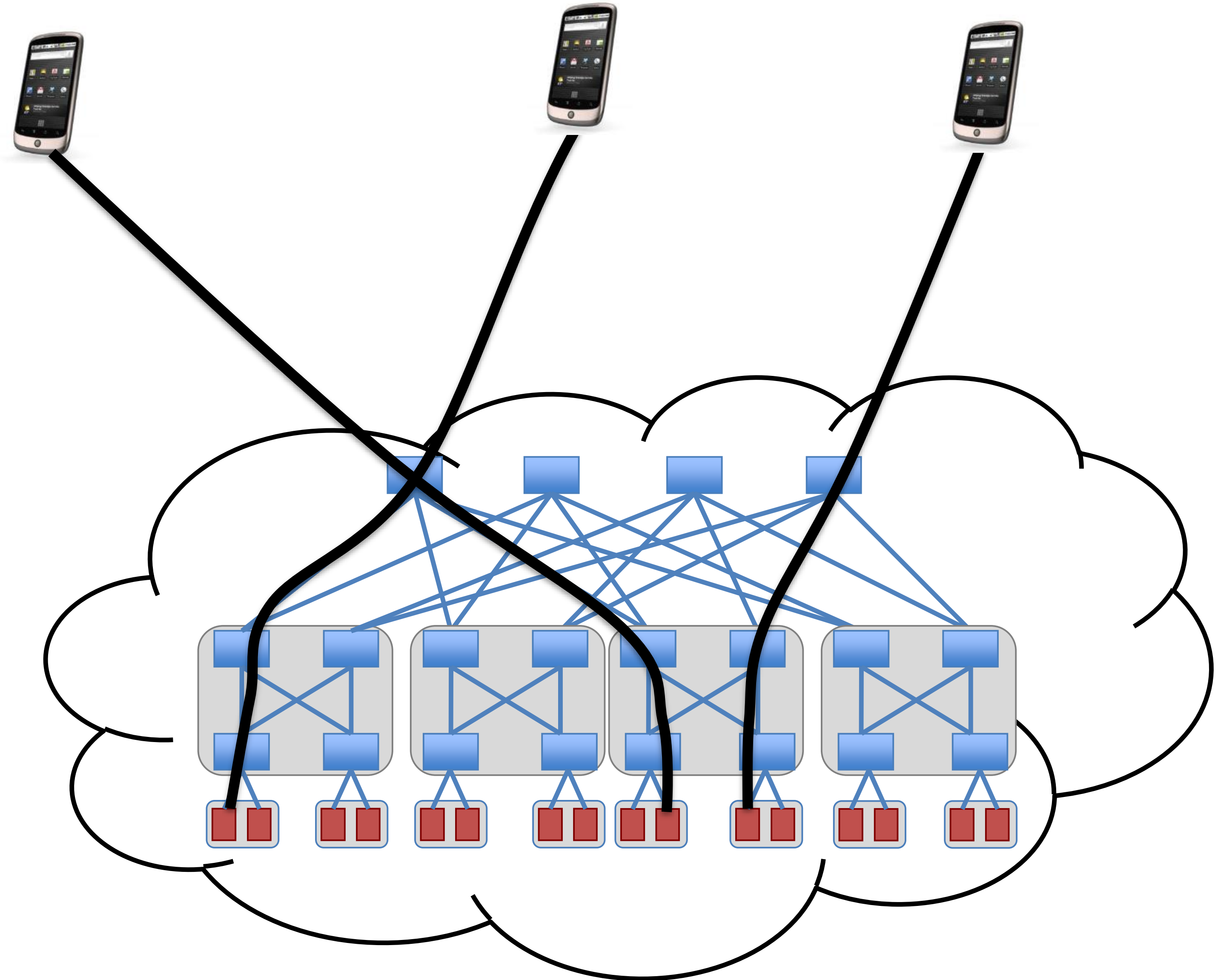
Stateless Datacenter Load Balancing with Beamer

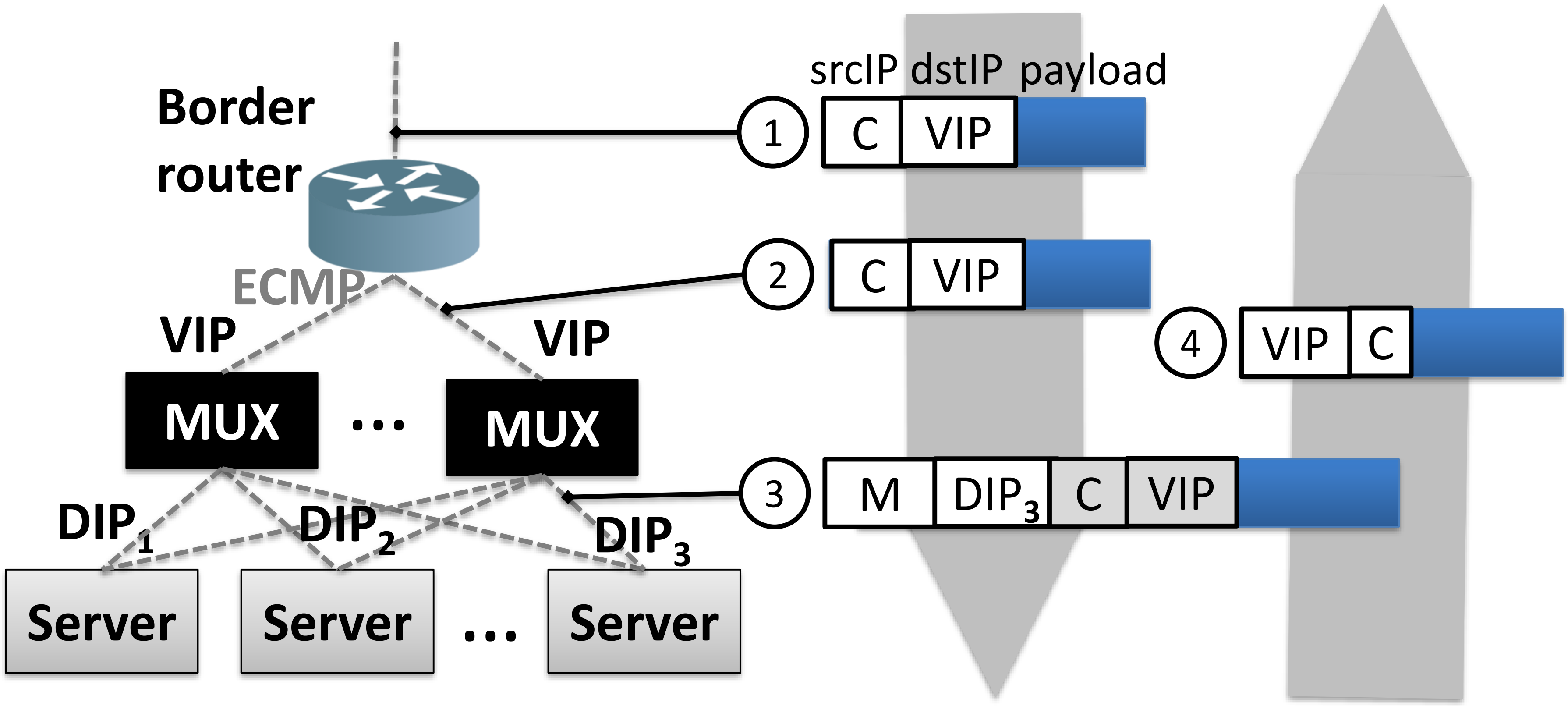
Vladimir Olteanu, Alexandru Agache,
Andrei Voinescu, Costin Raiciu
University Politehnica of Bucharest

