Hashing on broken assumptions

Lorenzo Saino (@lorenzosaino)
Fastly

Problem:

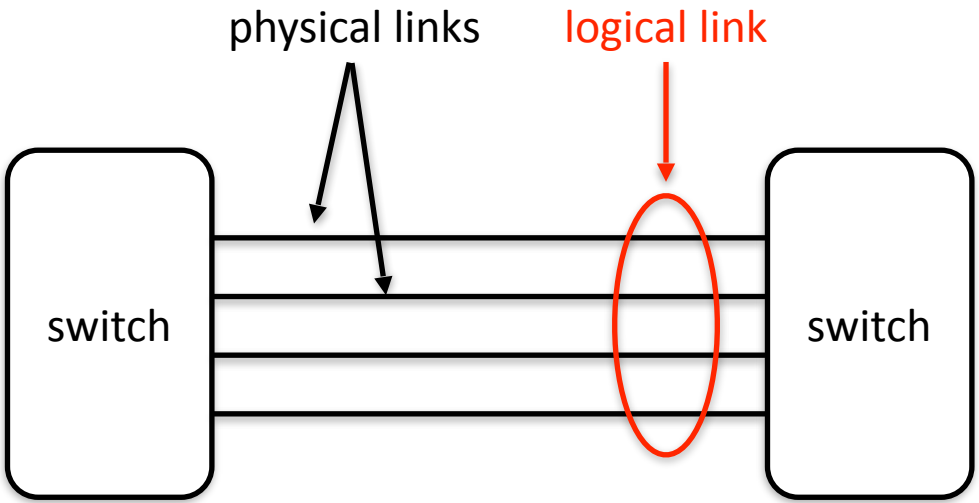
Spreading traffic across multiple links, paths, hosts

Solutions:

- Link Aggregation
- Equal Cost Multipath (ECMP)

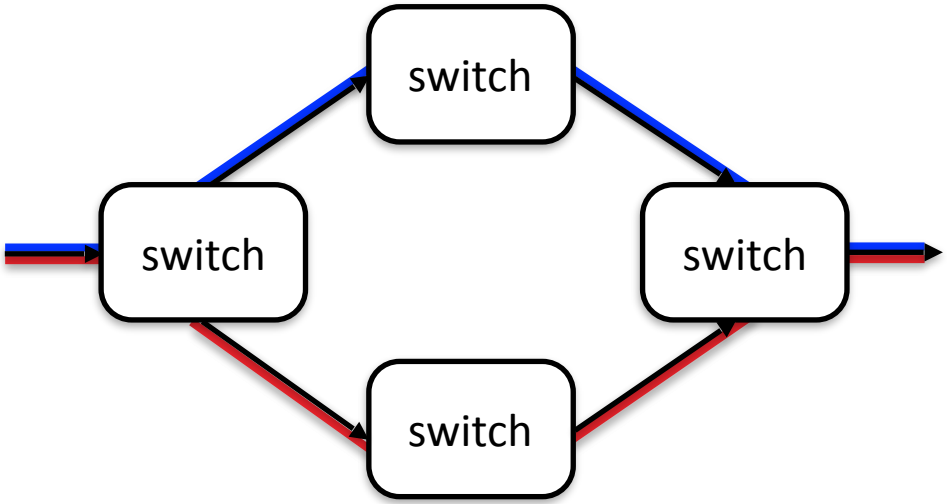
Link aggregation

Combine multiple physical links between network devices into one logical link

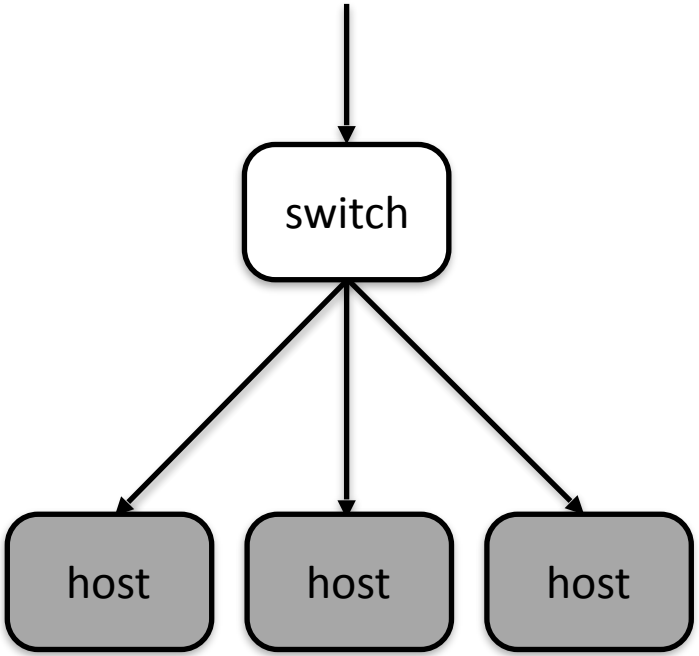


Equal Cost Multipath (ECMP)

Balance traffic across paths



Balance traffic across hosts



Requirements

Load balance

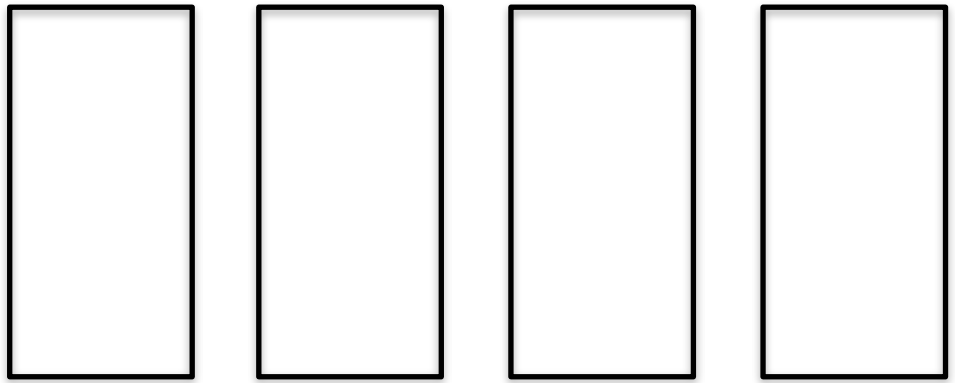
Traffic must be uniformly spread across next-hops

