P4air: Increasing Fairness among Competing Congestion Control Algorithms

Belma Turkovic and Fernando Kuipers

ICNP 2020, October 13-16, 2020

Congestion control algorithms

Congestion control algorithms

- New protocols and congestion control algorithms are continuously being developed

Congestion control algorithms

- New protocols and congestion control algorithms are continuously being developed

→ It is impossible to take their interactions with other protocols and algorithms into account

Why is this important?

Why is this important?

	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
HS-TCP	0.98	0.75	0.92	0.95	0.88	0.94	0.72	0.73	0.76	0.66	0.70	0.60	0.53	0.58
STCP	0.75	0.99	0.80	0.83	0.83	0.81	0.77	0.78	0.83	0.71	0.70	0.58	0.53	0.57
HTCP	0.92	0.80	0.99	0.84	0.96	0.99	0.81	0.88	0.88	0.78	0.86	0.57	0.52	0.56
BIC	0.95	0.83	0.84	0.98	0.80	0.85	0.66	0.68	0.66	0.61	0.67	0.59	0.53	0.67
Cubic	0.88	0.83	0.96	0.80	0.99	0.97	0.87	0.89	0.88	0.82	0.88	0.58	0.53	0.56
New Reno	0.94	0.81	0.99	0.85	0.97	0.99	0.83	0.88	0.89	0.78	0.87	0.57	0.53	0.55
Hybla	0.72	0.77	0.81	0.66	0.87	0.83	0.99	0.96	0.98	0.92	0.97	0.58	0.52	0.56
YeAH	0.73	0.78	0.88	0.68	0.89	0.88	0.96	0.99	0.98	0.92	0.97	0.62	0.52	0.56
Illinois	0.76	0.83	0.88	0.66	0.88	0.89	0.98	0.98	0.99	0.92	0.95	0.58	0.52	0.54
Veno	0.66	0.71	0.78	0.61	0.82	0.78	0.92	0.92	0.92	0.98	0.93	0.60	0.52	0.54
Westwood+	0.70	0.70	0.86	0.67	0.88	0.87	0.97	0.97	0.95	0.93	1.00	0.58	0.52	0.54
BBR	0.60	0.58	0.57	0.59	0.58	0.57	0.58	0.62	0.58	0.60	0.58	0.94	0.65	0.79
Vegas	0.53	0.53	0.52	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.65	1.00	0.67
LoLa	0.58	0.57	0.56	0.67	0.56	0.55	0.56	0.56	0.54	0.54	0.54	0.79	0.67	0.80

Why is this important?

	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
HS-TCP	0.98	0.75	0.92	0.95	0.88	0.94	0.72	0.73	0.76	0.66	0.70	0.60	0.53	0.58
STCP	0.75	0.99	0.80	0.83	0.83	0.81	0.77	0.78	0.83	0.71	0.70	0.58	0.53	0.57
HTCP	0.92	0.80	0.99	0.84	0.96	0.99	0.81	0.88	0.88	0.78	0.86	0.57	0.52	0.56
BIC	0.95	0.83	0.84	0.98	0.80	0.85	0.66	0.68	0.66	0.61	0.67	0.59	0.53	0.67
Cubic	0.88	0.83	0.96	0.80	0.99	0.97	0.87	0.89	0.88	0.82	0.88	0.58	0.53	0.56
New Reno	0.94	0.81	0.99	0.85	0.97	0.99	0.83	0.88	0.89	0.78	0.87	0.57	0.53	0.55
Hybla	0.72	0.77	0.81	0.66	0.87	0.83	0.99	0.96	0.98	0.92	0.97	0.58	0.52	0.56
YeAH	0.73	0.78	0.88	0.68	0.89	0.88	0.96	0.99	0.98	0.92	0.97	0.62	0.52	0.56
Illinois	0.76	0.83	0.88	0.66	0.88	0.89	0.98	0.98	0.99	0.92	0.95	0.58	0.52	0.54
Veno	0.66	0.71	0.78	0.61	0.82	0.78	0.92	0.92	0.92	0.98	0.93	0.60	0.52	0.54
Westwood+	0.70	0.70	0.86	0.67	0.88	0.87	0.97	0.97	0.95	0.93	1.00	0.58	0.52	0.54
BBR	0.60	0.58	0.57	0.59	0.58	0.57	0.58	0.62	0.58	0.60	0.58	0.94	0.65	0.79
Vegas	0.53	0.53	0.52	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.65	1.00	0.67
LoLa	0.58	0.57	0.56	0.67	0.56	0.55	0.56	0.56	0.54	0.54	0.54	0.79	0.67	0.80

Why is this important?

	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
0 ms	0.98	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.98	1.00	0.94	1.00	0.80
20 ms	0.79	0.92	0.94	0.74	0.84	0.85	0.89	0.86	0.92	0.91	0.86	0.56	0.83	0.73
40 ms	0.70	0.83	0.89	0.68	0.80	0.74	0.89	0.78	0.82	0.85	0.77	0.54	0.82	0.59
60 ms	0.67	0.79	0.88	0.66	0.72	0.69	0.94	0.74	0.77	0.83	0.71	0.55	0.78	0.59
80 ms	0.62	0.73	0.87	0.63	0.75	0.67	0.95	0.74	0.74	0.80	0.69	0.56	0.82	0.59
100 ms	0.59	0.74	0.84	0.63	0.73	0.66	0.95	0.73	0.80	0.79	0.65	0.56	0.80	0.62
120 ms	0.59	0.68	0.82	0.60	0.82	0.60	0.96	0.74	0.82	0.78	0.63	0.58	0.82	0.56
140 ms	0.57	0.65	0.80	0.59	0.78	0.59	0.95	0.71	0.83	0.76	0.61	0.57	0.85	0.57
160 ms	0.56	0.64	0.79	0.58	0.76	0.60	0.95	0.69	0.83	0.75	0.59	0.58	0.72	0.55
180 ms	0.56	0.63	0.74	0.56	0.78	0.59	0.95	0.82	0.79	0.72	0.59	0.58	0.81	0.55
200 ms	0.54	0.61	0.70	0.55	0.73	0.58	0.95	0.90	0.78	0.74	0.57	0.59	0.77	0.54
220 ms	0.54	0.61	0.69	0.55	0.76	0.59	0.95	0.79	0.64	0.65	0.56	0.58	0.79	0.59
240 ms	0.56	0.55	0.70	0.56	0.71	0.58	0.94	0.82	0.68	0.65	0.55	0.58	0.8	0.59
260 ms	0.55	0.54	0.65	0.56	0.69	0.56	0.94	0.80	0.63	0.56	0.55	0.58	0.73	0.56

Goal

Goal

To improve fairness between all flows present on a switch by grouping them based on their congestion control algorithm

Goal

To improve fairness between all flows present on a switch by grouping them based on their congestion control algorithm

- from within the data-plane

Goal

improve fairness between all flows present on a switch by grouping them based on their congestion control algorithm