Thanks to SSICLOPS SUPERFLUIDITY

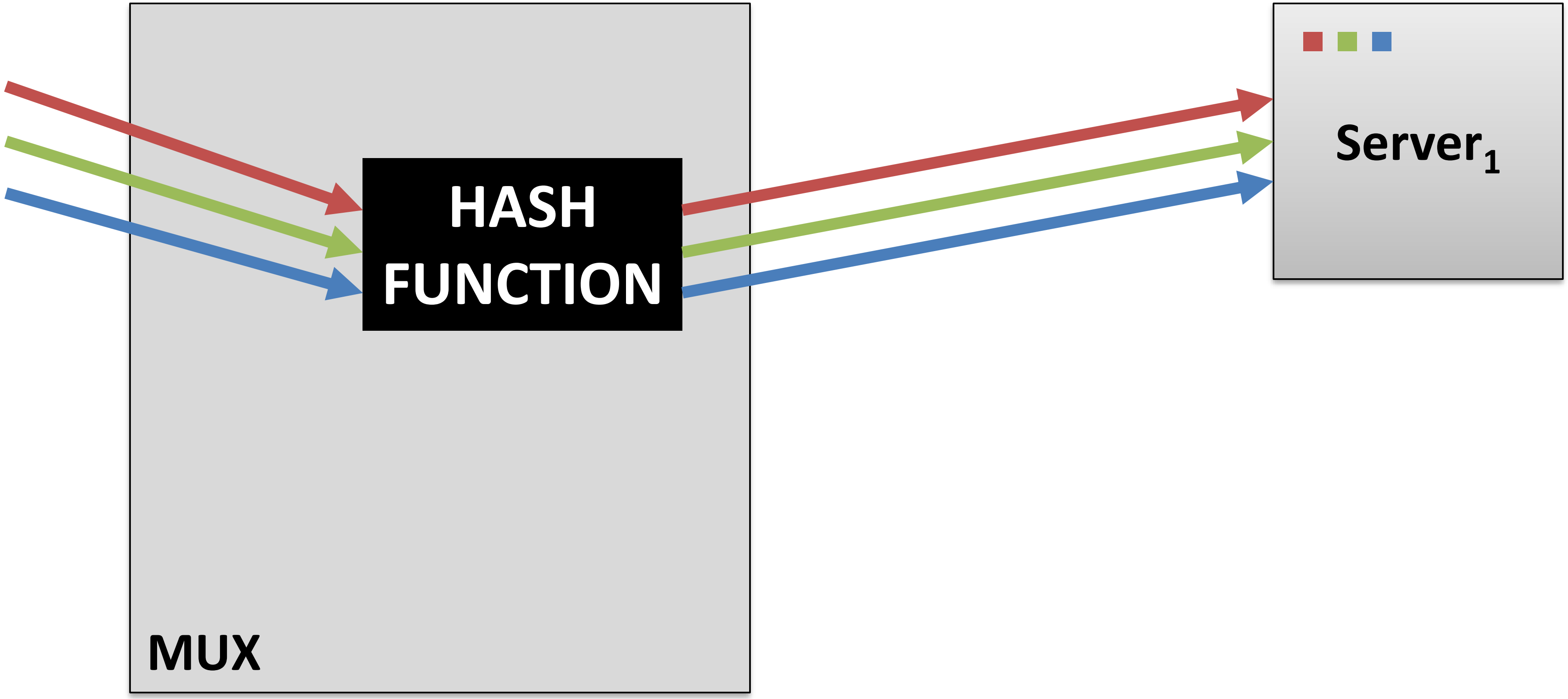
Datacenter load balancing



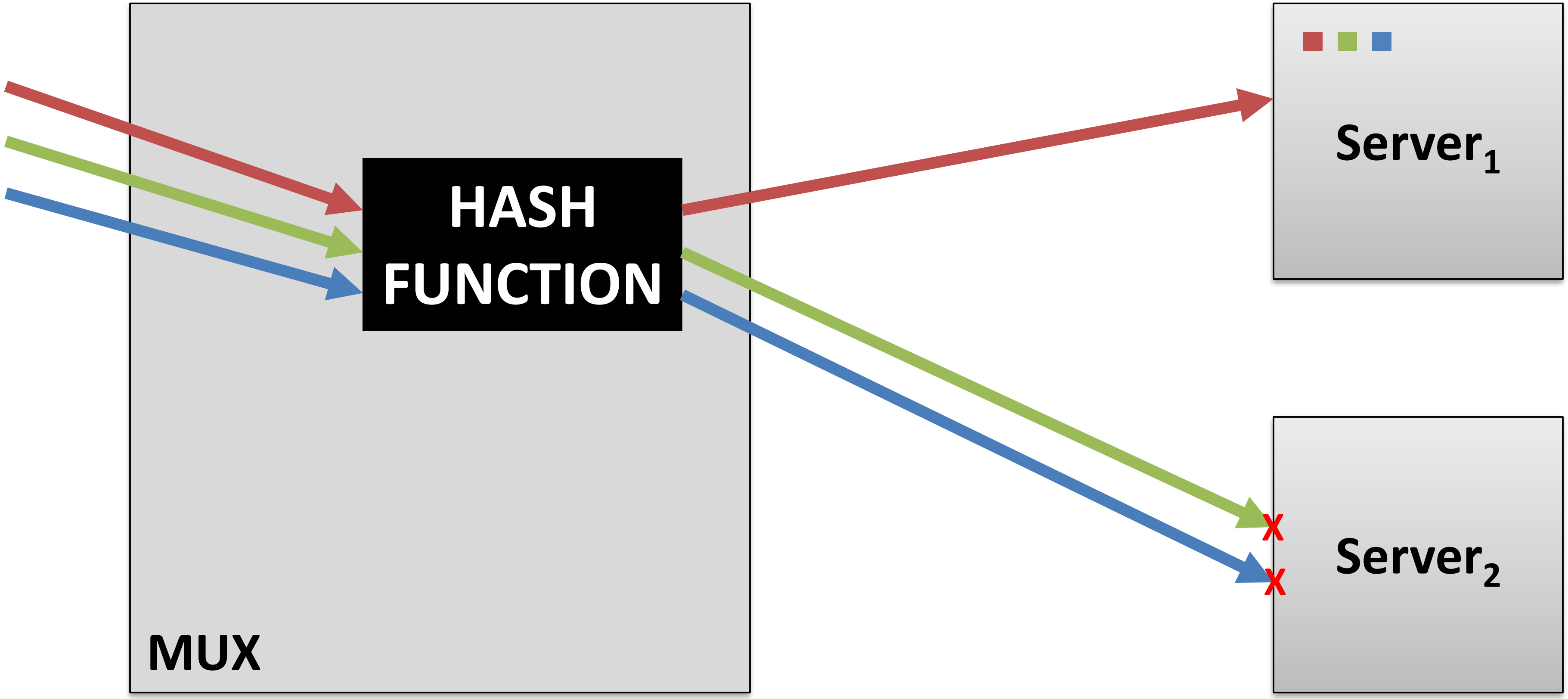
Datacenter load balancing today



Strawman approach

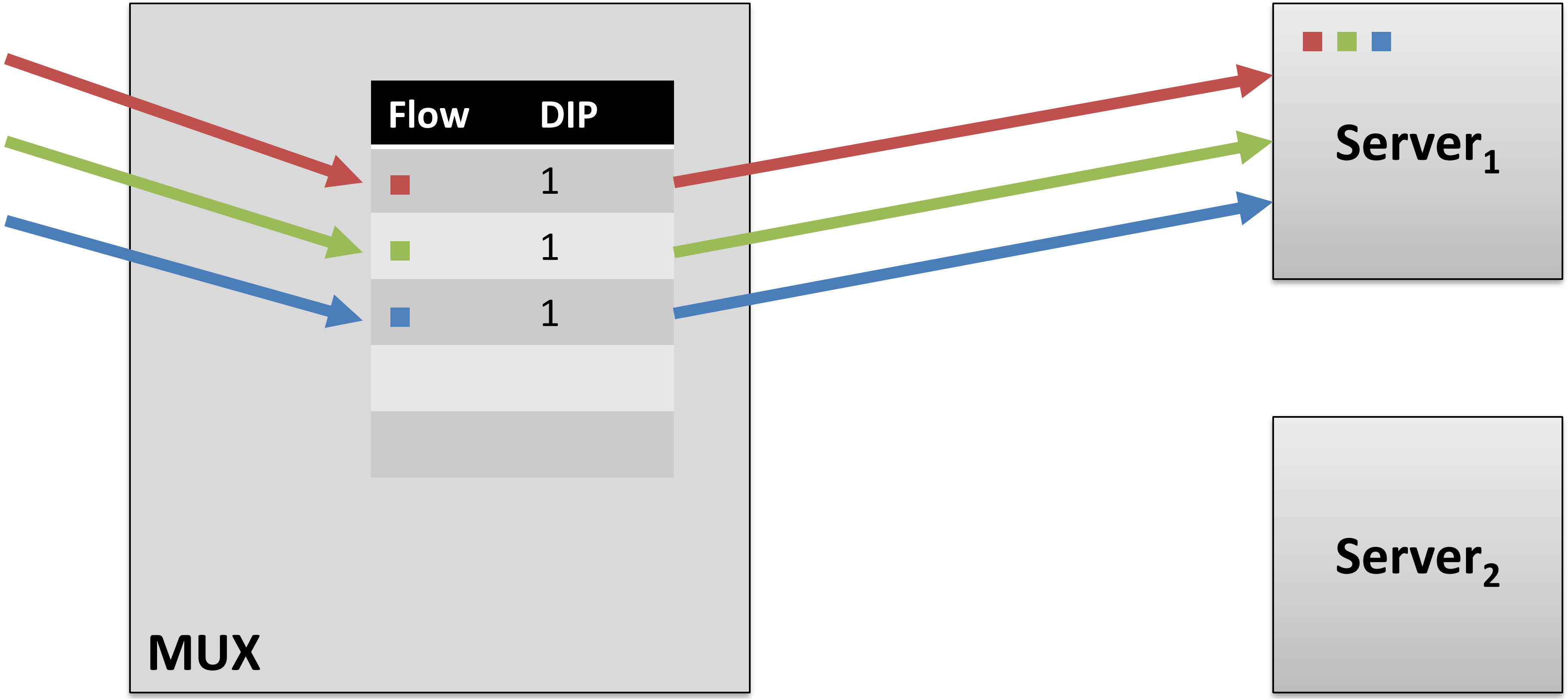


Strawman approach

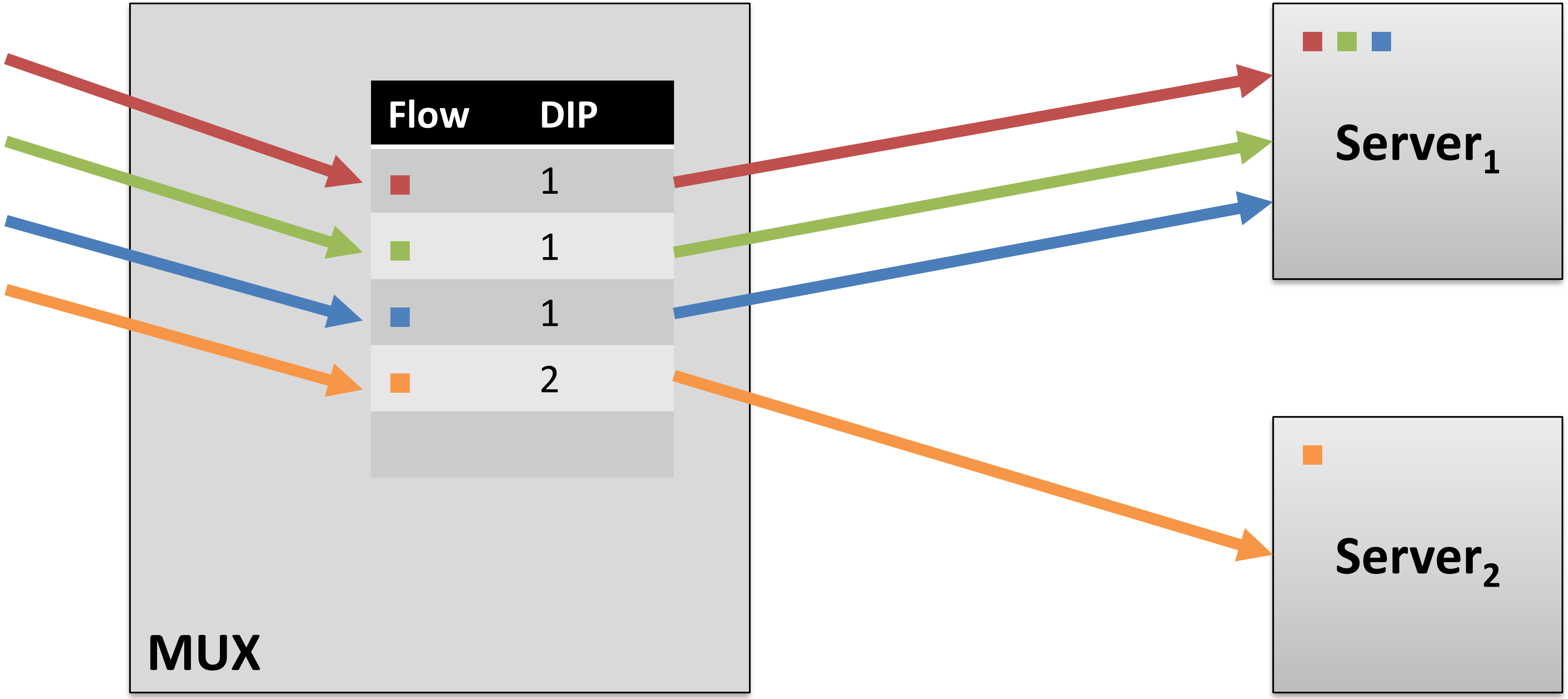


- Adding/removing servers breaks connection affinity

Load balancers use state to ensure connection affinity

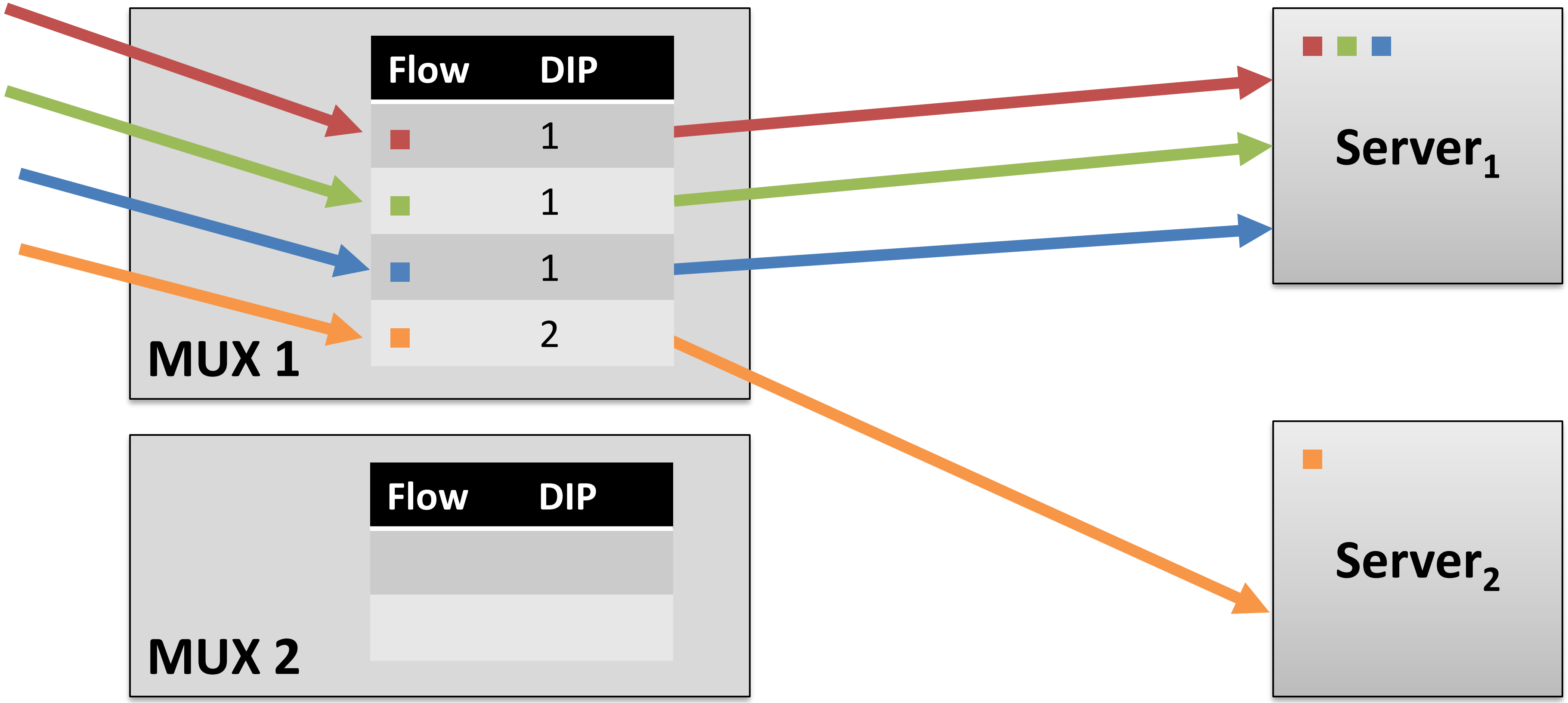