Stateless-but-sticky path pinning

All packets of a flow must take the same path

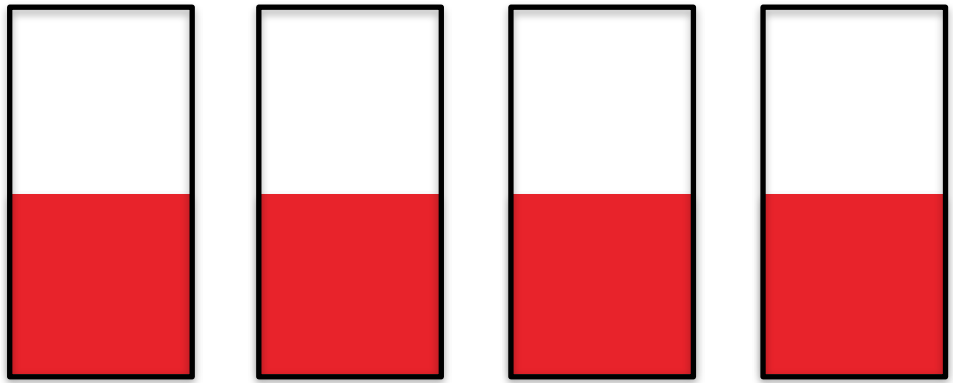
Load imbalance

Load imbalance reduces system capacity



Load imbalance

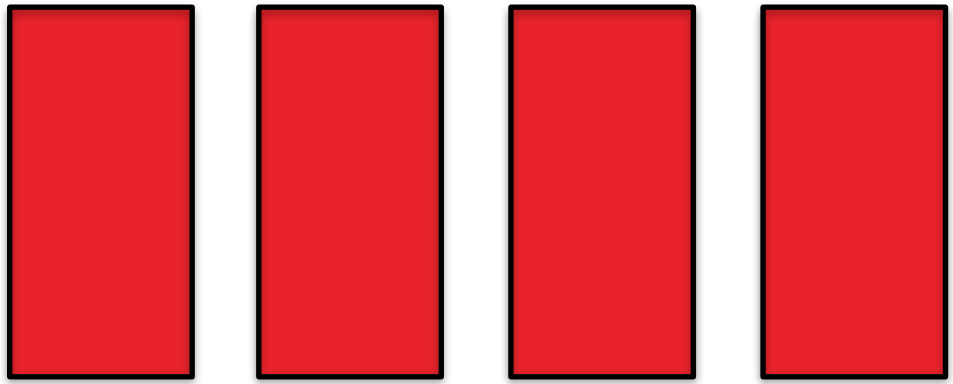
Load imbalance reduces system capacity



Perfect load balance

Load imbalance

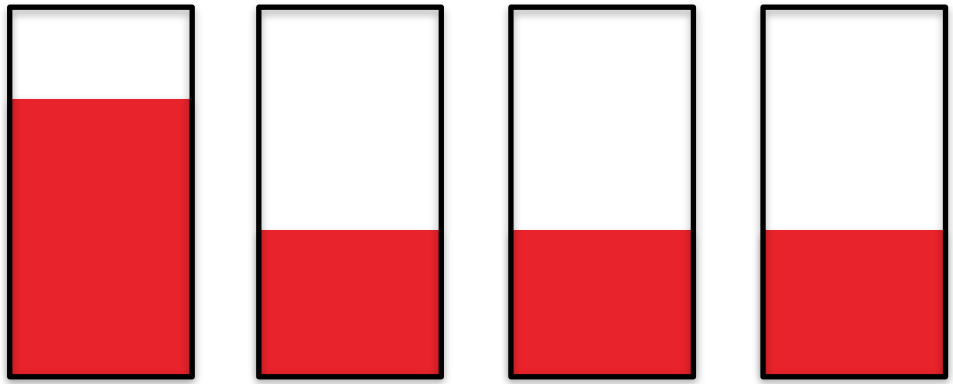
Load imbalance reduces system capacity



All resources fully utilized

Load imbalance

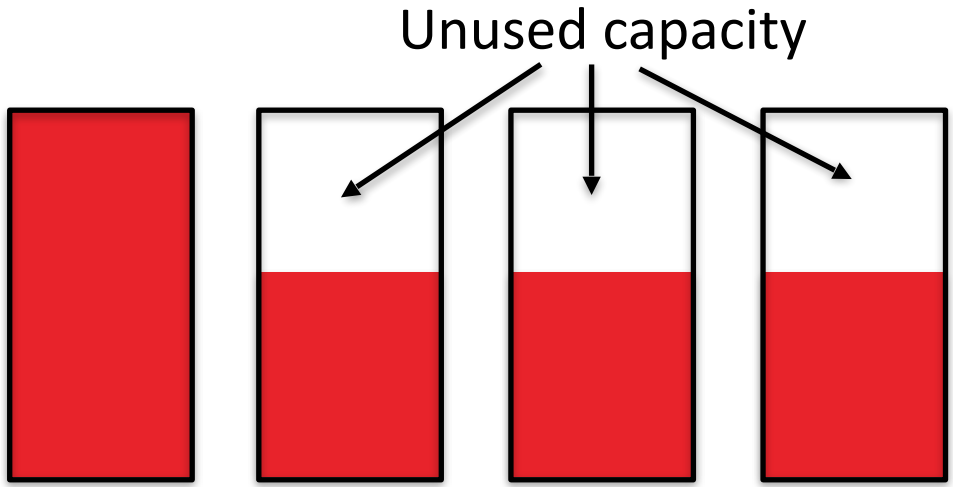