- from within the data-plane

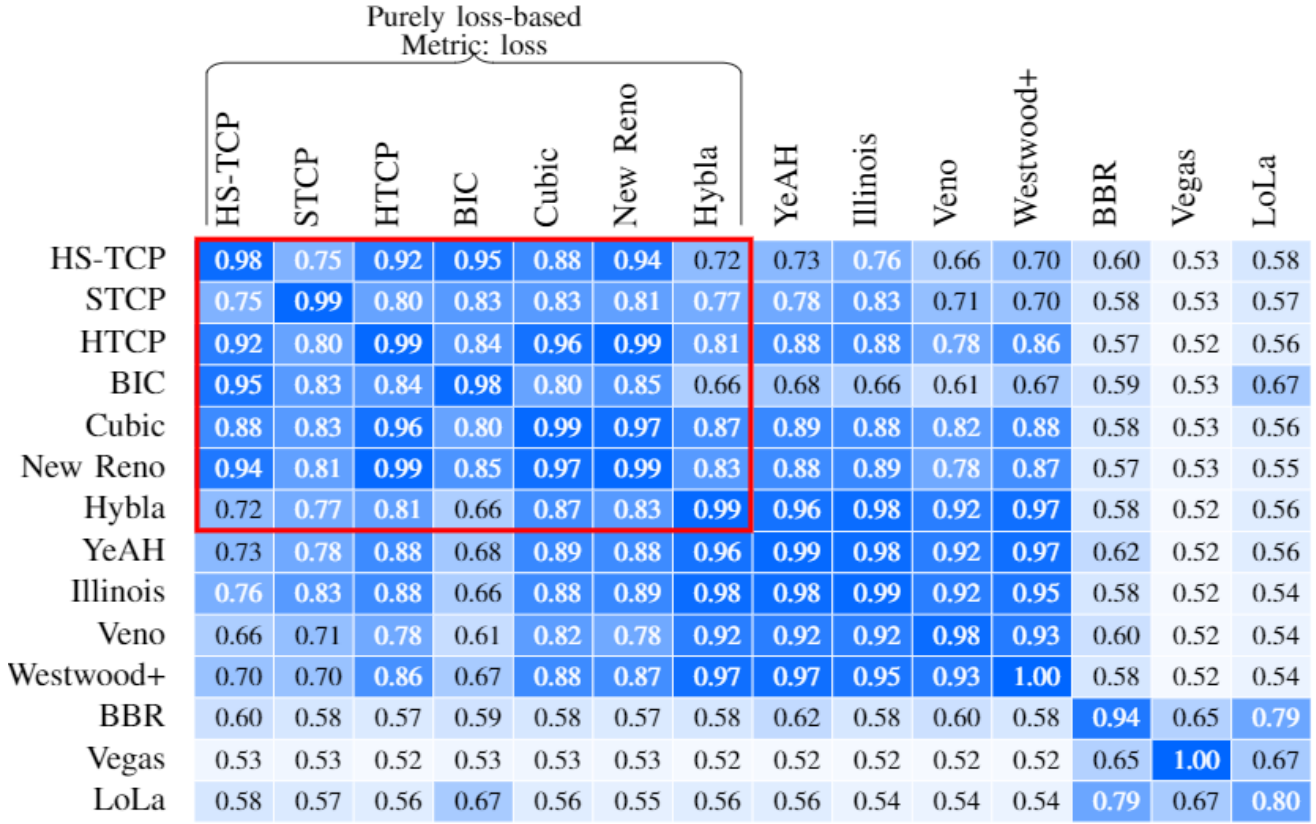
→ and by taking into account limitations on actions and/or memory accesses

Classification

Classification

	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
HS-TCP	0.98	0.75	0.92	0.95	0.88	0.94	0.72	0.73	0.76	0.66	0.70	0.60	0.53	0.58
STCP	0.75	0.99	0.80	0.83	0.83	0.81	0.77	0.78	0.83	0.71	0.70	0.58	0.53	0.57
HTCP	0.92	0.80	0.99	0.84	0.96	0.99	0.81	0.88	0.88	0.78	0.86	0.57	0.52	0.56
BIC	0.95	0.83	0.84	0.98	0.80	0.85	0.66	0.68	0.66	0.61	0.67	0.59	0.53	0.67
Cubic	0.88	0.83	0.96	0.80	0.99	0.97	0.87	0.89	0.88	0.82	0.88	0.58	0.53	0.56
New Reno	0.94	0.81	0.99	0.85	0.97	0.99	0.83	0.88	0.89	0.78	0.87	0.57	0.53	0.55
Hybla	0.72	0.77	0.81	0.66	0.87	0.83	0.99	0.96	0.98	0.92	0.97	0.58	0.52	0.56
YeAH	0.73	0.78	0.88	0.68	0.89	0.88	0.96	0.99	0.98	0.92	0.97	0.62	0.52	0.56
Illinois	0.76	0.83	0.88	0.66	0.88	0.89	0.98	0.98	0.99	0.92	0.95	0.58	0.52	0.54
Veno	0.66	0.71	0.78	0.61	0.82	0.78	0.92	0.92	0.92	0.98	0.93	0.60	0.52	0.54
Westwood+	0.70	0.70	0.86	0.67	0.88	0.87	0.97	0.97	0.95	0.93	1.00	0.58	0.52	0.54
BBR	0.60	0.58	0.57	0.59	0.58	0.57	0.58	0.62	0.58	0.60	0.58	0.94	0.65	0.79
Vegas	0.53	0.53	0.52	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.65	1.00	0.67
LoLa	0.58	0.57	0.56	0.67	0.56	0.55	0.56	0.56	0.54	0.54	0.54	0.79	0.67	0.80

Classification



Classification

	Purely loss-based Metric: loss												Delay-based Metric: delay	
	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
HS-TCP	0.98	0.75	0.92	0.95	0.88	0.94	0.72	0.73	0.76	0.66	0.70	0.60	0.53	0.58
STCP	0.75	0.99	0.80	0.83	0.83	0.81	0.77	0.78	0.83	0.71	0.70	0.58	0.53	0.57
HTCP	0.92	0.80	0.99	0.84	0.96	0.99	0.81	0.88	0.88	0.78	0.86	0.57	0.52	0.56
BIC	0.95	0.83	0.84	0.98	0.80	0.85	0.66	0.68	0.66	0.61	0.67	0.59	0.53	0.67
Cubic	0.88	0.83	0.96	0.80	0.99	0.97	0.87	0.89	0.88	0.82	0.88	0.58	0.53	0.56
New Reno	0.94	0.81	0.99	0.85	0.97	0.99	0.83	0.88	0.89	0.78	0.87	0.57	0.53	0.55
Hybla	0.72	0.77	0.81	0.66	0.87	0.83	0.99	0.96	0.98	0.92	0.97	0.58	0.52	0.56
YeAH	0.73	0.78	0.88	0.68	0.89	0.88	0.96	0.99	0.98	0.92	0.97	0.62	0.52	0.56
Illinois	0.76	0.83	0.88	0.66	0.88	0.89	0.98	0.98	0.99	0.92	0.95	0.58	0.52	0.54
Veno	0.66	0.71	0.78	0.61	0.82	0.78	0.92	0.92	0.92	0.98	0.93	0.60	0.52	0.54
Westwood+	0.70	0.70	0.86	0.67	0.88	0.87	0.97	0.97	0.95	0.93	1.00	0.58	0.52	0.54
BBR	0.60	0.58	0.57	0.59	0.58	0.57	0.58	0.62	0.58	0.60	0.58	0.94	0.65	0.79
Vegas	0.53	0.53	0.52	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.65	1.00	0.67
LoLa	0.58	0.57	0.56	0.67	0.56	0.55	0.56	0.56	0.54	0.54	0.54	0.79	0.67	0.80