Load balancers use state to ensure connection affinity

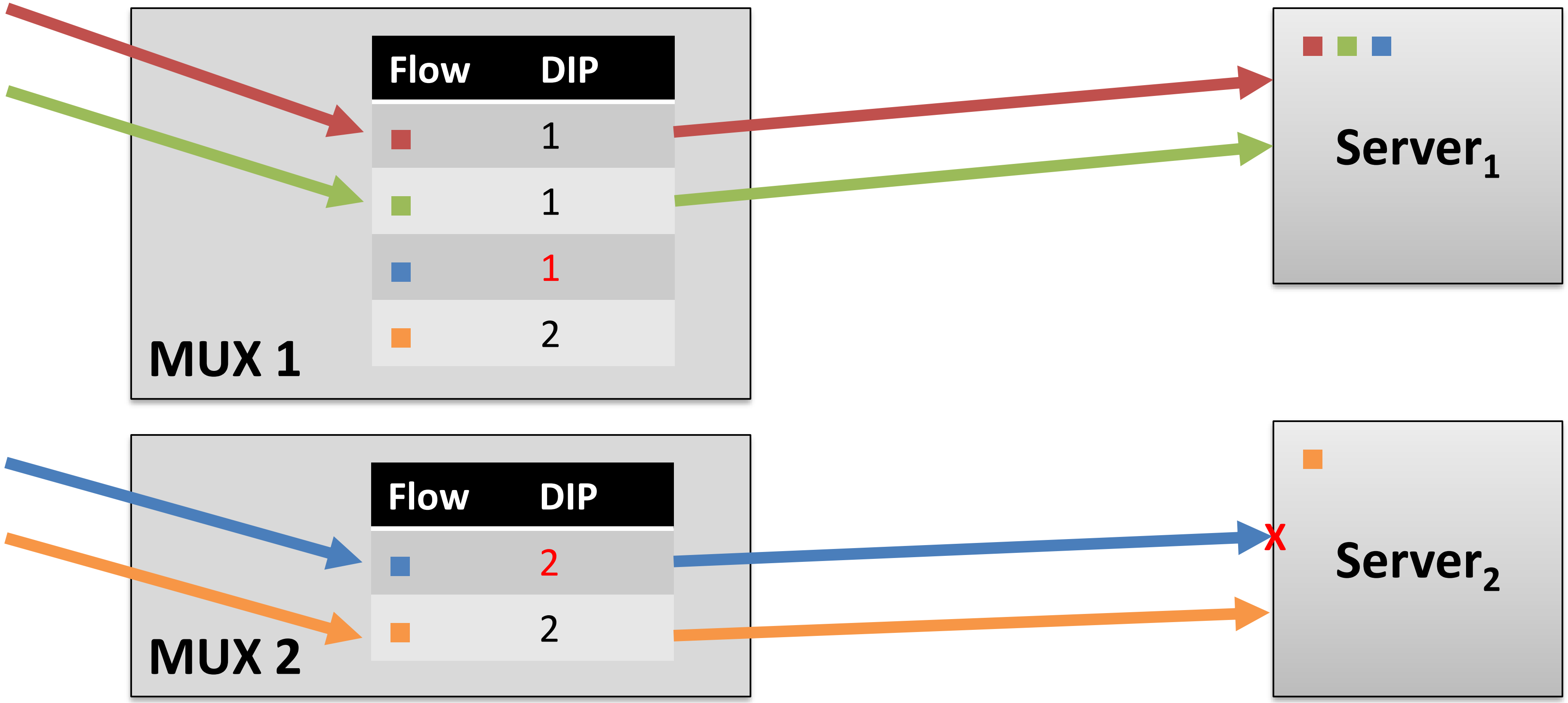


- Only new connections are hashed

Load balancers use state to ensure connection affinity

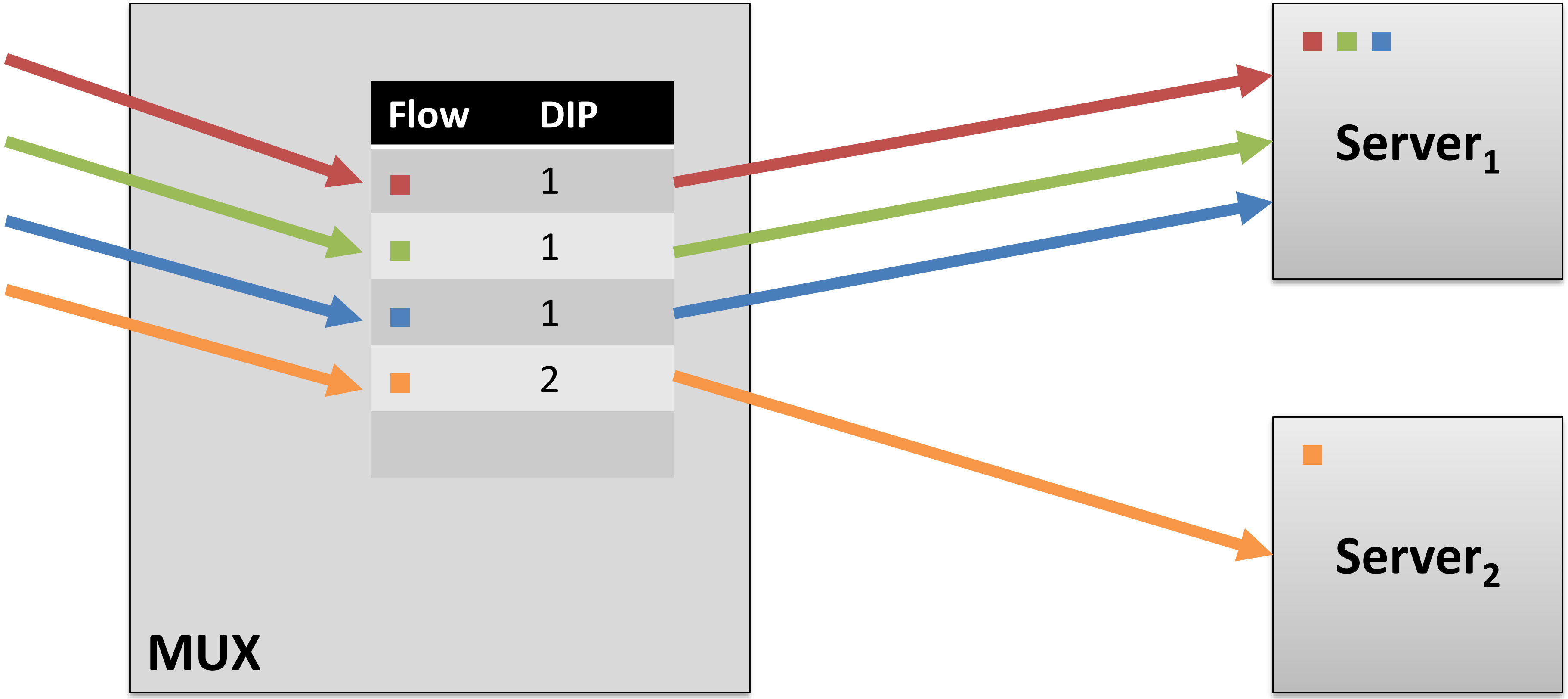


Load balancers use state to ensure connection affinity

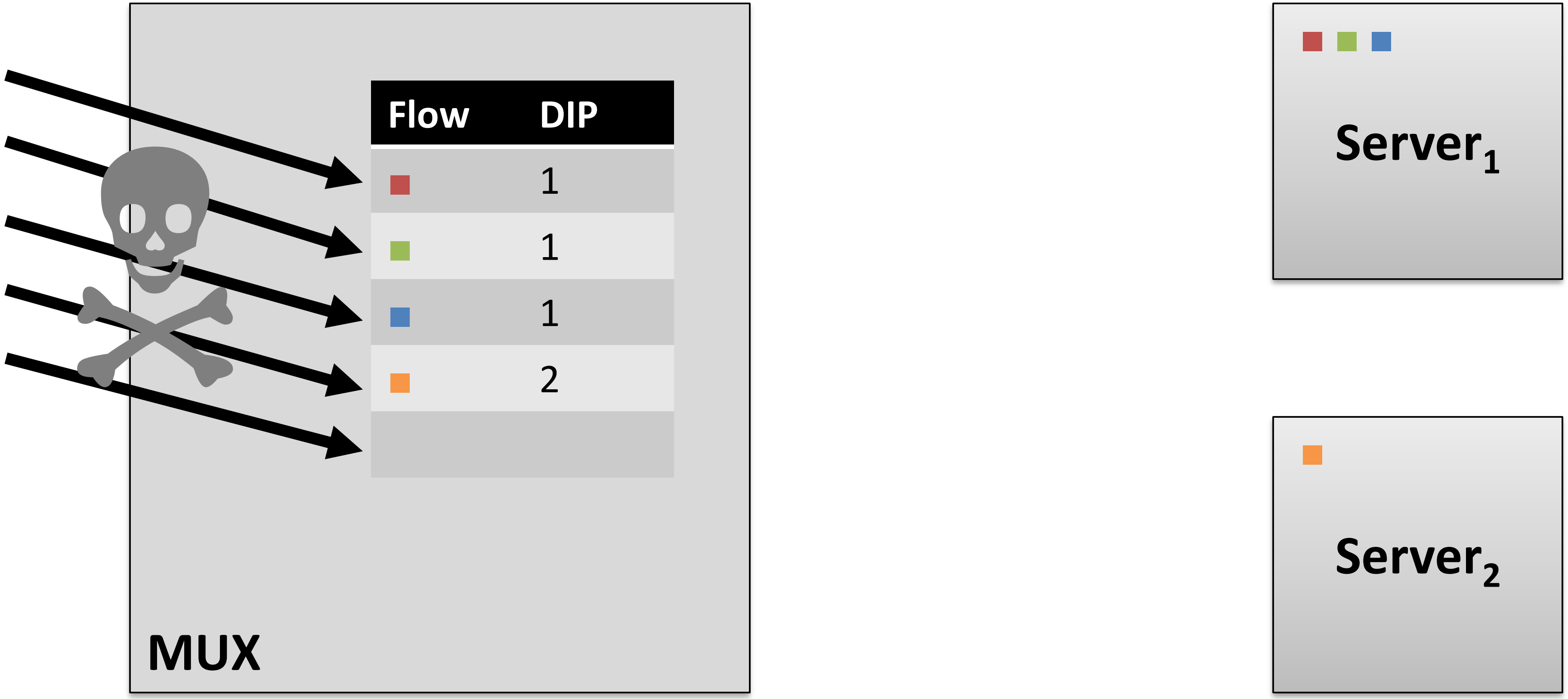


- Scaling mux pool may reset some connections

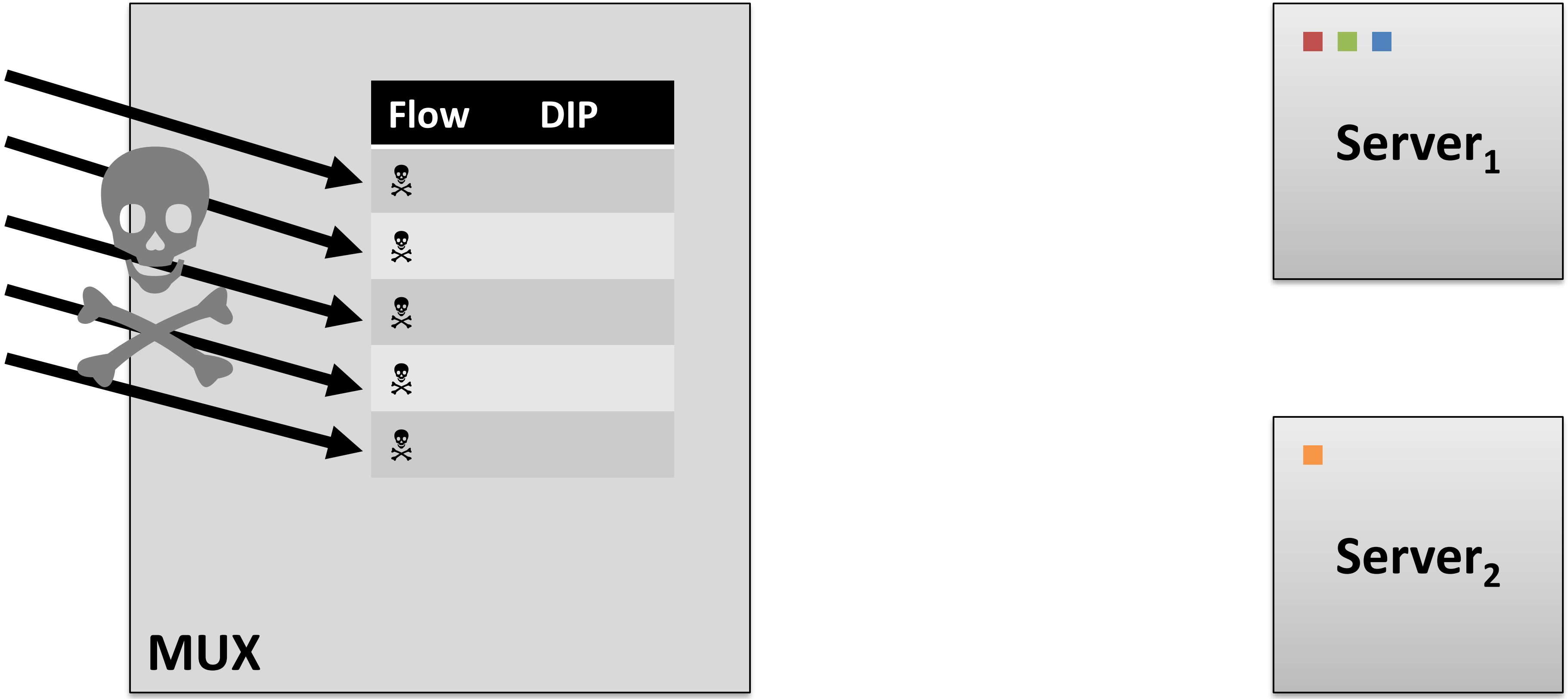
Load balancers use state to ensure connection affinity