Load imbalance reduces system capacity



Load imbalance

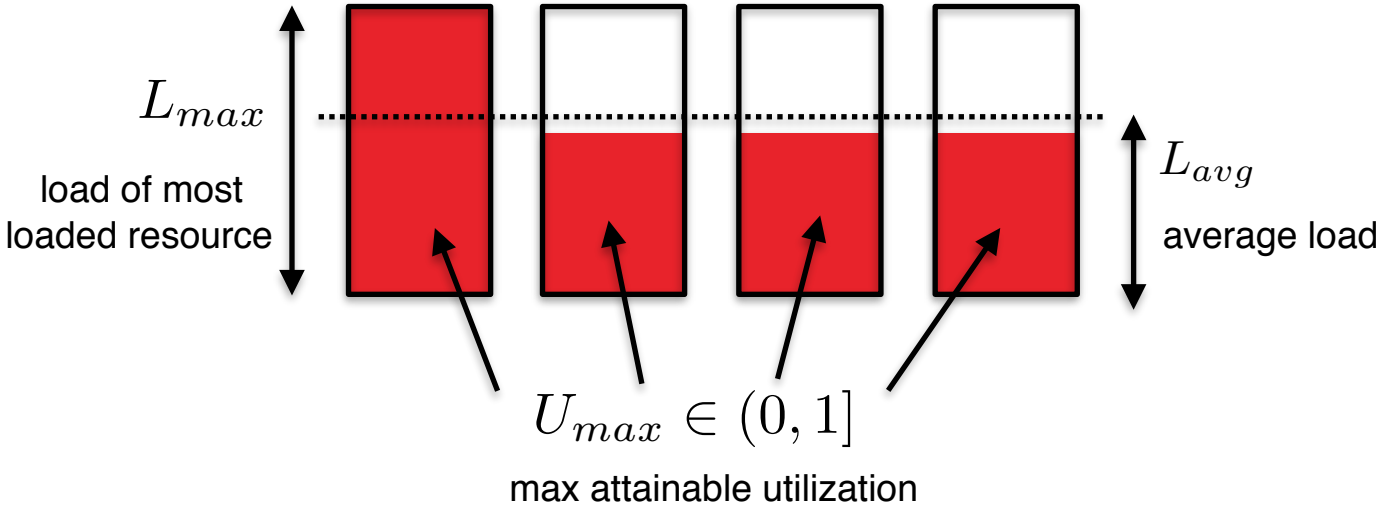
Load imbalance

Load imbalance reduces system capacity



Cannot take any additional load

Quantifying impact of load imbalance



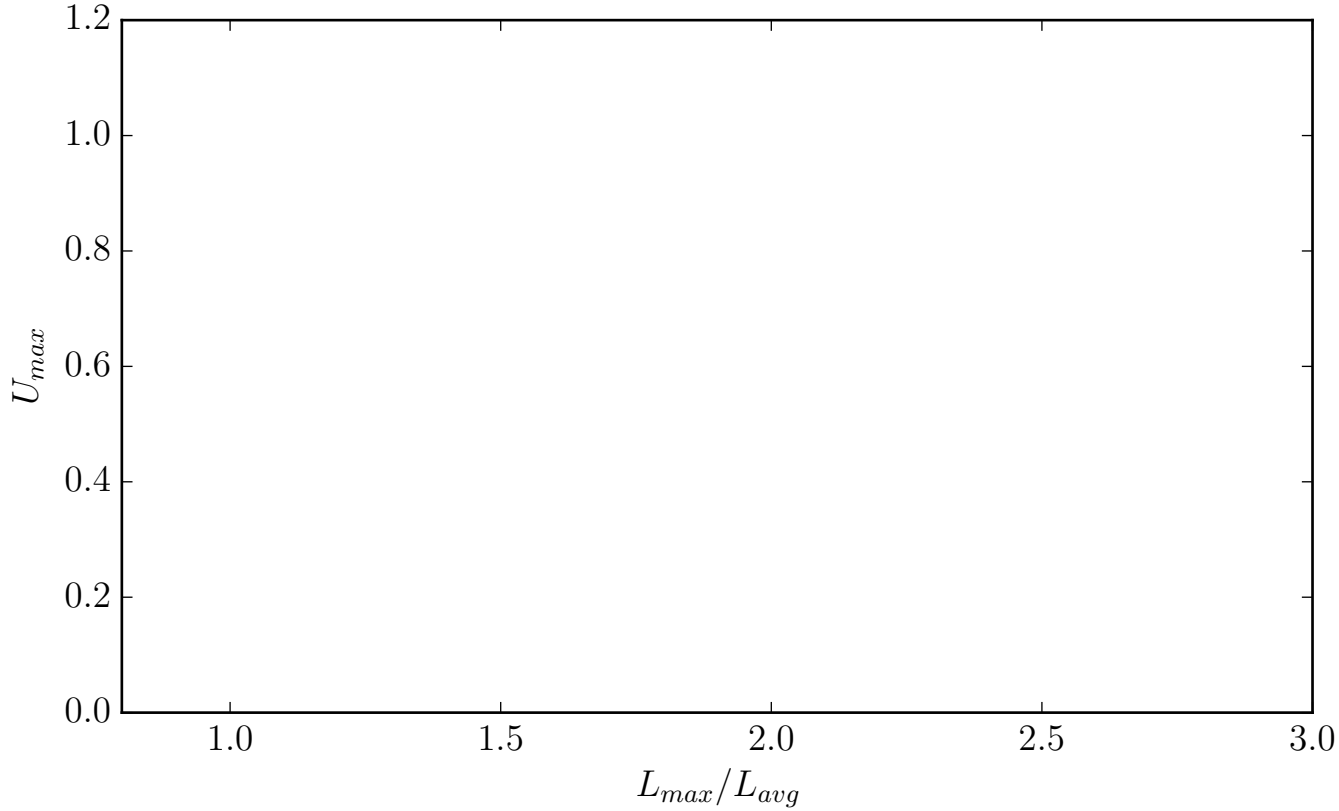
Load imbalance:

$$\frac{L_{max}}{L_{avg}} = [1, +\infty)$$

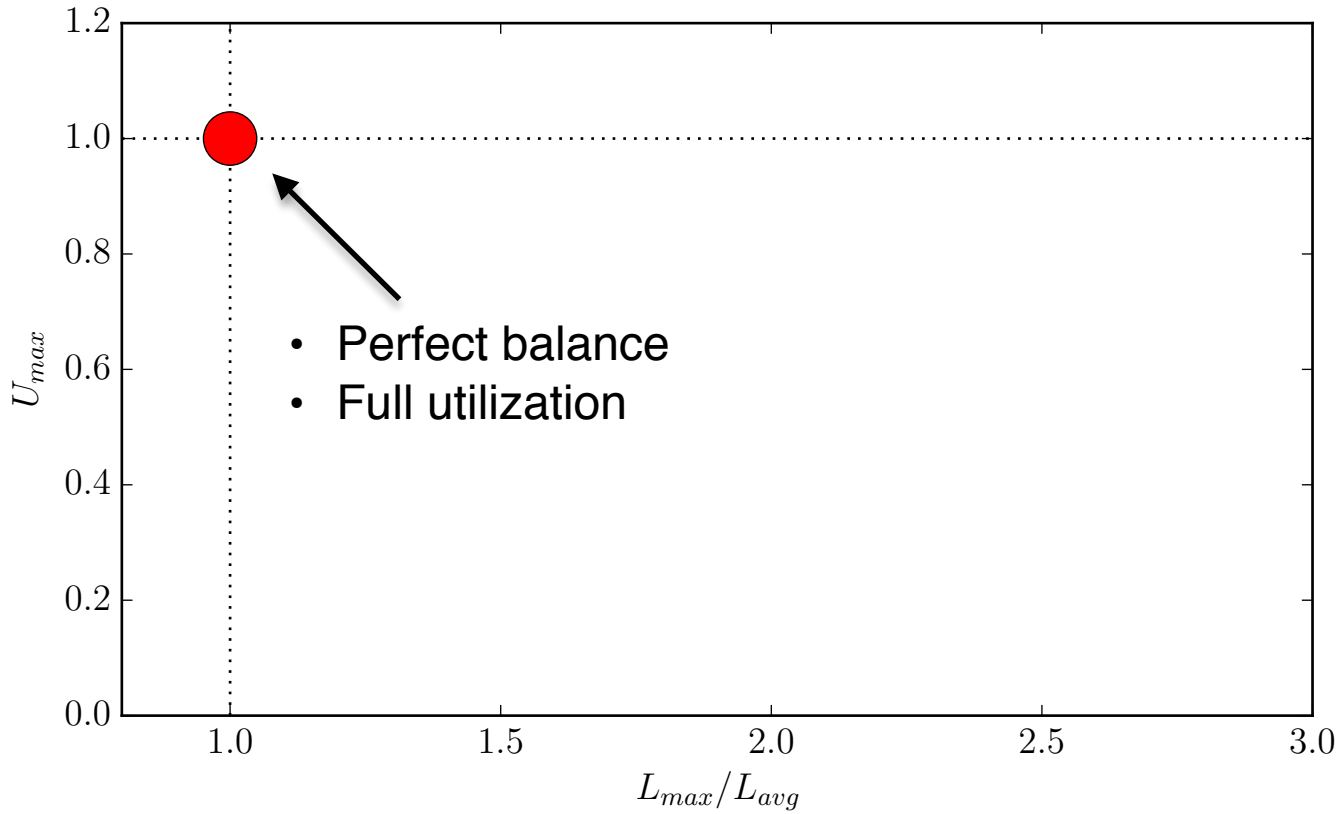
Max attainable utilization:

$$U_{max} = \left(\frac{L_{max}}{L_{avg}}\right)^{-1} = \frac{L_{avg}}{L_{max}}$$