Classification

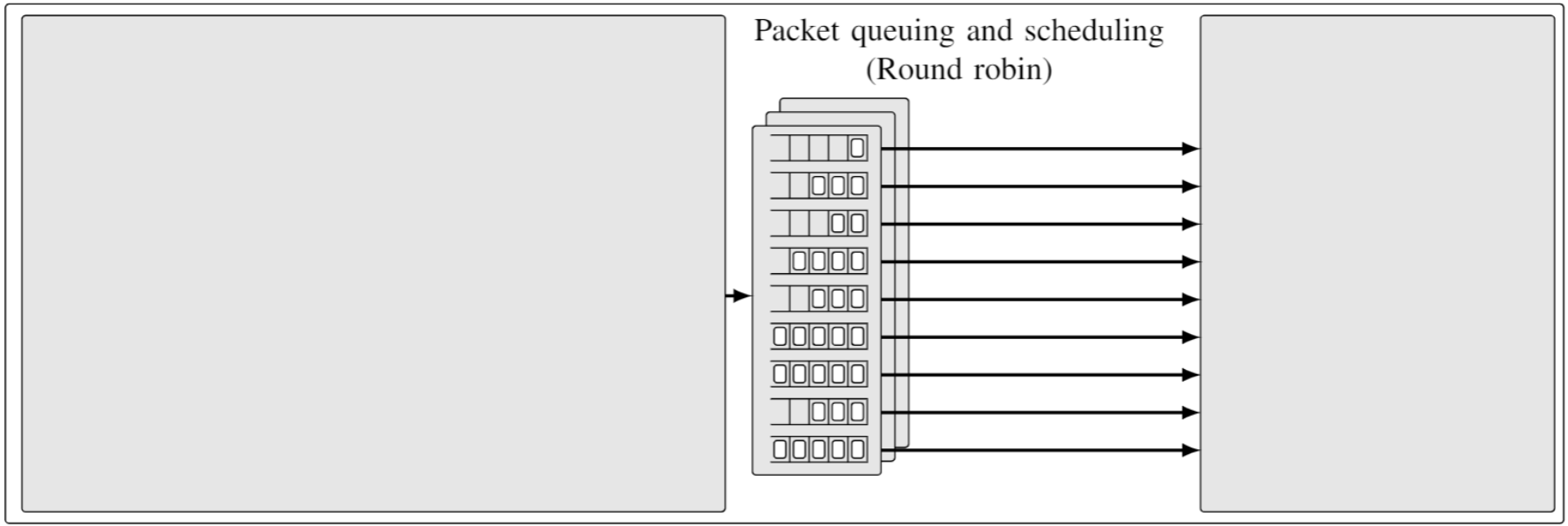
	Purely loss-based Metric: loss							Loss-delay Metric: loss, delay					Delay-based Metric: delay	
	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
HS-TCP	0.98	0.75	0.92	0.95	0.88	0.94	0.72	0.73	0.76	0.66	0.70	0.60	0.53	0.58
STCP	0.75	0.99	0.80	0.83	0.83	0.81	0.77	0.78	0.83	0.71	0.70	0.58	0.53	0.57
HTCP	0.92	0.80	0.99	0.84	0.96	0.99	0.81	0.88	0.88	0.78	0.86	0.57	0.52	0.56
BIC	0.95	0.83	0.84	0.98	0.80	0.85	0.66	0.68	0.66	0.61	0.67	0.59	0.53	0.67
Cubic	0.88	0.83	0.96	0.80	0.99	0.97	0.87	0.89	0.88	0.82	0.88	0.58	0.53	0.56
New Reno	0.94	0.81	0.99	0.85	0.97	0.99	0.83	0.88	0.89	0.78	0.87	0.57	0.53	0.55
Hybla	0.72	0.77	0.81	0.66	0.87	0.83	0.99	0.96	0.98	0.92	0.97	0.58	0.52	0.56
YeAH	0.73	0.78	0.88	0.68	0.89	0.88	0.96	0.99	0.98	0.92	0.97	0.62	0.52	0.56
Illinois	0.76	0.83	0.88	0.66	0.88	0.89	0.98	0.98	0.99	0.92	0.95	0.58	0.52	0.54
Veno	0.66	0.71	0.78	0.61	0.82	0.78	0.92	0.92	0.92	0.98	0.93	0.60	0.52	0.54
Westwood+	0.70	0.70	0.86	0.67	0.88	0.87	0.97	0.97	0.95	0.93	1.00	0.58	0.52	0.54
BBR	0.60	0.58	0.57	0.59	0.58	0.57	0.58	0.62	0.58	0.60	0.58	0.94	0.65	0.79
Vegas	0.53	0.53	0.52	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.65	1.00	0.67
LoLa	0.58	0.57	0.56	0.67	0.56	0.55	0.56	0.56	0.54	0.54	0.54	0.79	0.67	0.80

Classification

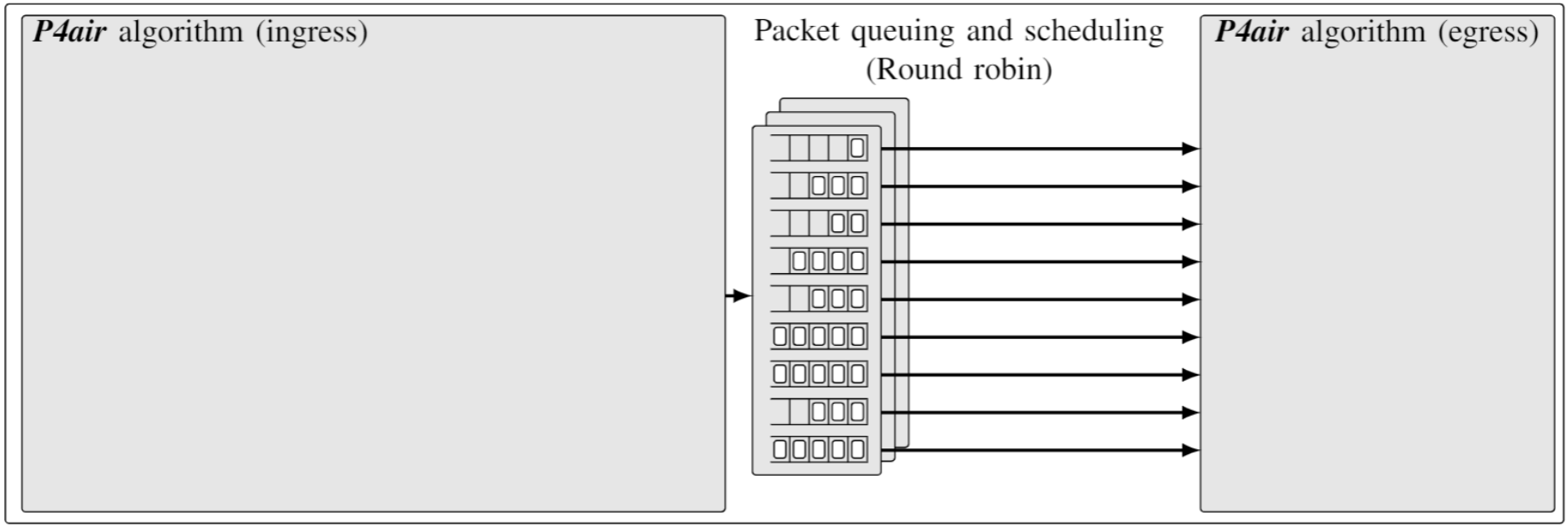
	Purely loss-based Metric: loss							Loss-delay Metric: loss, delay				Model-based	Delay-based Metric: delay	
	HS-TCP	STCP	HTCP	BIC	Cubic	New Reno	Hybla	YeAH	Illinois	Veno	Westwood+	BBR	Vegas	LoLa
HS-TCP	0.98	0.75	0.92	0.95	0.88	0.94	0.72	0.73	0.76	0.66	0.70	0.60	0.53	0.58
STCP	0.75	0.99	0.80	0.83	0.83	0.81	0.77	0.78	0.83	0.71	0.70	0.58	0.53	0.57
HTCP	0.92	0.80	0.99	0.84	0.96	0.99	0.81	0.88	0.88	0.78	0.86	0.57	0.52	0.56
BIC	0.95	0.83	0.84	0.98	0.80	0.85	0.66	0.68	0.66	0.61	0.67	0.59	0.53	0.67
Cubic	0.88	0.83	0.96	0.80	0.99	0.97	0.87	0.89	0.88	0.82	0.88	0.58	0.53	0.56
New Reno	0.94	0.81	0.99	0.85	0.97	0.99	0.83	0.88	0.89	0.78	0.87	0.57	0.53	0.55
Hybla	0.72	0.77	0.81	0.66	0.87	0.83	0.99	0.96	0.98	0.92	0.97	0.58	0.52	0.56
YeAH	0.73	0.78	0.88	0.68	0.89	0.88	0.96	0.99	0.98	0.92	0.97	0.62	0.52	0.56
Illinois	0.76	0.83	0.88	0.66	0.88	0.89	0.98	0.98	0.99	0.92	0.95	0.58	0.52	0.54
Veno	0.66	0.71	0.78	0.61	0.82	0.78	0.92	0.92	0.92	0.98	0.93	0.60	0.52	0.54
Westwood+	0.70	0.70	0.86	0.67	0.88	0.87	0.97	0.97	0.95	0.93	1.00	0.58	0.52	0.54
BBR	0.60	0.58	0.57	0.59	0.58	0.57	0.58	0.62	0.58	0.60	0.58	0.94	0.65	0.79
Vegas	0.53	0.53	0.52	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.65	1.00	0.67
LoLa	0.58	0.57	0.56	0.67	0.56	0.55	0.56	0.56	0.54	0.54	0.54	0.79	0.67	0.80