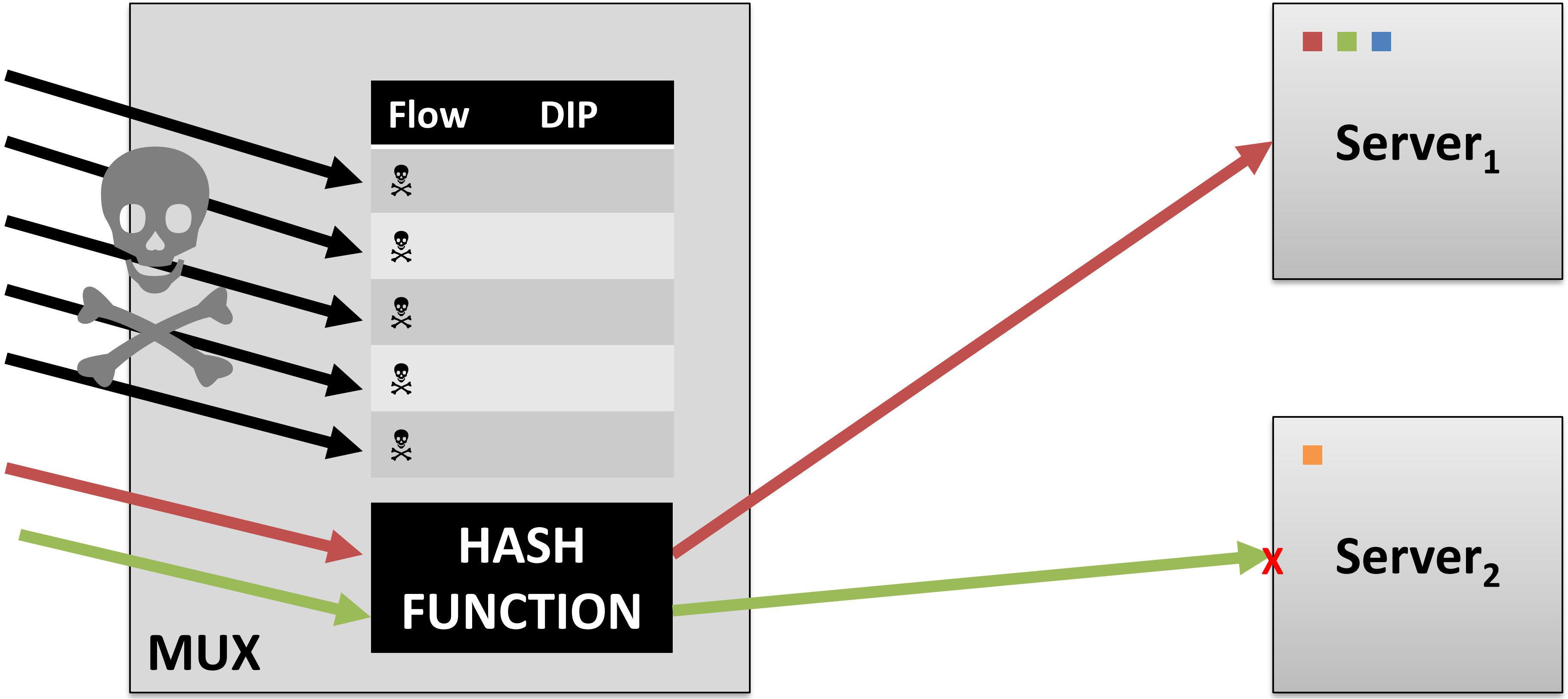
SYN floods use up state memory



SYN floods use up state memory



SYN floods use up state memory



- Back to the straw man approach

Stateful designs don’t guarantee connection affinity

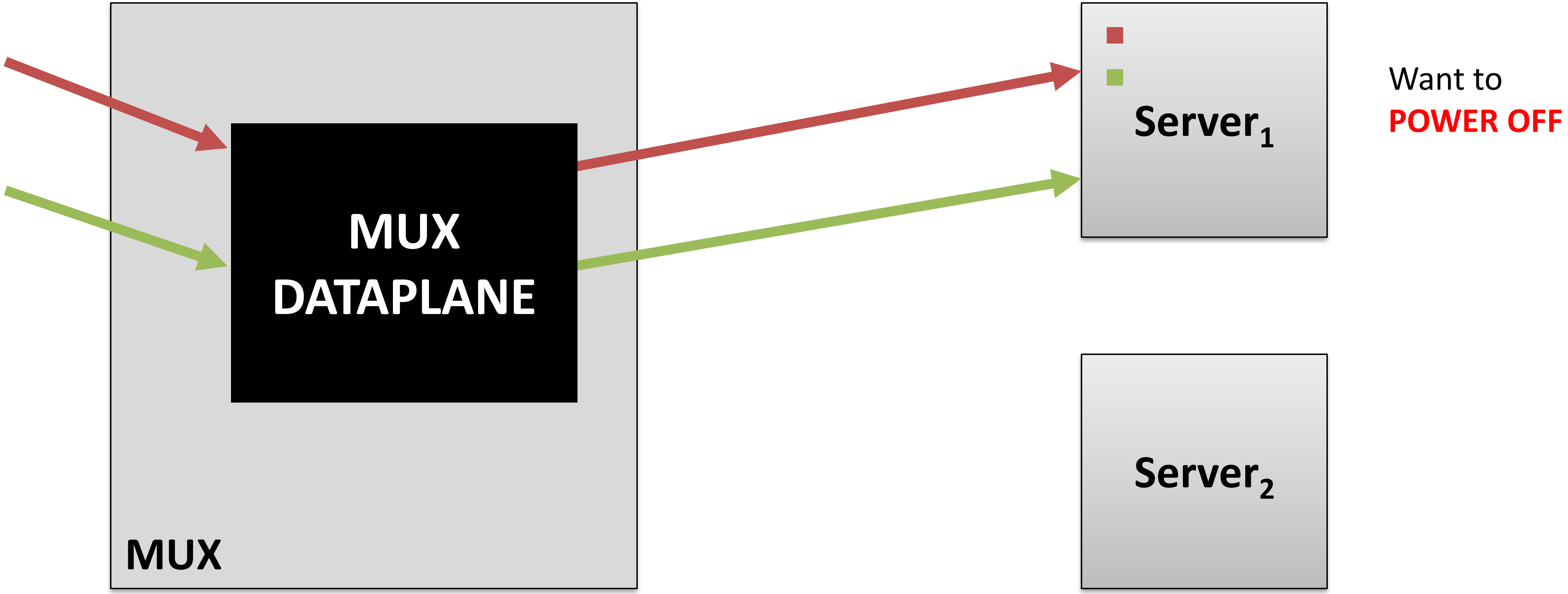
Beamer: **stateless** load balancing

Beamer muxes do not keep per-connection state;
each packet is forwarded independently.

When the target server changes, connections may break.