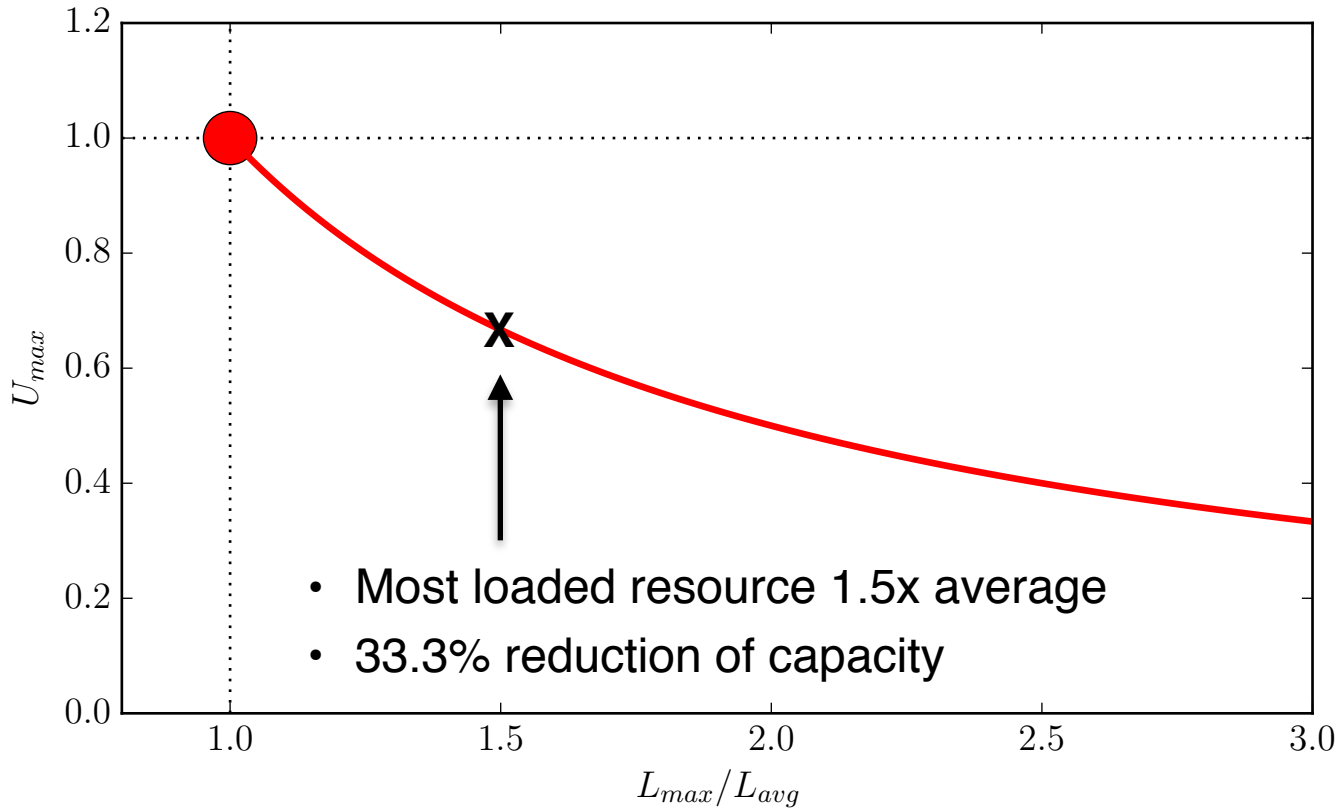
Quantifying impact of load imbalance



Quantifying impact of load imbalance



Quantifying impact of load imbalance



What happens without path pinning?

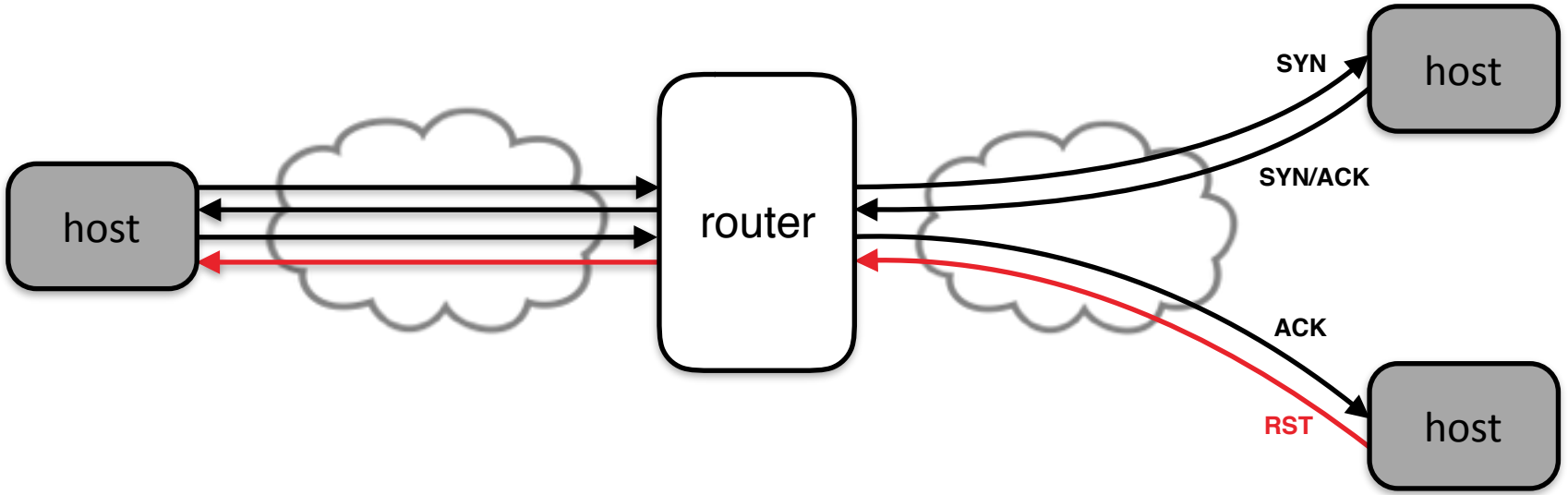
Same endpoints, different paths:

- Out-of-order packets
- Frequent drops of TCP congestion window (CWND)
- Poor throughput performance

Different endpoints:

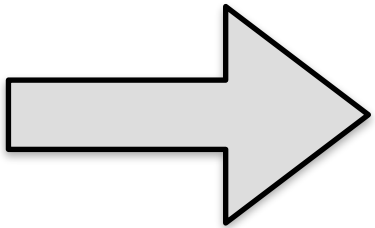
- TCP resets

TCP resets



Requirements:

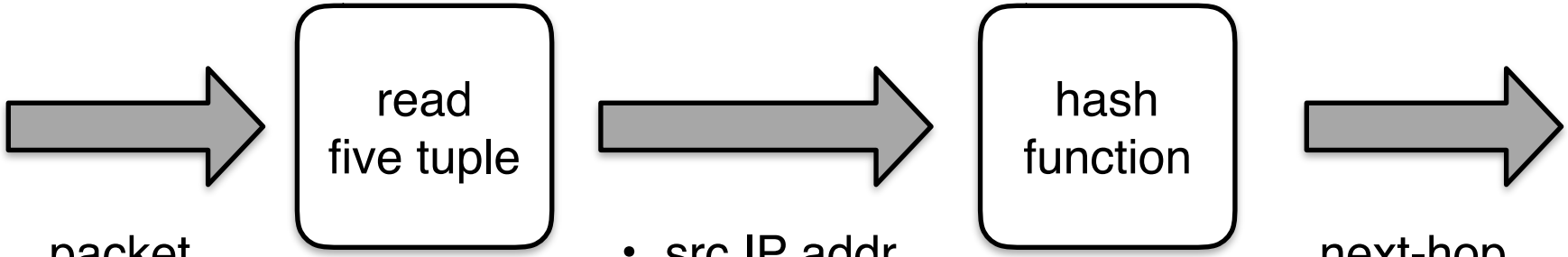
- Load balance
- Path pinning



Solution:

Flow-level hashing

Flow-level hashing



- src IP addr
- dst IP addr
- protocol
- src port
- dst port

Assumptions

Load balance

Hashing uniformly spread traffic across next-hops

Path pinning

Hashing pins packets of a flow to the same path

Do these assumptions hold?

Assumptions

Load balance

Hashing uniformly spread traffic across next-hops

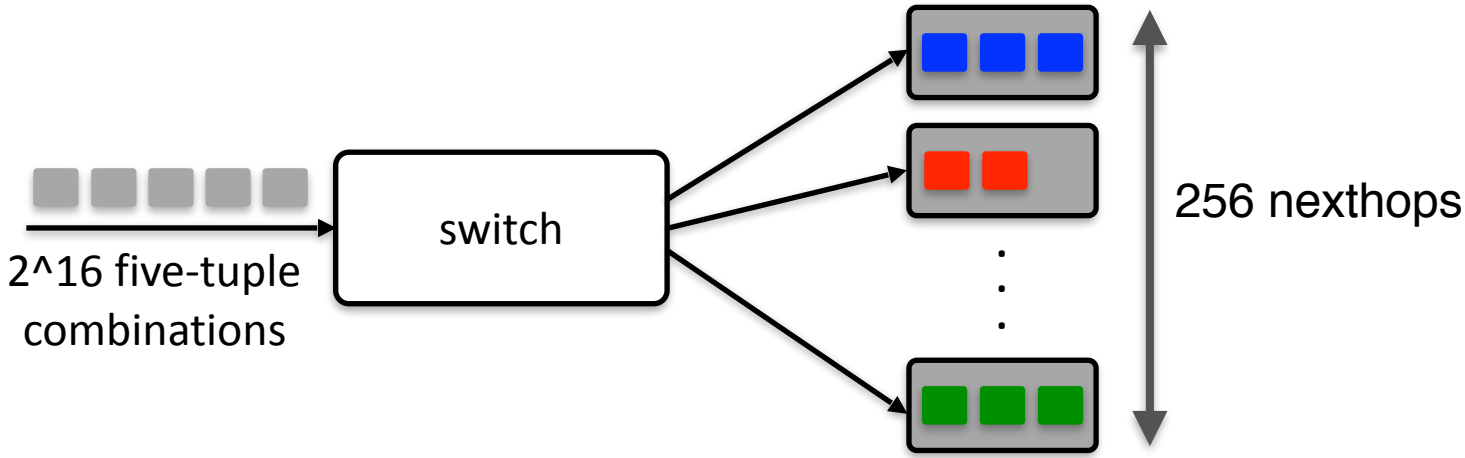
Path pinning

Hashing pins packets of a flow to the same path

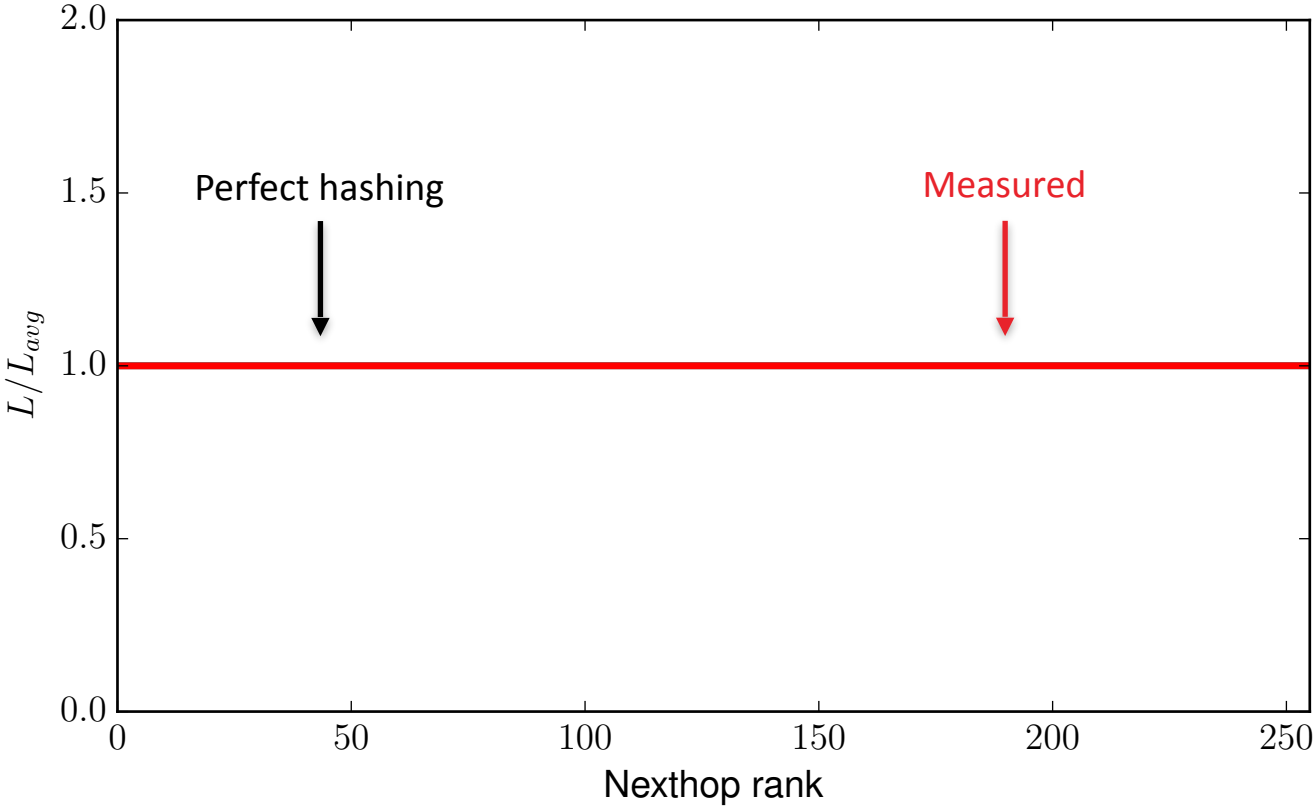
Hashing quality

Two switch models:

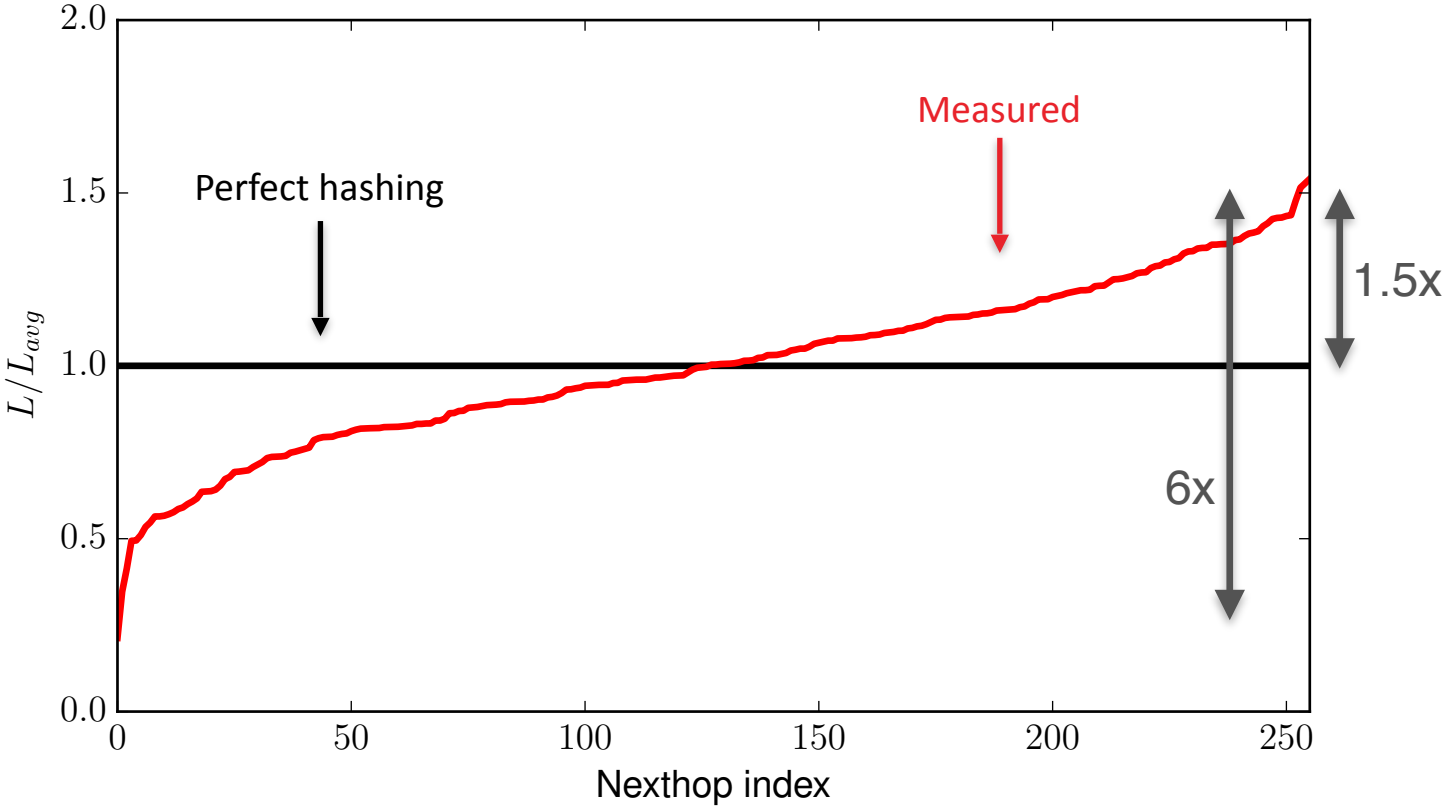
- Switch A
- Switch B



Switch A


2.0
1.5
1.0
0.5
0.0
L/L_{avg}
Perfect hashing
Measured
0
50
100
150
200
250
Nexthop rank

Switch B


2.0
1.5
1.0
0.5
0.0
L/L_{avg}
Perfect hashing
Measured
1.5x
6x
0
50
100
150
200
250
Nexthop index

Switch B

Vendor claims supporting an arbitrary number of next-hops [1, 256]

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	49	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Switch B

Only a subset of next-hops are actually supported

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	18		20		22		24		26		28		30		32
	34		36		38		40				44				48
			52				56				60				64
							72								80
							88								96
							104								112
							120								128
															…

Switch B

Only a subset of next-hops are actually supported

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	18		20		22		24		26		28		30		32
	34		36		38		40				44				48
			52				56				60				64
							72								80
							88								96
							104								112
							120						**126**		128
															…

Switch B

Only a subset of next-hops are actually supported

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	18		20		22		24		26		28		30		32
	34		36		38		40				44				48
			52				56				60				64
							72								80
							88								96
							104								112
							120	X	X	X	X	X	X		128
															...