P4air

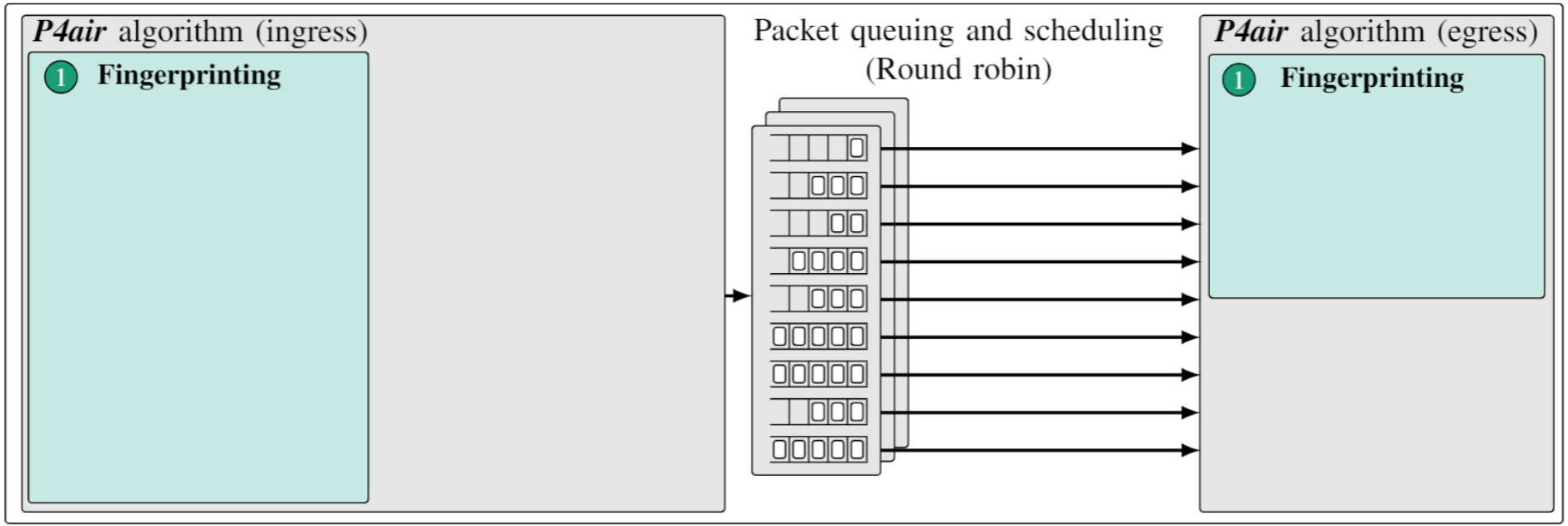
P4air



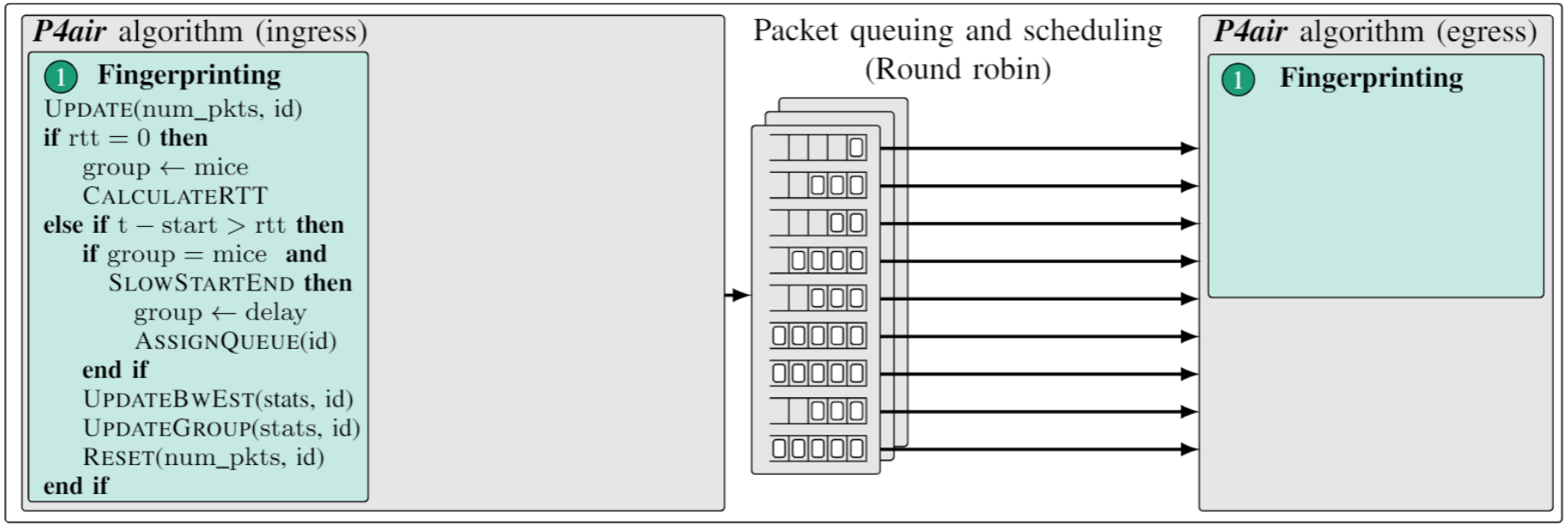
P4air


P4air algorithm (ingress)
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)