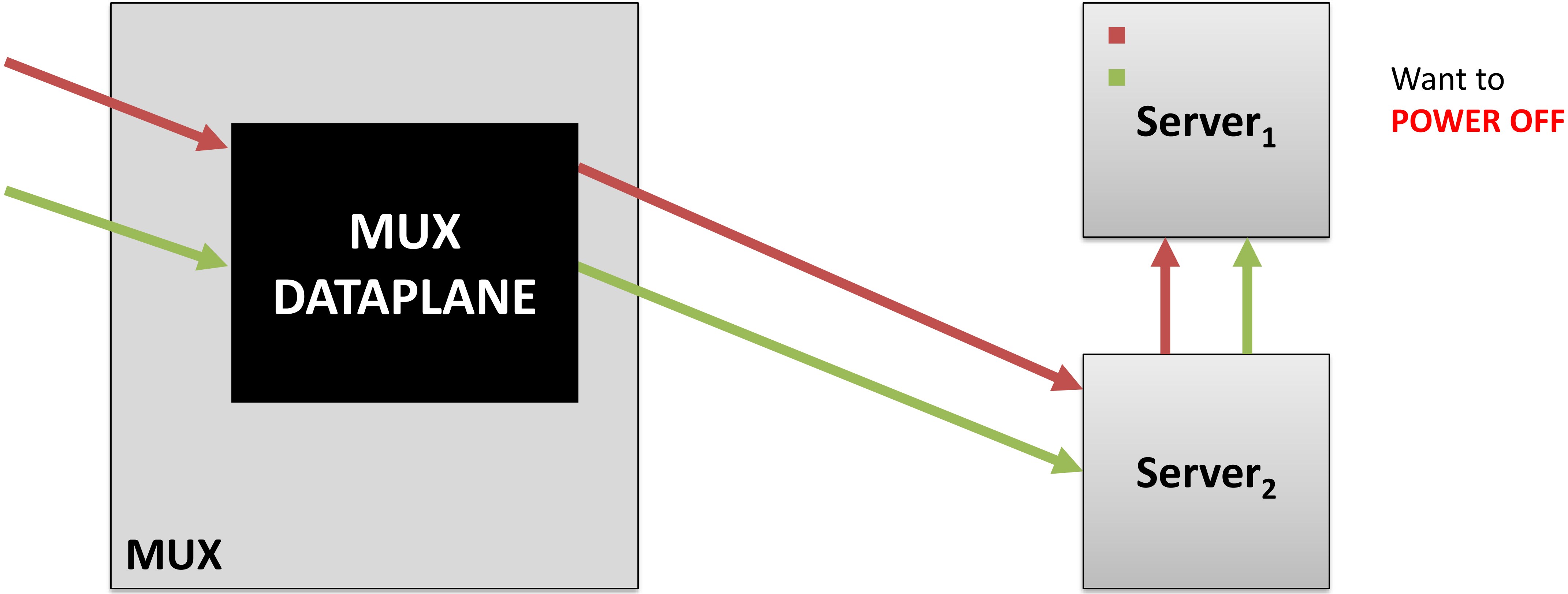
Beamer uses state stored in servers to
redirect stray packets.

Beamer daisy chaining



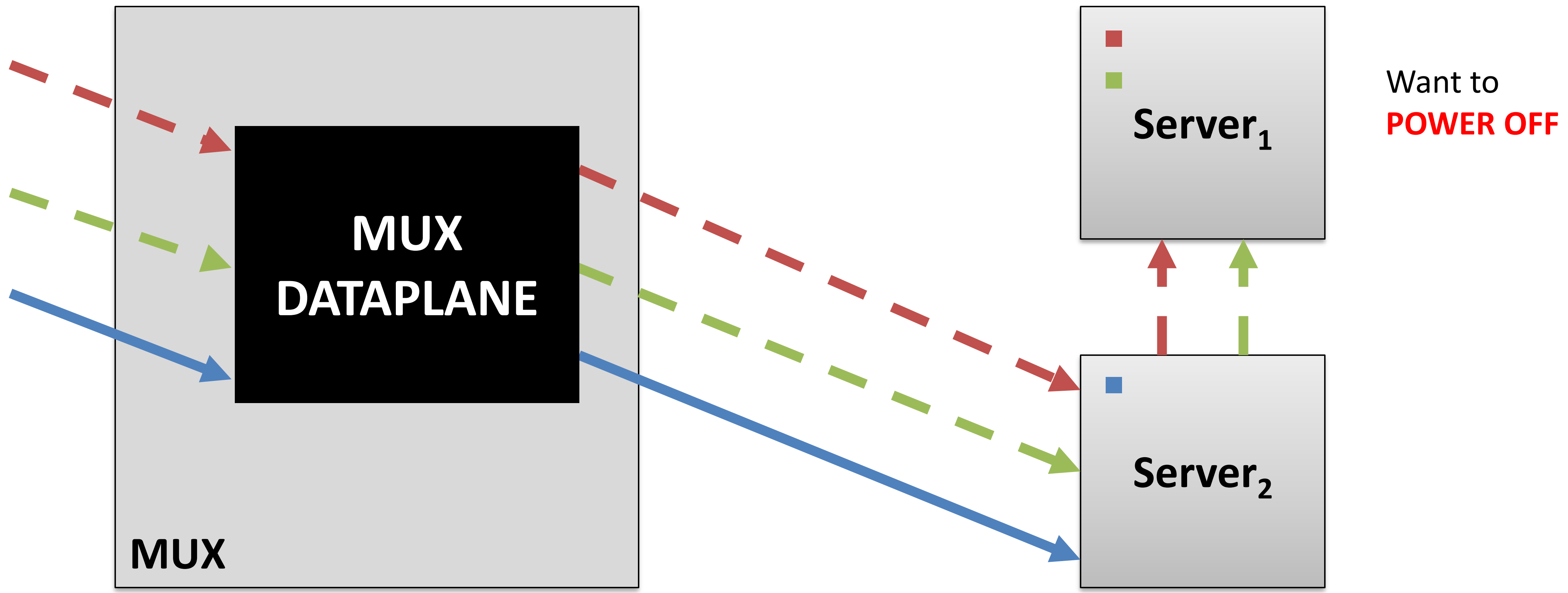
- Used when reassigning traffic

Beamer daisy chaining



- Used when reassigning traffic

Beamer daisy chaining



- Daisy-chained connections die off in time

Balancing packets in Beamer

Which hashing algorithm is best?

	Low churn	Good load balancing	Few rules in dataplane
ECMP	✗	✓	✓
Consistent Hashing	✓	✗	✓
Maglev Hashing	✓	✓	✗

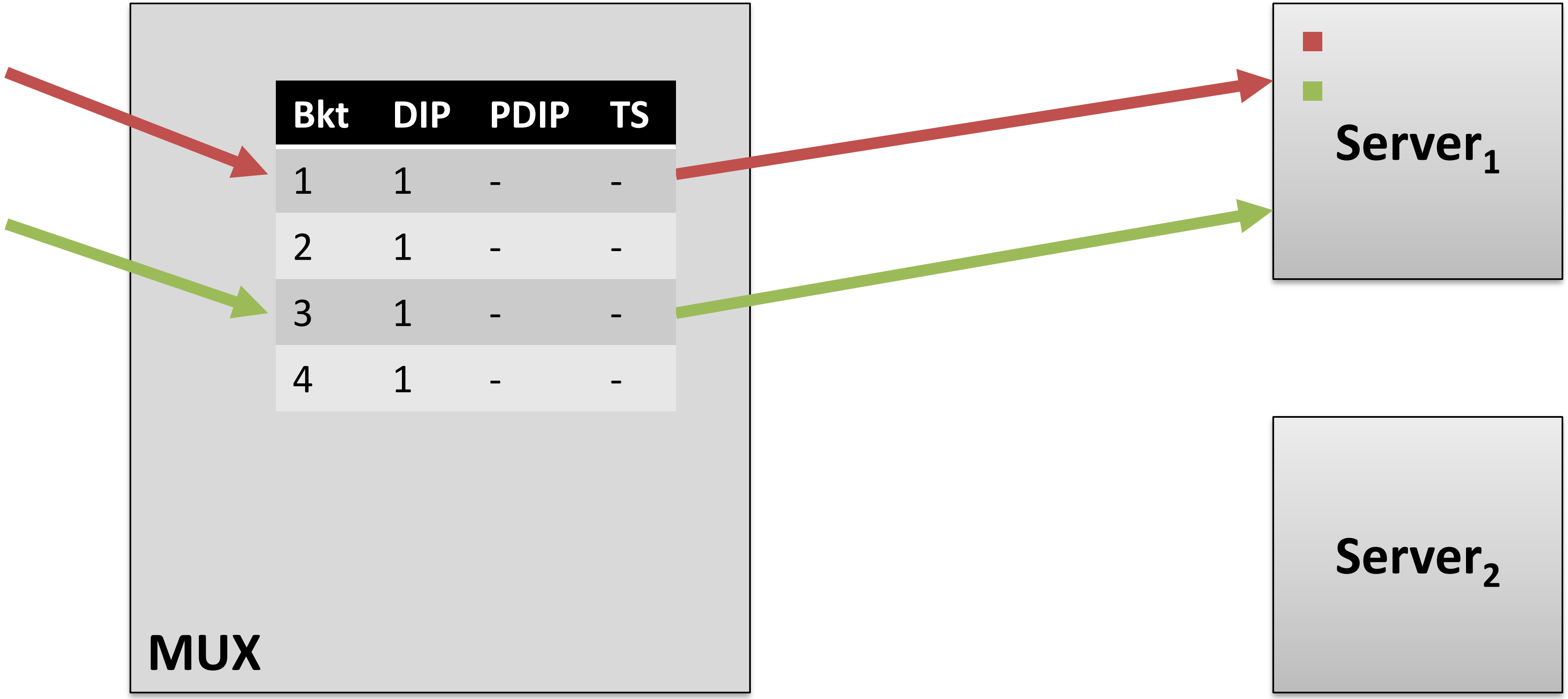
Beamer hashing

Indirection layer
Pick number of buckets B > N, number of servers

Mux dataplane:
- Assign each bucket to one server
- Bucket-to-server mappings known by all muxes
- Maintained by a centralized controller

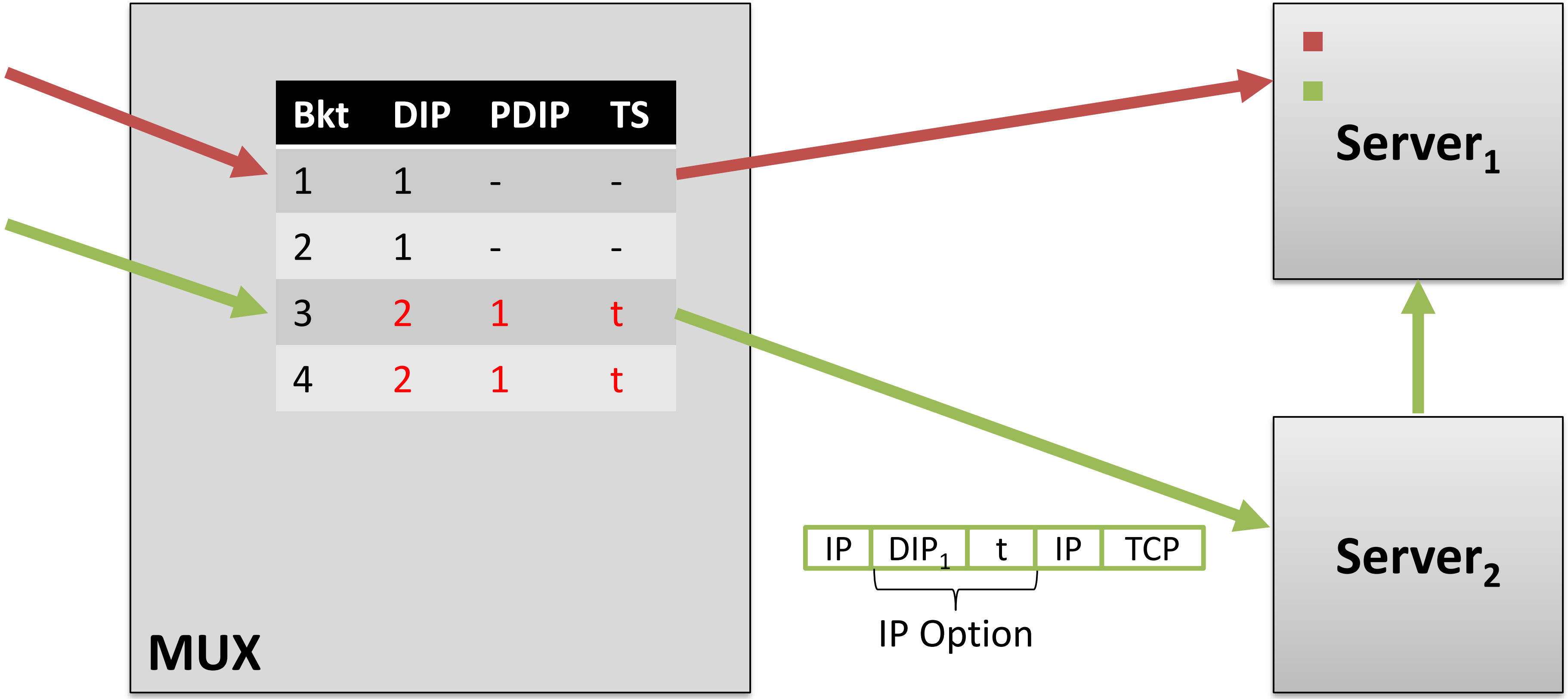
Mux algorithm:
- Hash each packet modulo B
- Send to corresponding server