6 next-hops don't get any traffic

Assumptions

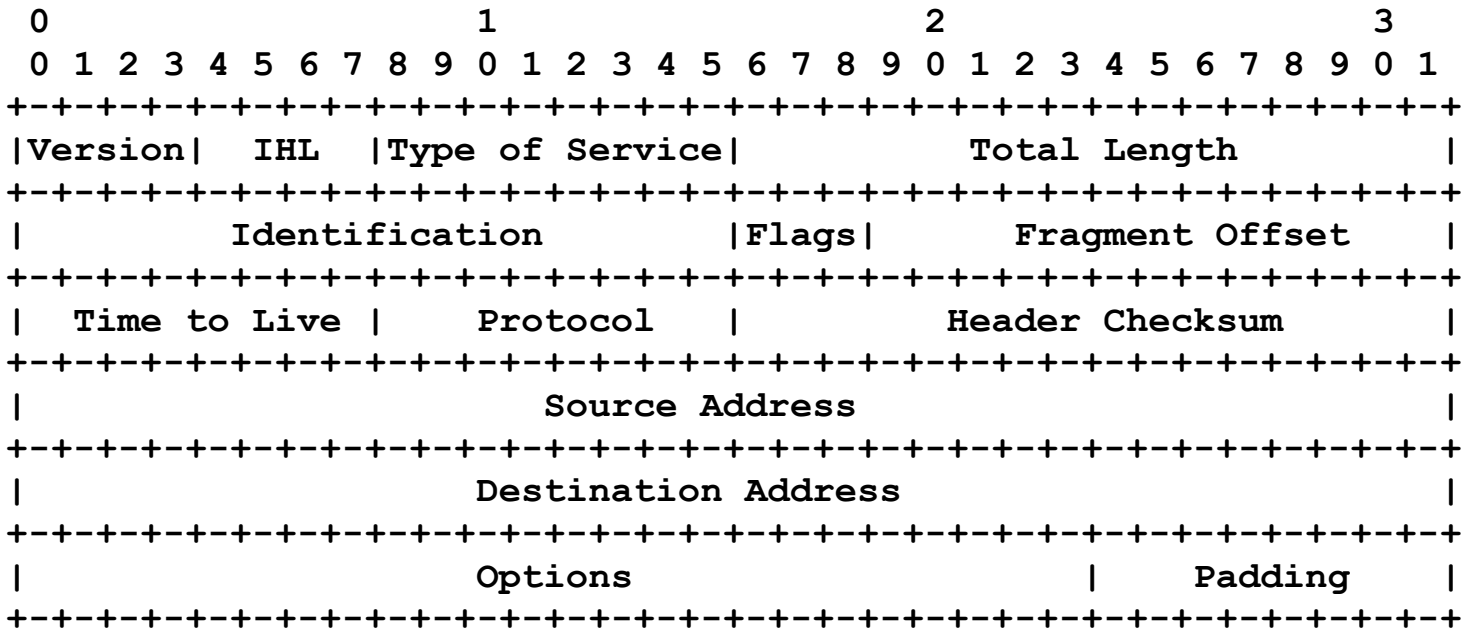
Load balance

Hashing uniformly spread traffic across next-hops

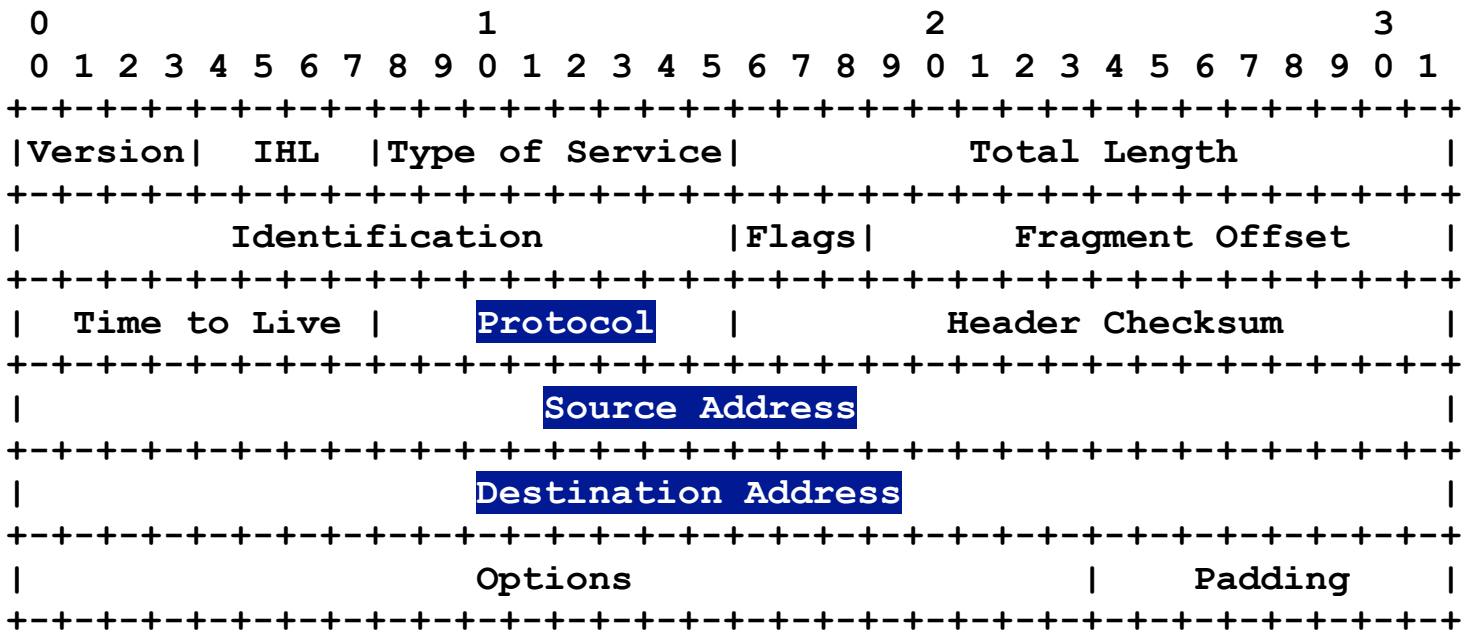
Path pinning

Hashing pins packets of a flow to the same path

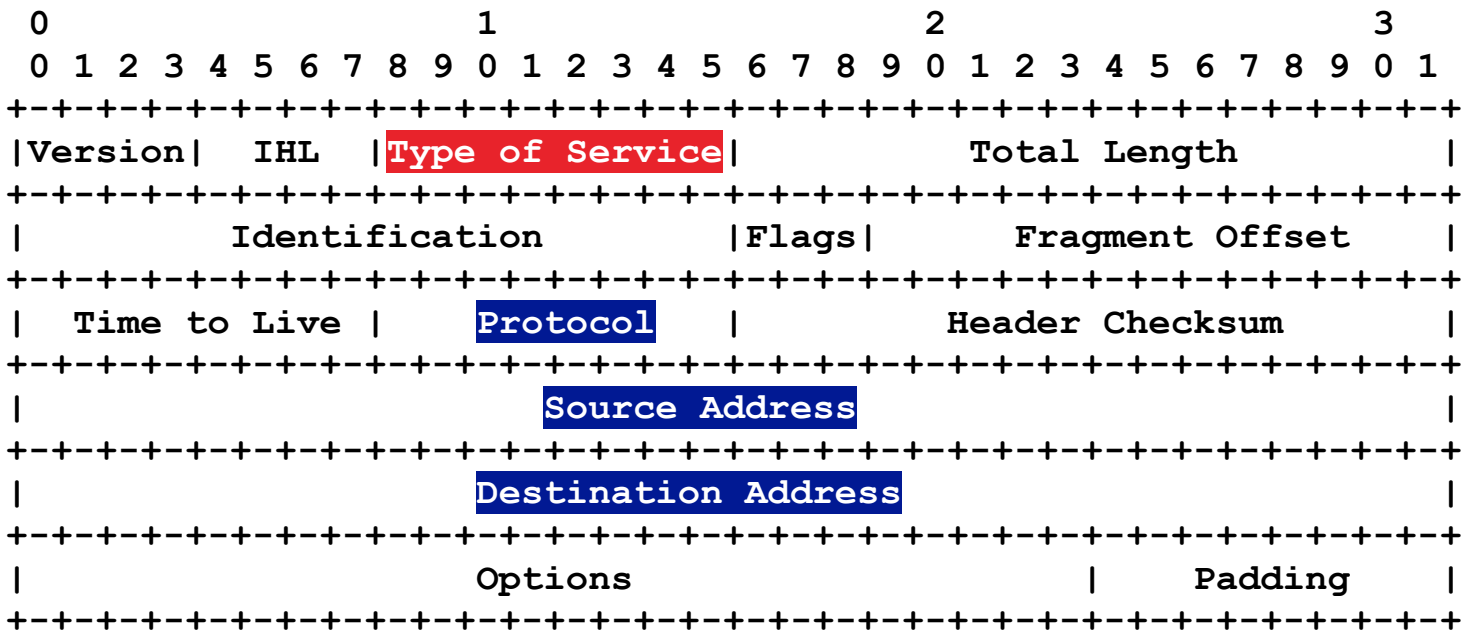
Hashing on IPv4 TOS field



Hashing on IPv4 TOS field



Hashing on IPv4 TOS field



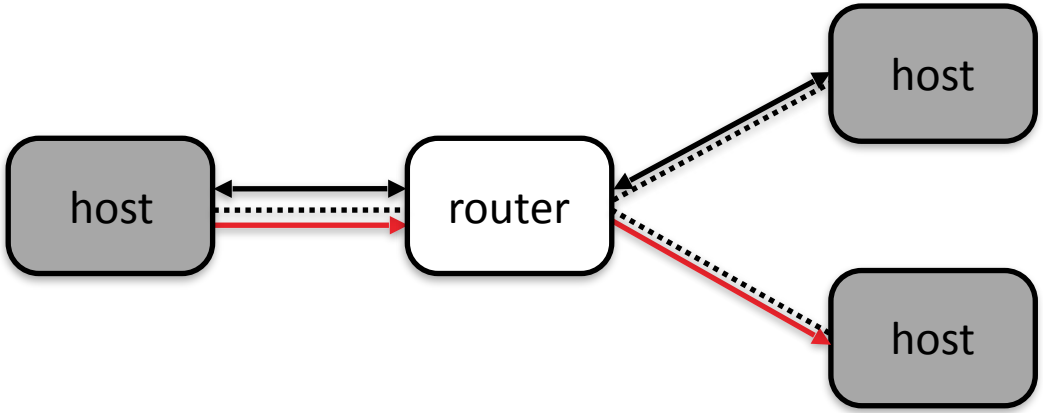
Hashing on IPv4 TOS field

RFC 1812 - Requirements for IP Version 4 Routers
explicitly permits to involve the second-to-last bit of the TOS/DS octet in routing decisions

RFC 2474 - Definition of the Differentiated Services Field
deprecates the IPv4 Type of Service field redefines it as the Differentiated Services field

RFC 3168 - The Addition of Explicit Congestion Notification (ECN) to IP
reserves the last two bits of the DS octet for ECN

Hashing on IPv4 TOS field



TCP handshake:

- Hosts negotiate ECN support
- ECN-capable bits unset

Flow data:

- ECN-capable bits set