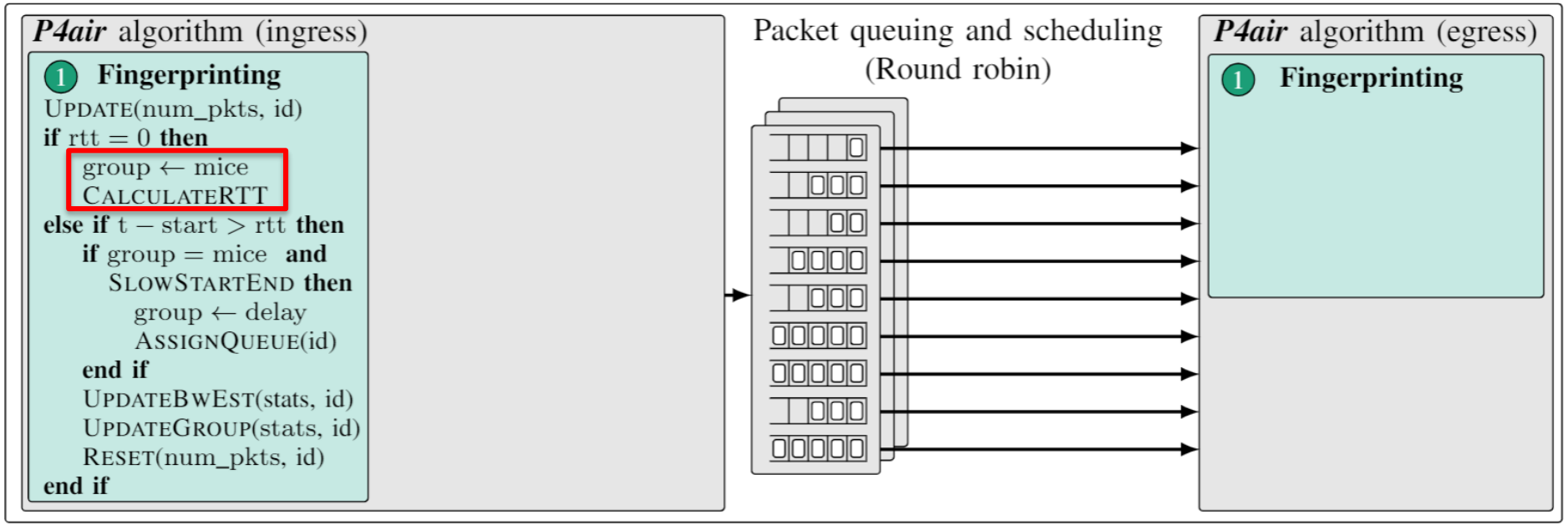
P4air


P4air algorithm (ingress)
1 Fingerprinting
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting

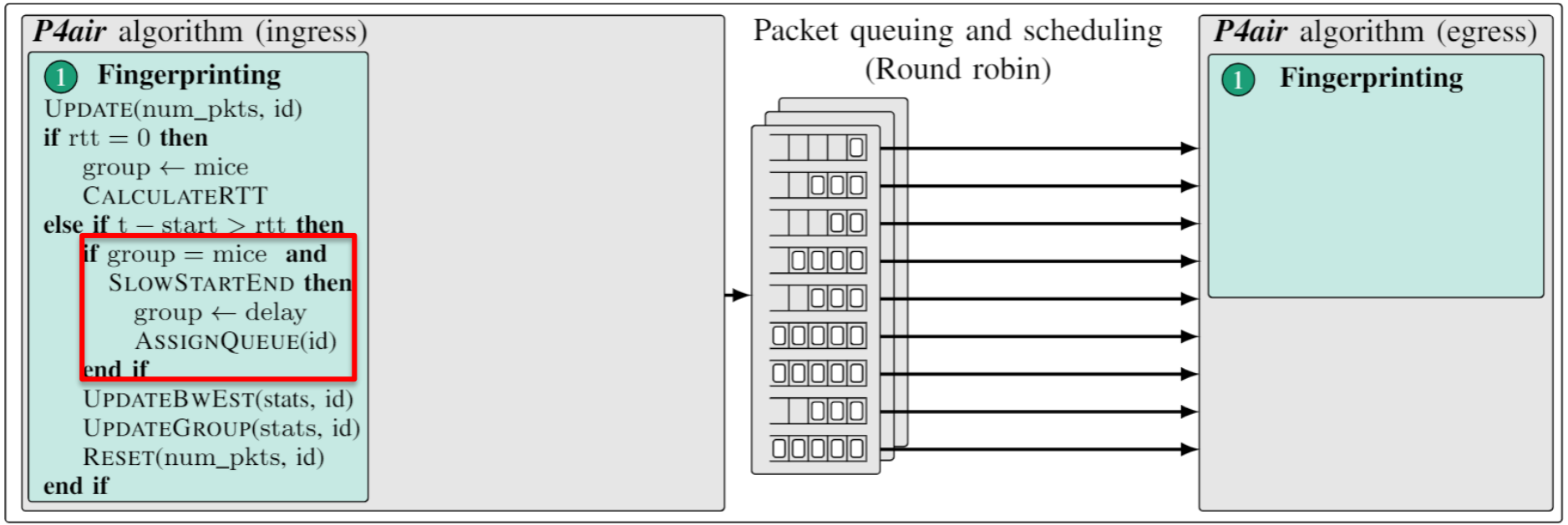
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting

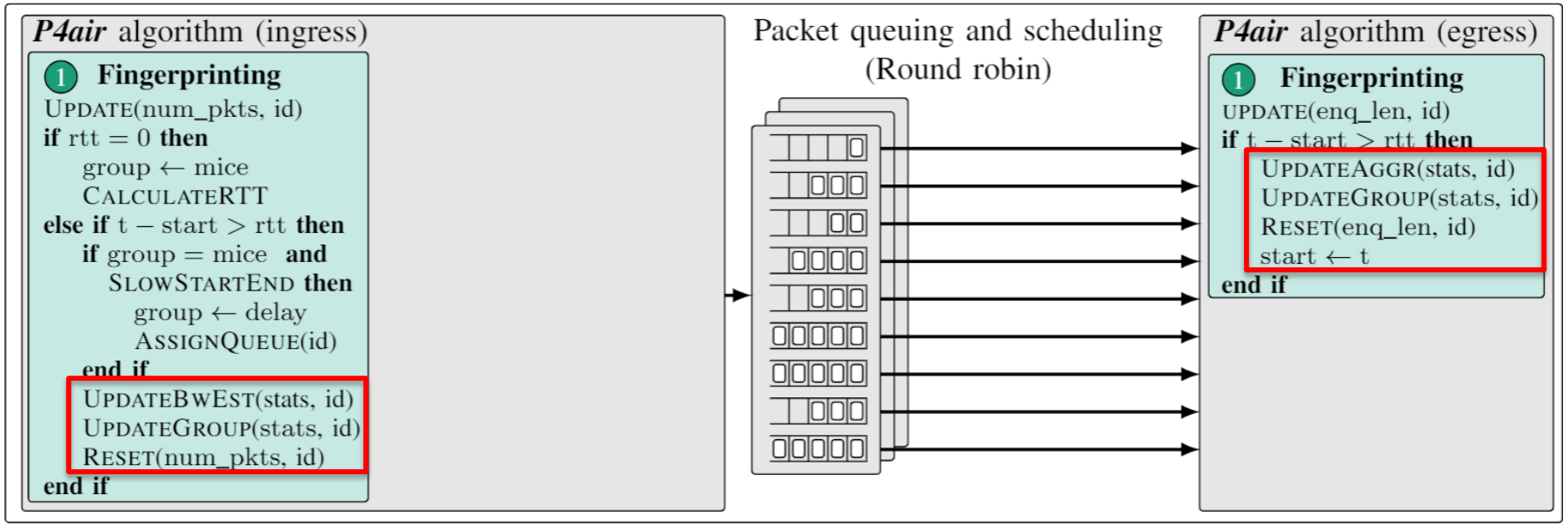
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting

P4air

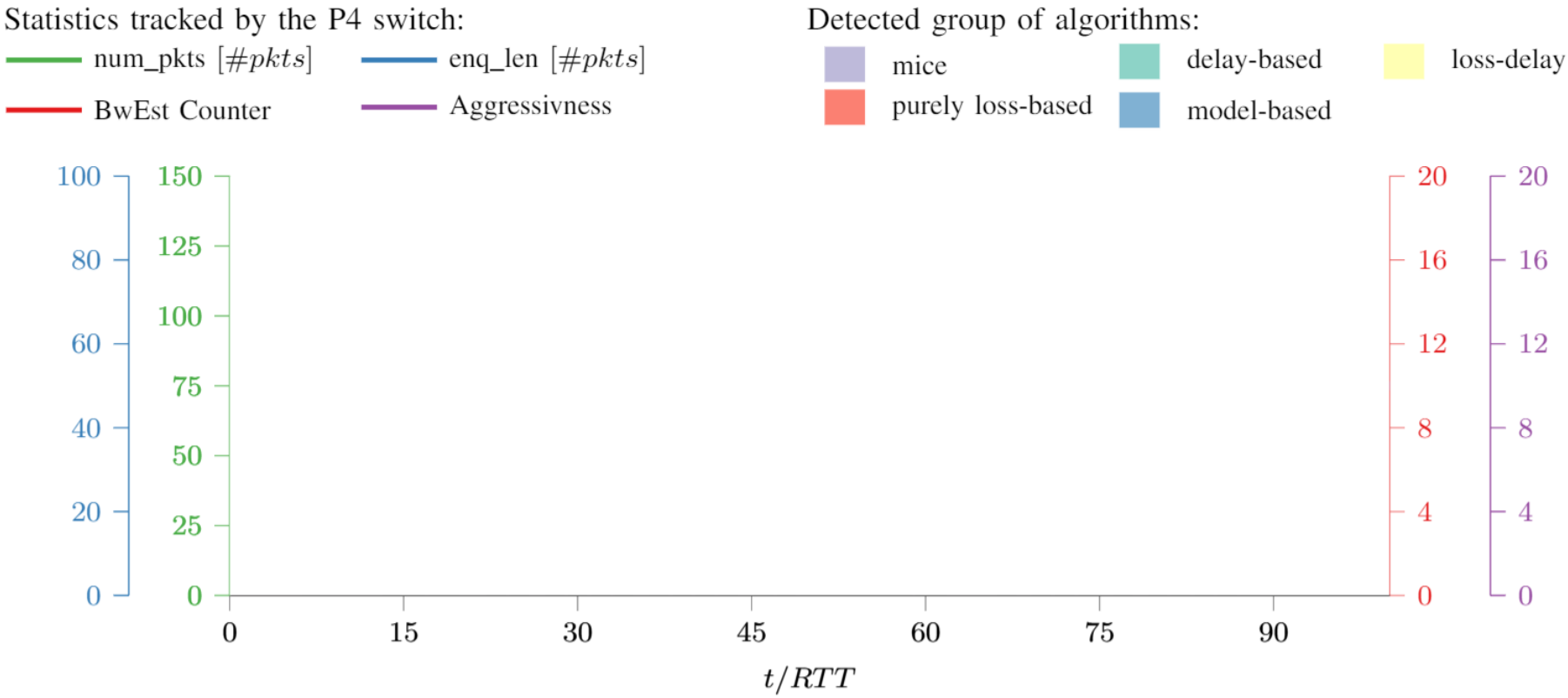

P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting

P4air

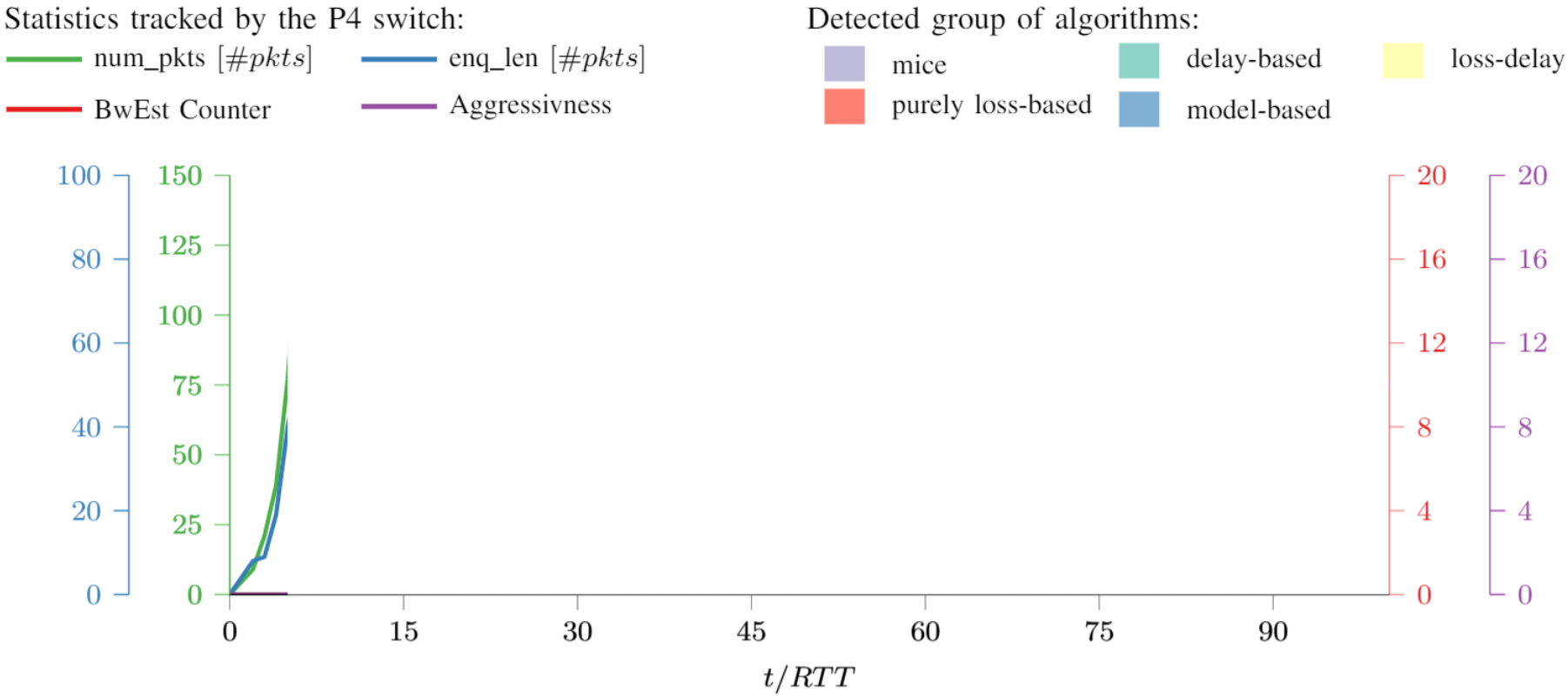