Beamer at work



Bkt	DIP	PDIP	TS
1	1	-	-
2	1	-	-
3	1	-	-
4	1	-	-

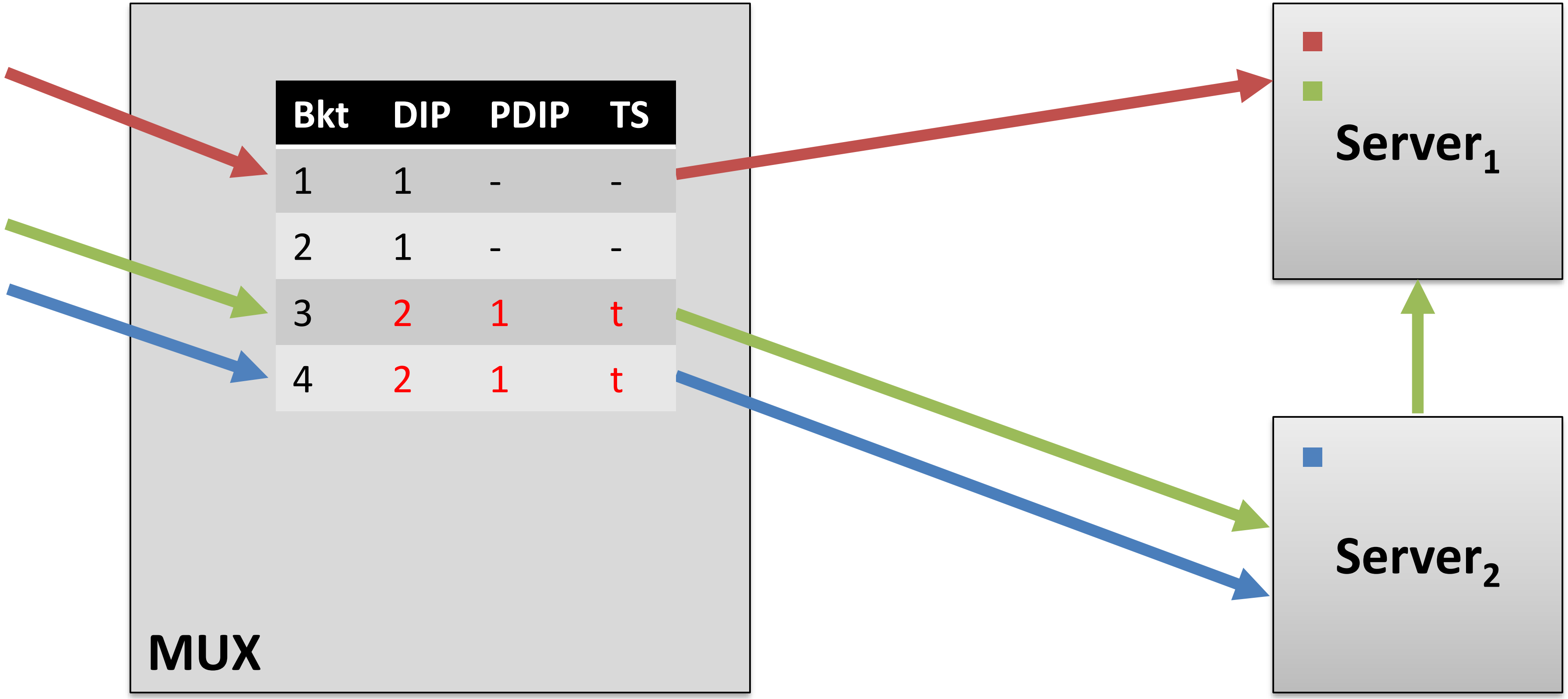
Beamer at work



Bkt	DIP	PDIP	TS
1	1	-	-
2	1	-	-
3	2	1	t
4	2	1	t

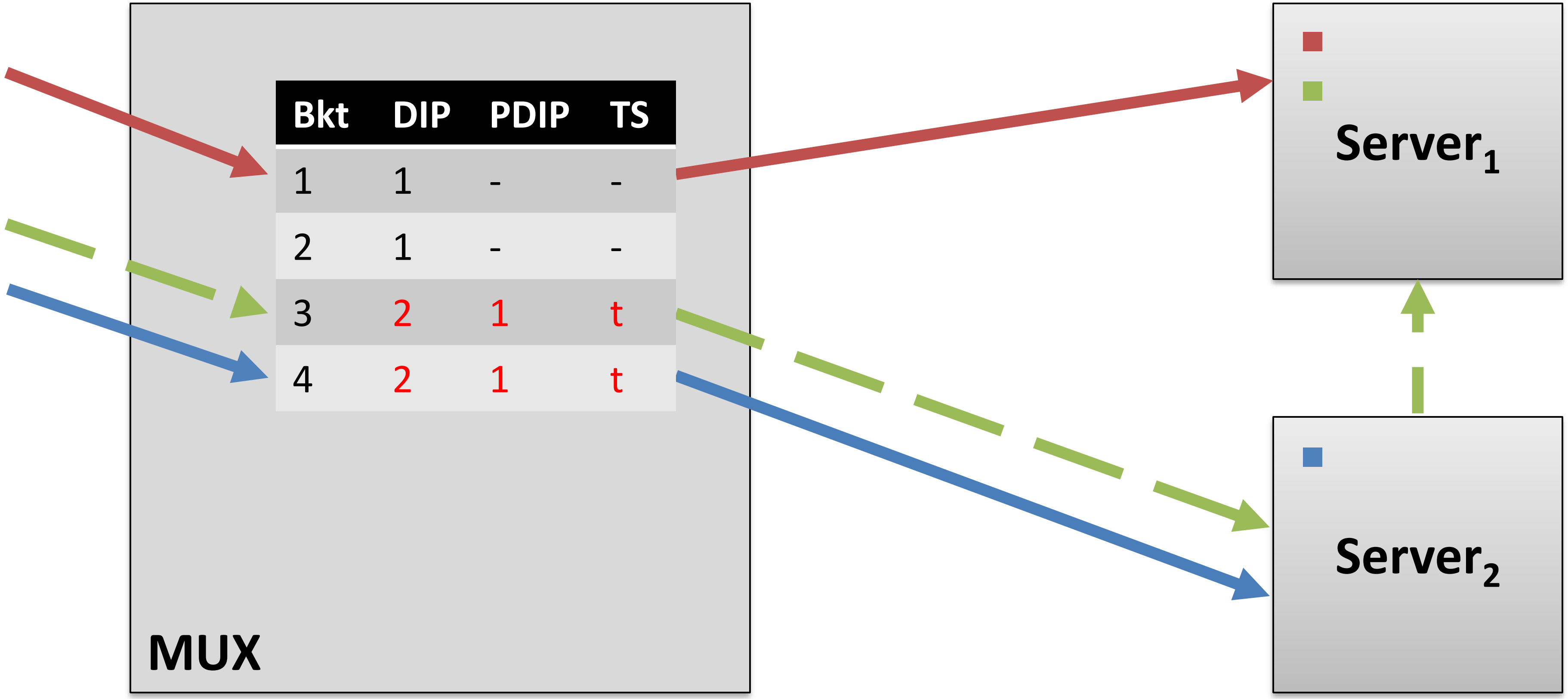
- Packets contain previous server and time of reassignment

Beamer at work



- New connections are handled locally

Beamer at work



Bkt	DIP	PDIP	TS
1	1	-	-
2	1	-	-
3	2	1	t
4	2	1	t

- Daisy chained connections die off in time

Benefits of Beamer muxes

Less memory usage and cache thrashing

Implementable in hardware: P4

Interchangeable

Resilient to SYN flood attacks

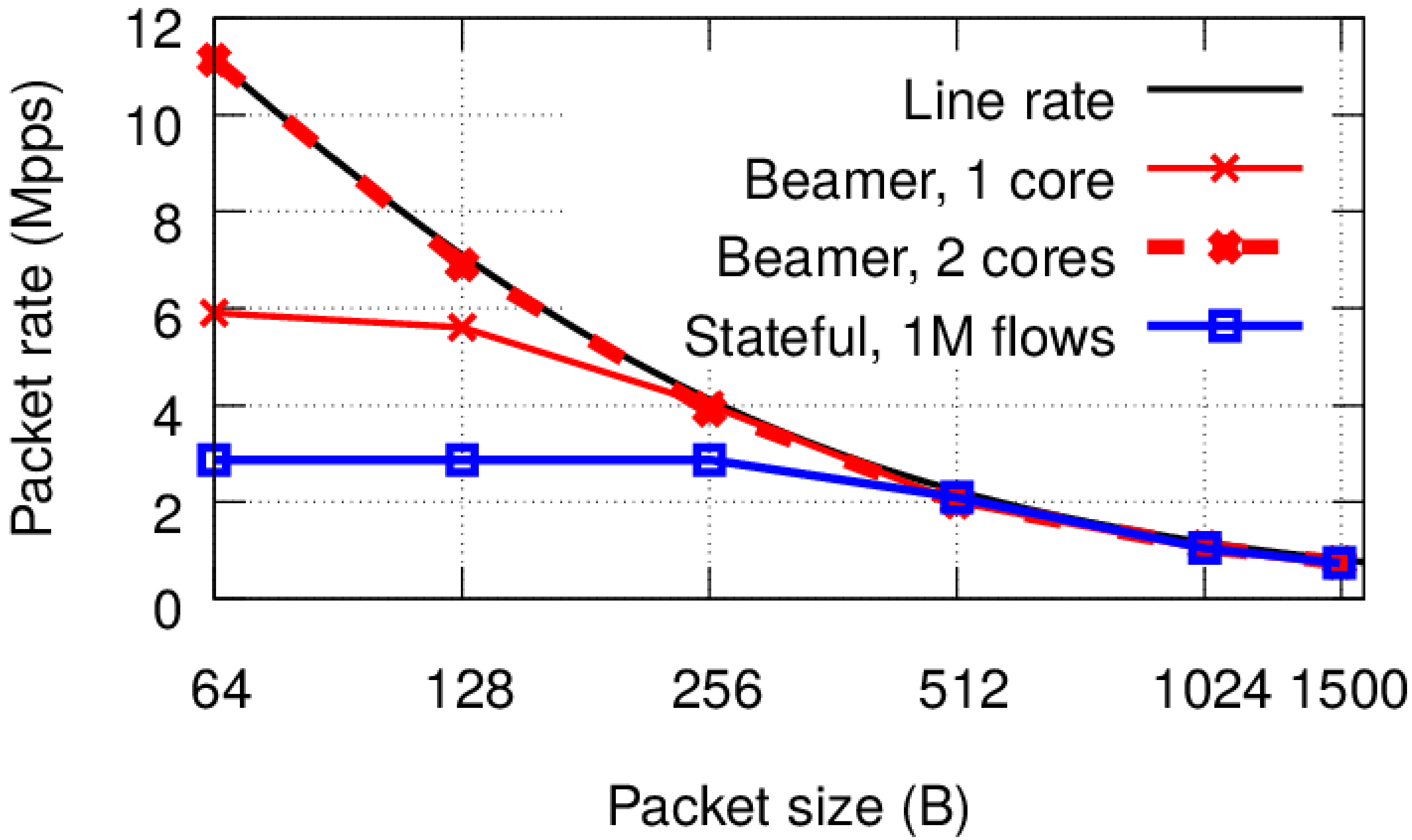
Cost: 16B encapsulation overhead per packet