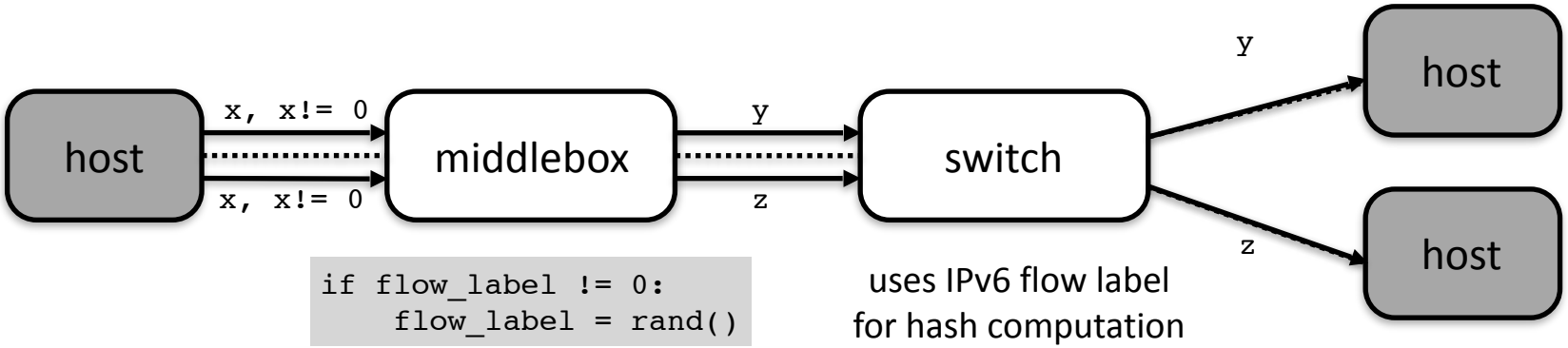
Scenario

- Hosts are ECN capable
- Router uses IPv4 TOS for hash computation (RFC 1812)

TCP handshake

flow data

IPv6 flow label rewrite



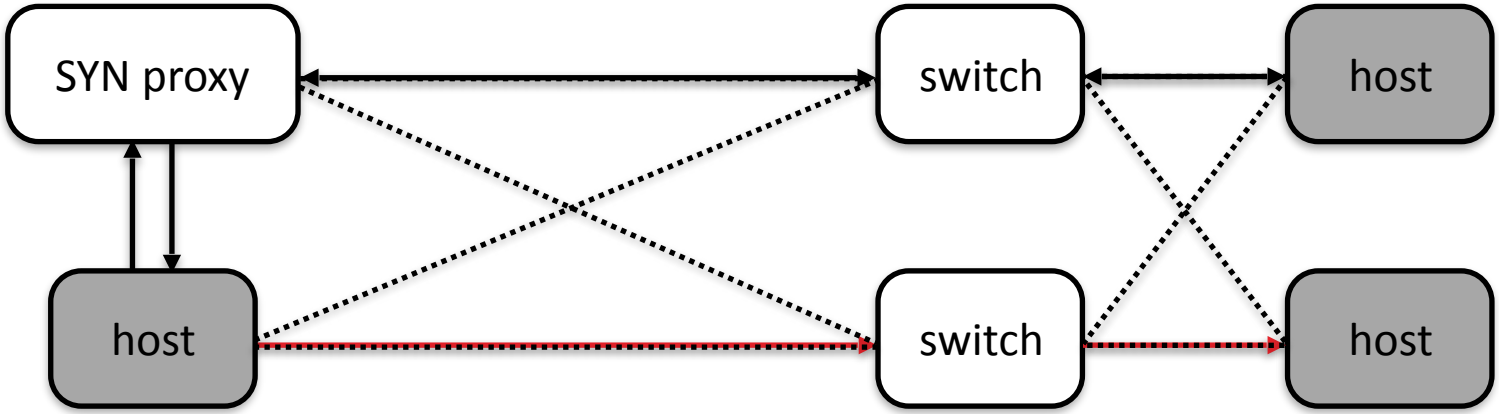

```
if flow_label != 0:
    flow_label = rand()
```

forbidden by RFC 6437

uses IPv6 flow label
for hash computation

allowed by RFC 6437

SYN proxies



Switches:

- use ingress interface for hash computation, or
- use different hash function seeds

TCP handshake

flow data

Conclusions

Load balancing

There are devices that do not hash traffic uniformly

Path pinning

Hashing on fields other than five tuples breaks ECMP

- Ingress port
- IPv4 TOS
- IPv6 flow label

Recommendations

Operators:

- Ensure that your network devices hash flows uniformly or that could cost you money
- Disable additional inputs if you do not need extra entropy

Vendors:

- Disable hashing inputs other than five-tuple by default
- Make hash input fields configurable
- Make hash seed configurable

FIN