P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if

Fingerprinting

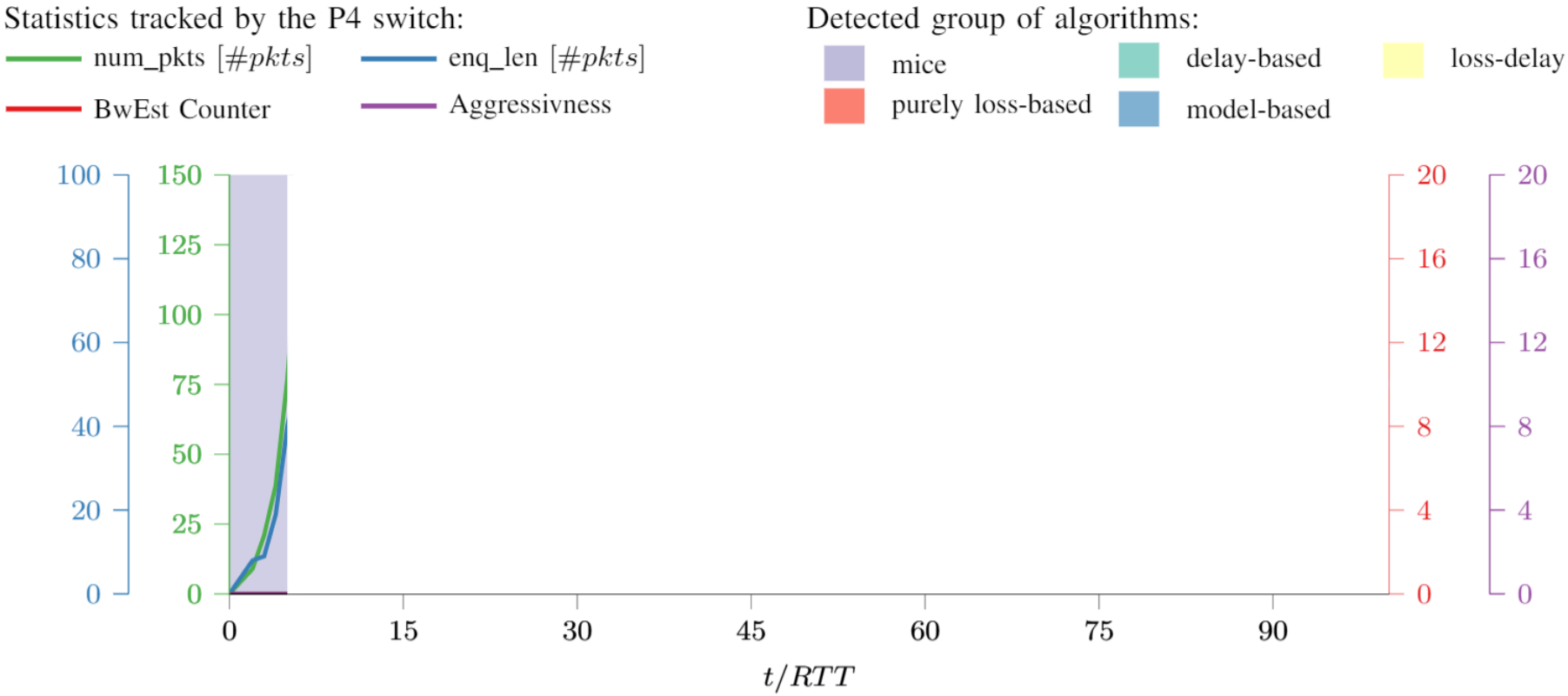
Fingerprinting - Cubic



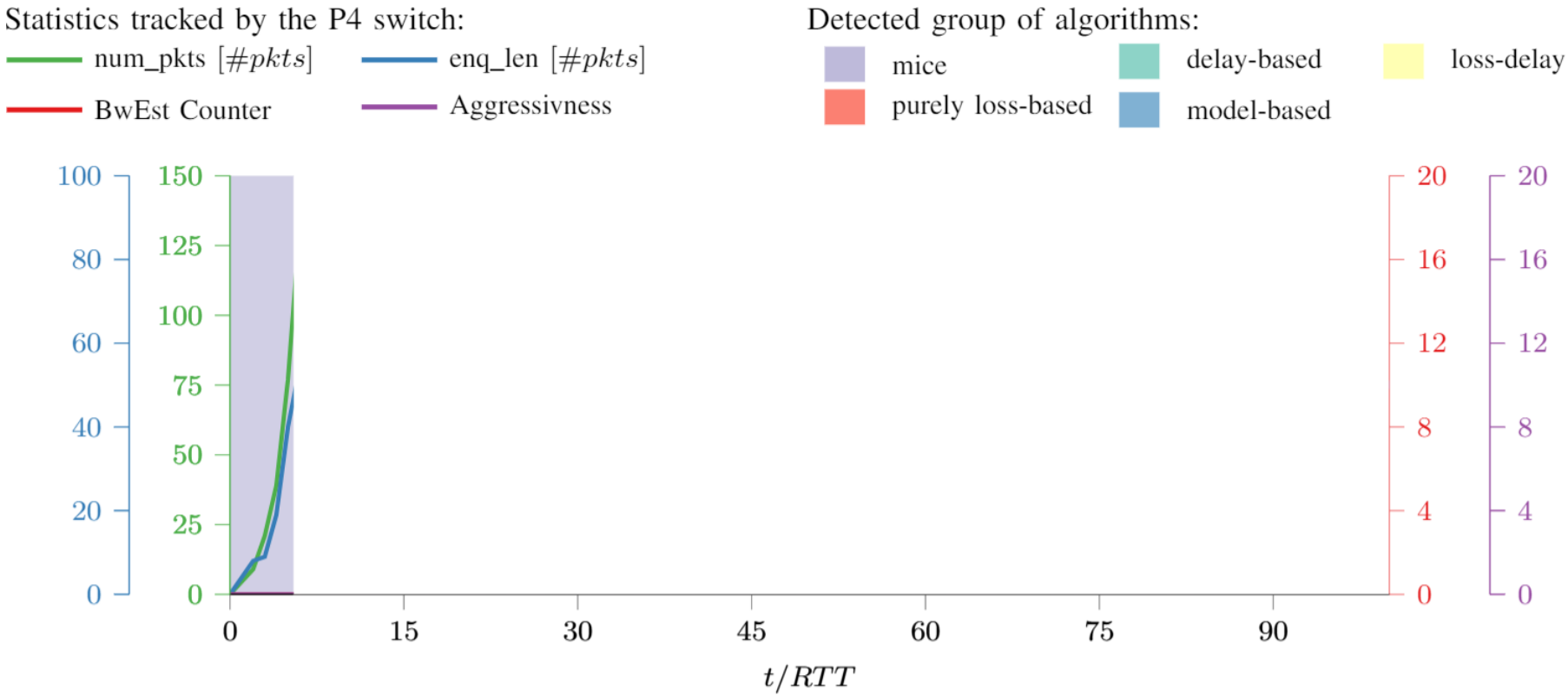