Beamer mux performance

- Software implementation on top of netmap
- Machine: Xeon E5-2697 v2 @ 2.70GHz, Intel 82599 NIC

- Compared against:
  – Stateful – similar performance to Google’s Maglev [NSDI’16]

Single mux performance



Realistic traffic

HTTP traffic from recent MAWI trace

- Packets replayed back-to-back

36Gbps of upstream traffic on 7 cores

- 15 times more downstream traffic: 540Gbps

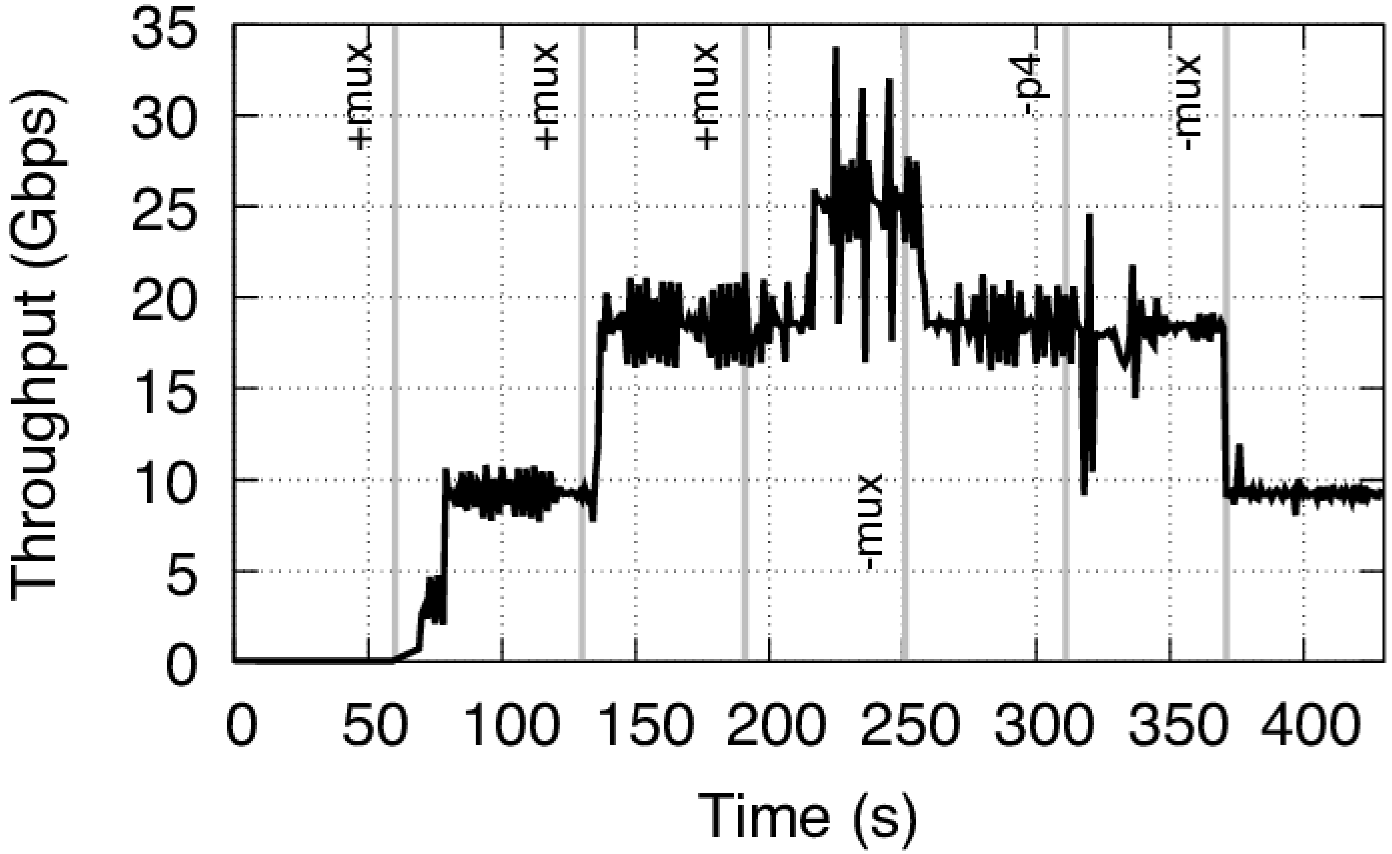
Rough estimate: 50-500 servers/mux

- Assuming servers source 1-10Gbps of traffic

Testbed evaluation

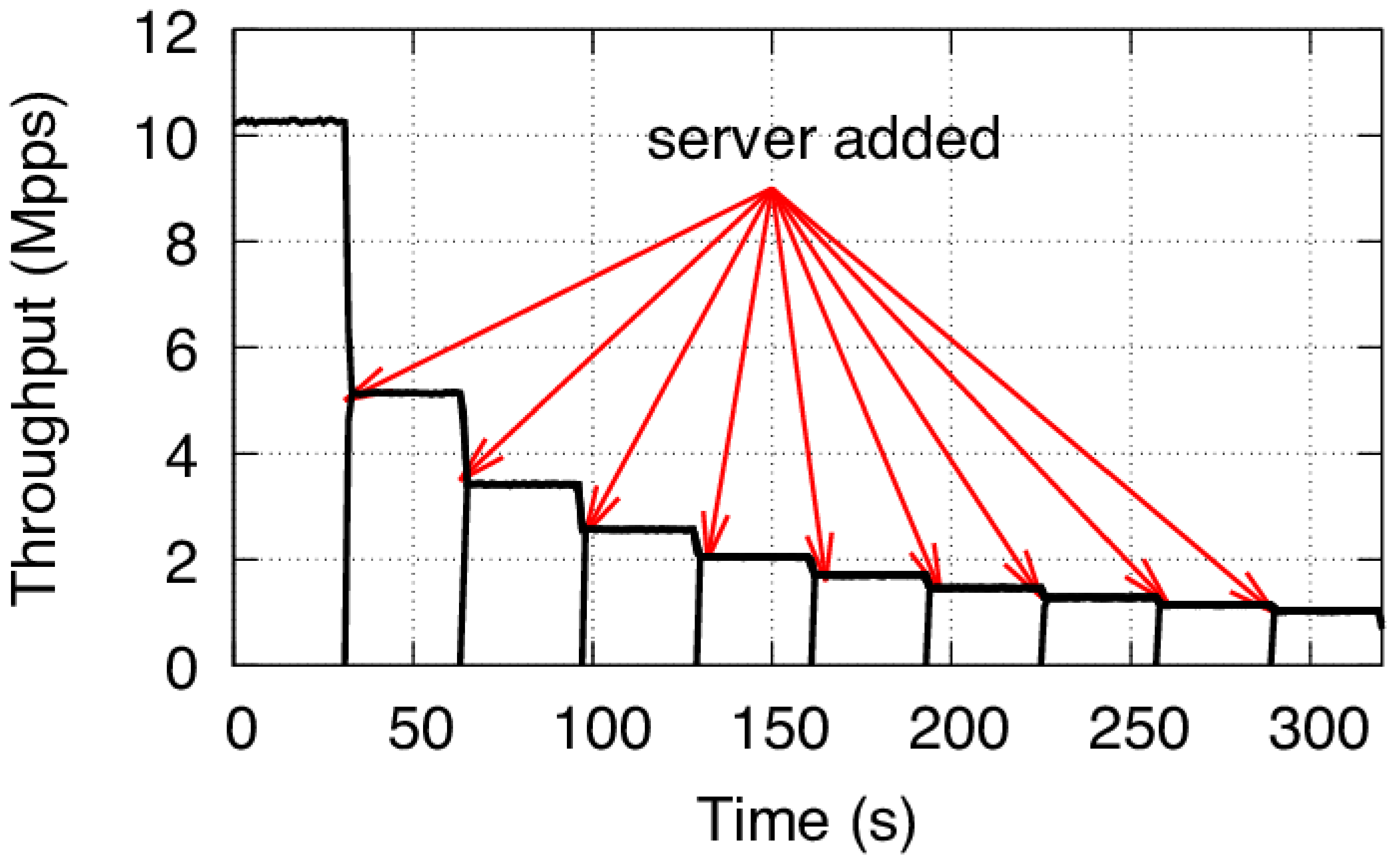
- 20 machines
  - 10Gbps NICs
- IBM RackSwitch 8264 as border router
- Software muxes
  - P4 reference implementation also used

Adding and removing muxes



- Mux failures and churn are handled smoothly

Adding servers



- Beamer spreads traffic evenly across servers

Connection affinity under SYN flood attacks

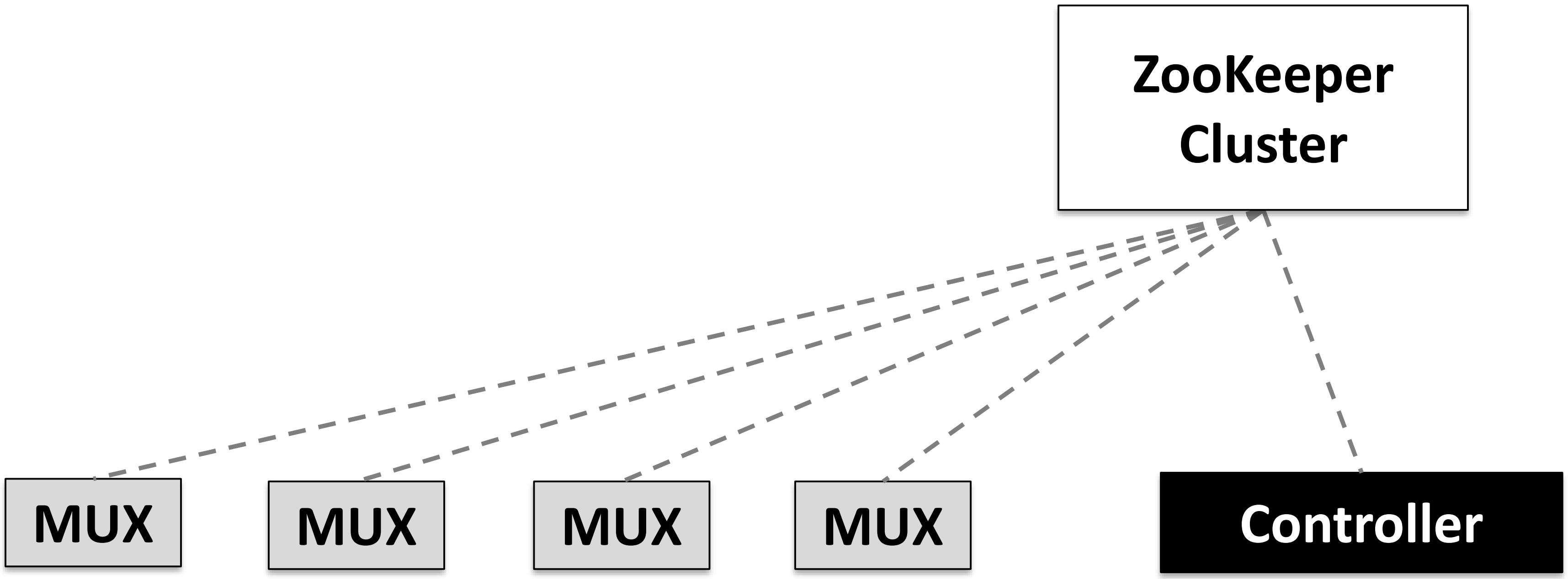
1Mpps SYN flood

2 muxes, 8 servers, 700 running connections

Drain servers during SYN flood

DIPs Drained	0	1	2	4
Stateful	0	87±2	148±8	351±21
Beamer	0	0	0	0

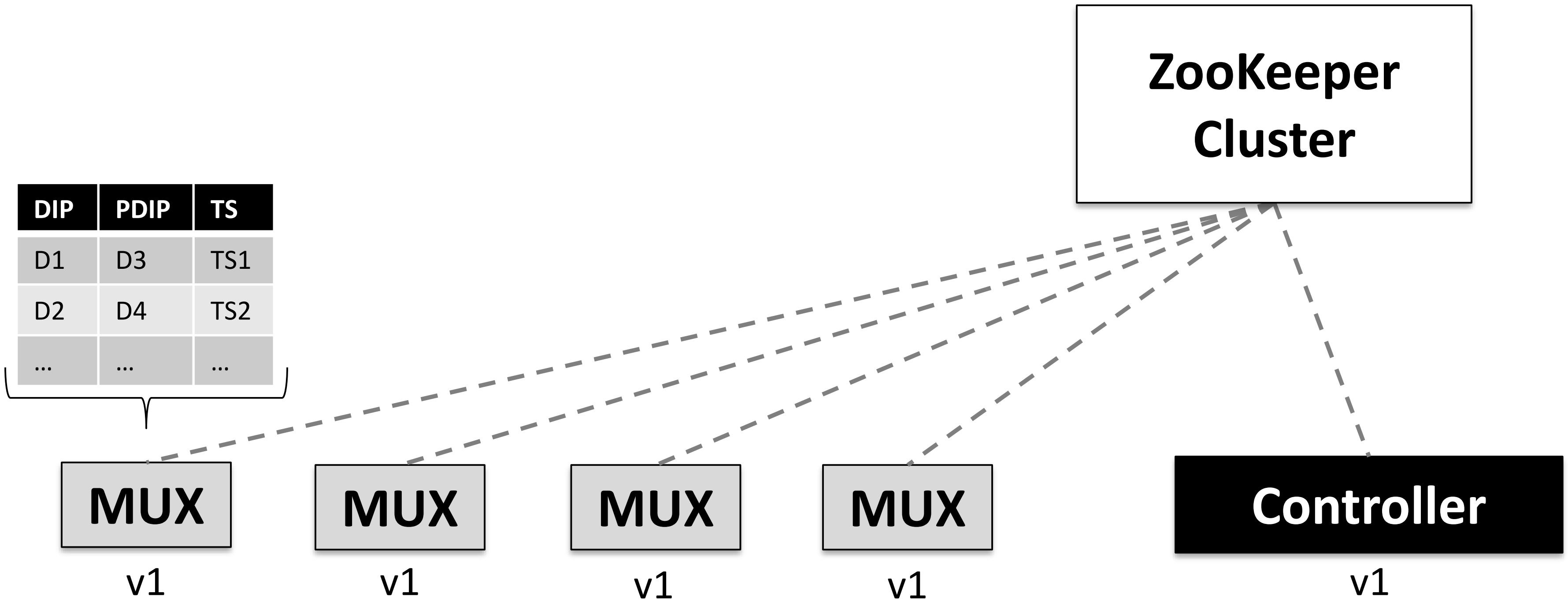
Control plane



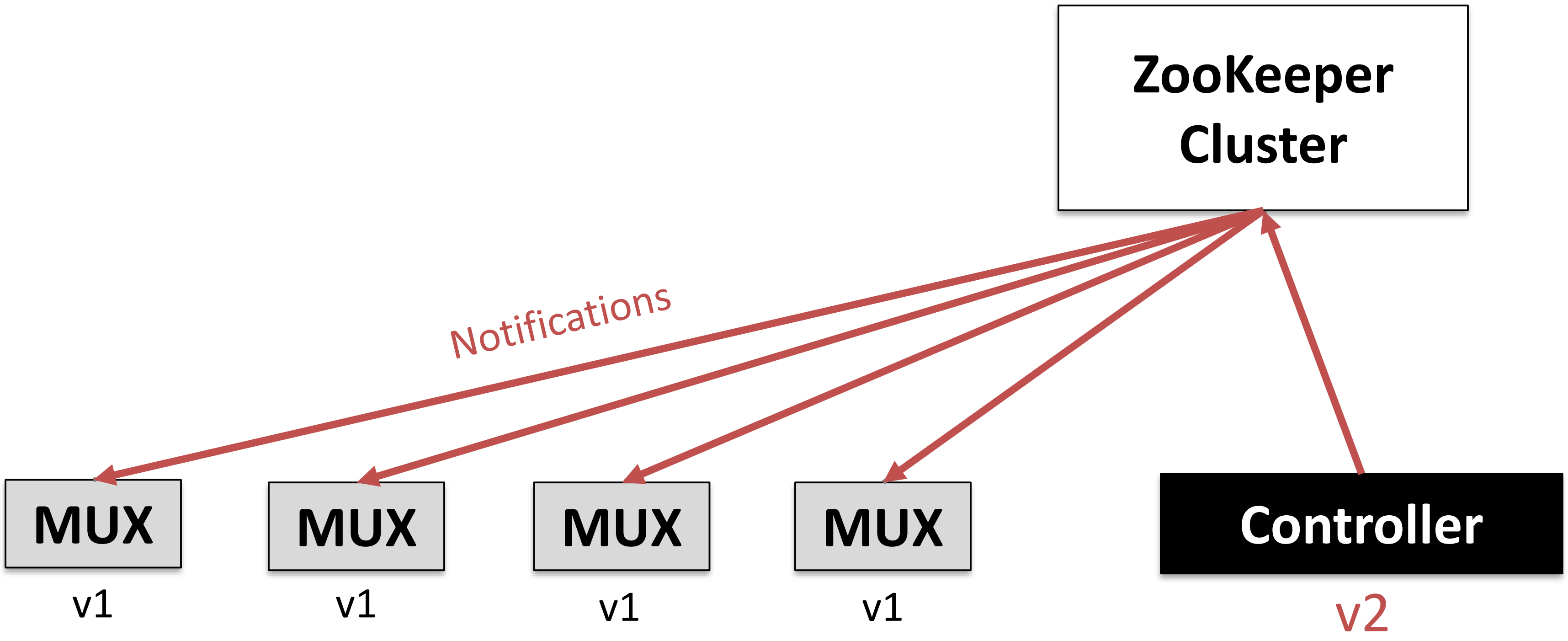
- A centralized fault-tolerant **controller** manages the dataplane

Control plane

DIP	PDIP	TS
D1	D3	TS1
D2	D4	TS2
…	…	…

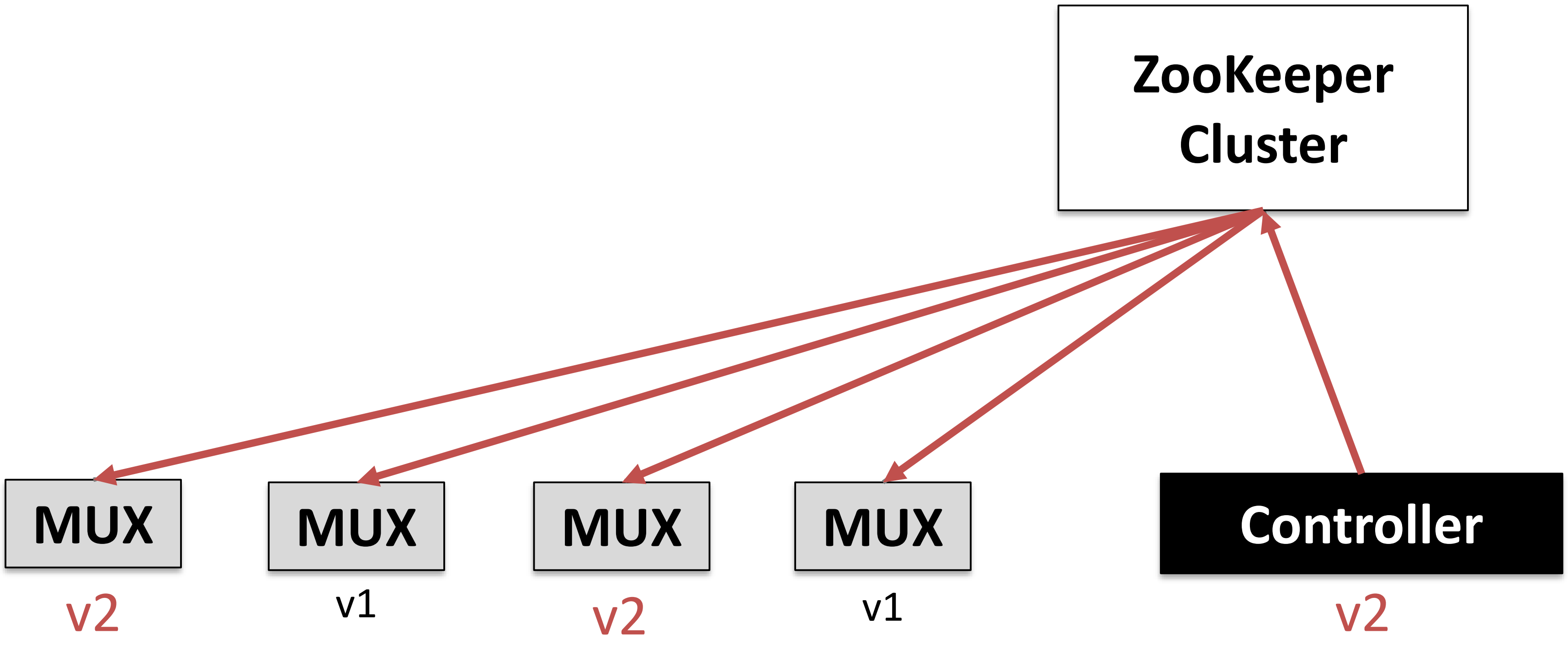


Control plane



DIP	PDIP	TS
D4	D1	NOW
D2	D4	TS2
…	…	…

Control plane



- Muxes download update
- Daisy chaining allows for temporarily stale muxes

Control plane experiments

- Tested on Amazon EC2
- 3 ZooKeeper daemons, 100 muxes
- Large simulated service: 64K servers, 6.4M buckets
- Stress-tested controller

Control plane experiments

When adding 32.000 servers:

- Controller takes 1-10s to update ZooKeeper
- Muxes take 0.5-6s to get new dataplane information
- Total control traffic: 1GB (10MB/mux)

Please see paper for:

- MPTCP support in Beamer

- Minimizing # of rules required in muxes
  – 1 rule / server, rather than 1 rule / bucket

- Avoiding reset connections in corner cases

Conclusions

- Stateless load balancing using daisy chaining
- 36Gbps of HTTP traffic on 7 cores
  – 540Gbps of downlink traffic
- Scalable, fault tolerant control plane
- Beamer is open-source: https://github.com/Beamer-LB