Fingerprinting - Cubic



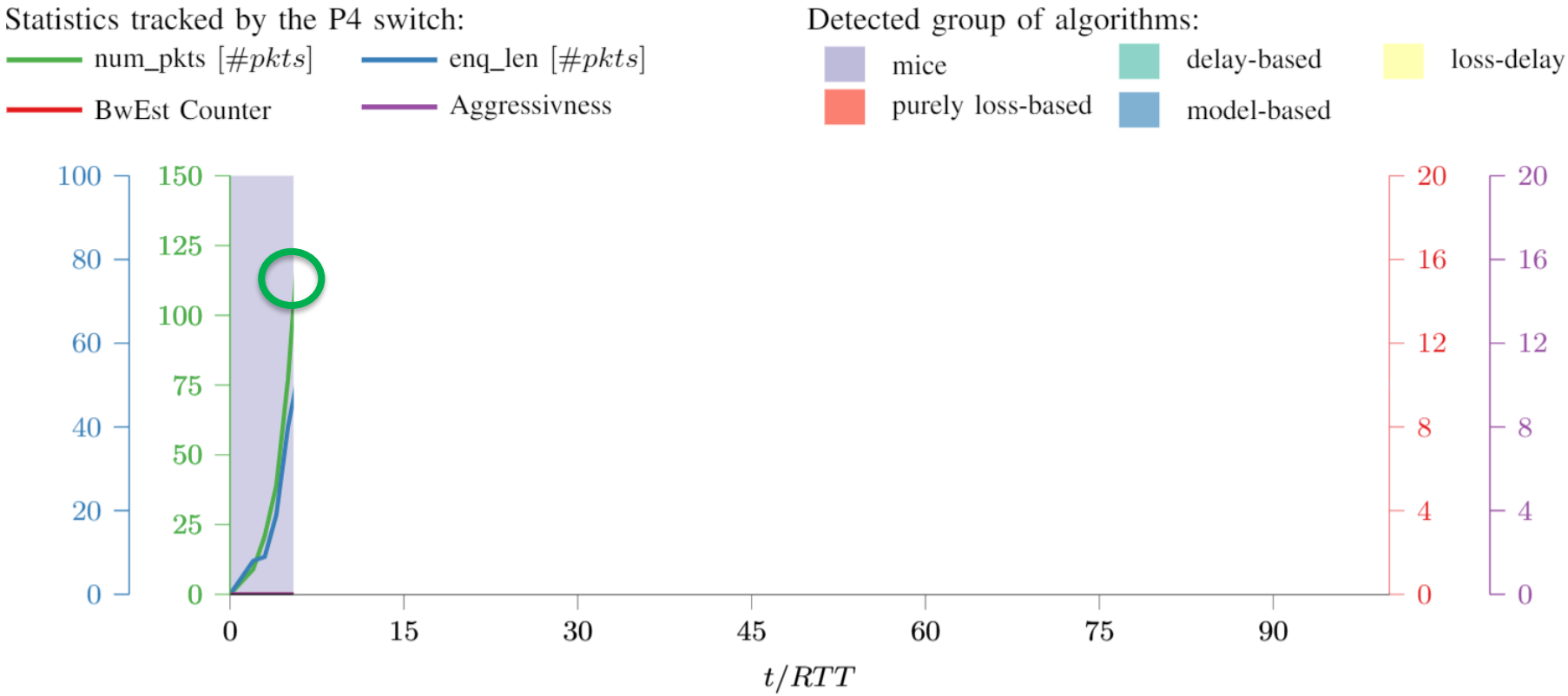
Fingerprinting - Cubic



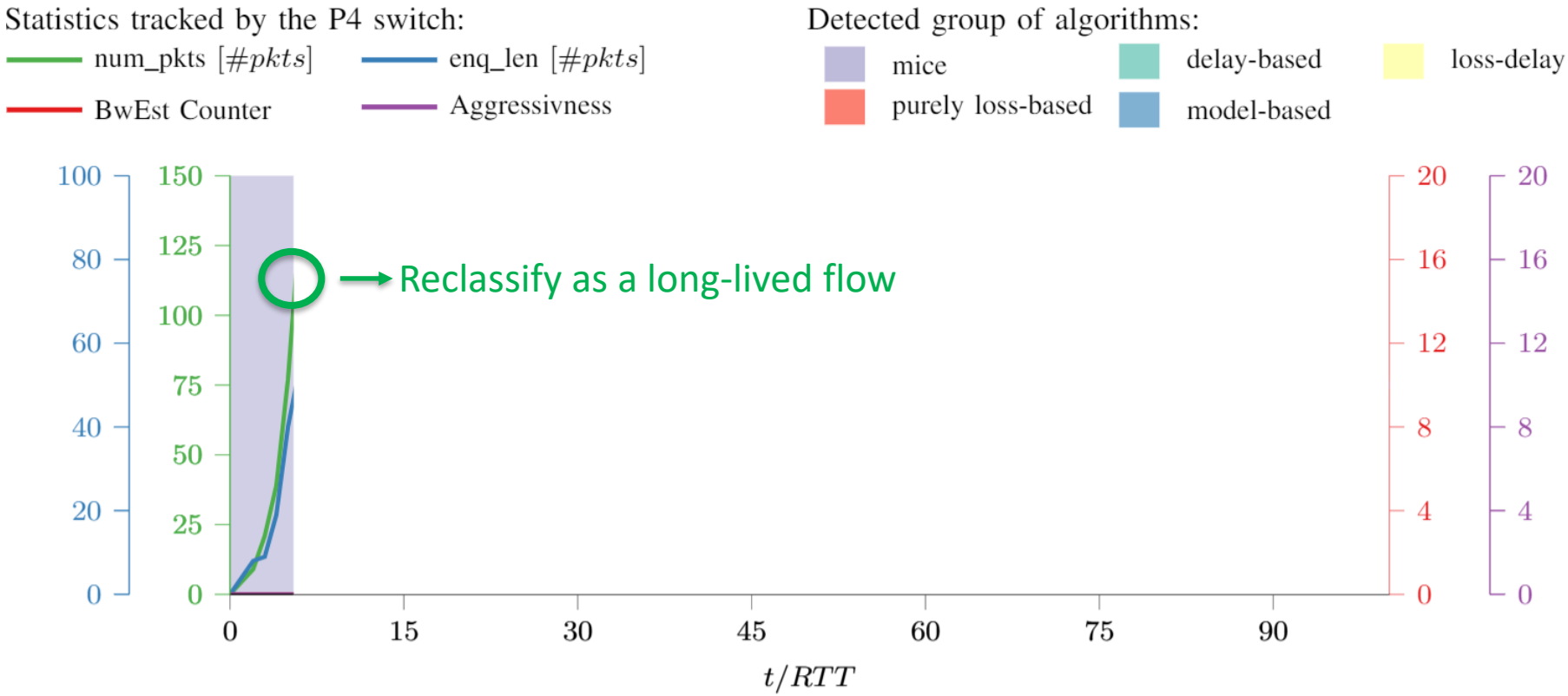
Fingerprinting - Cubic



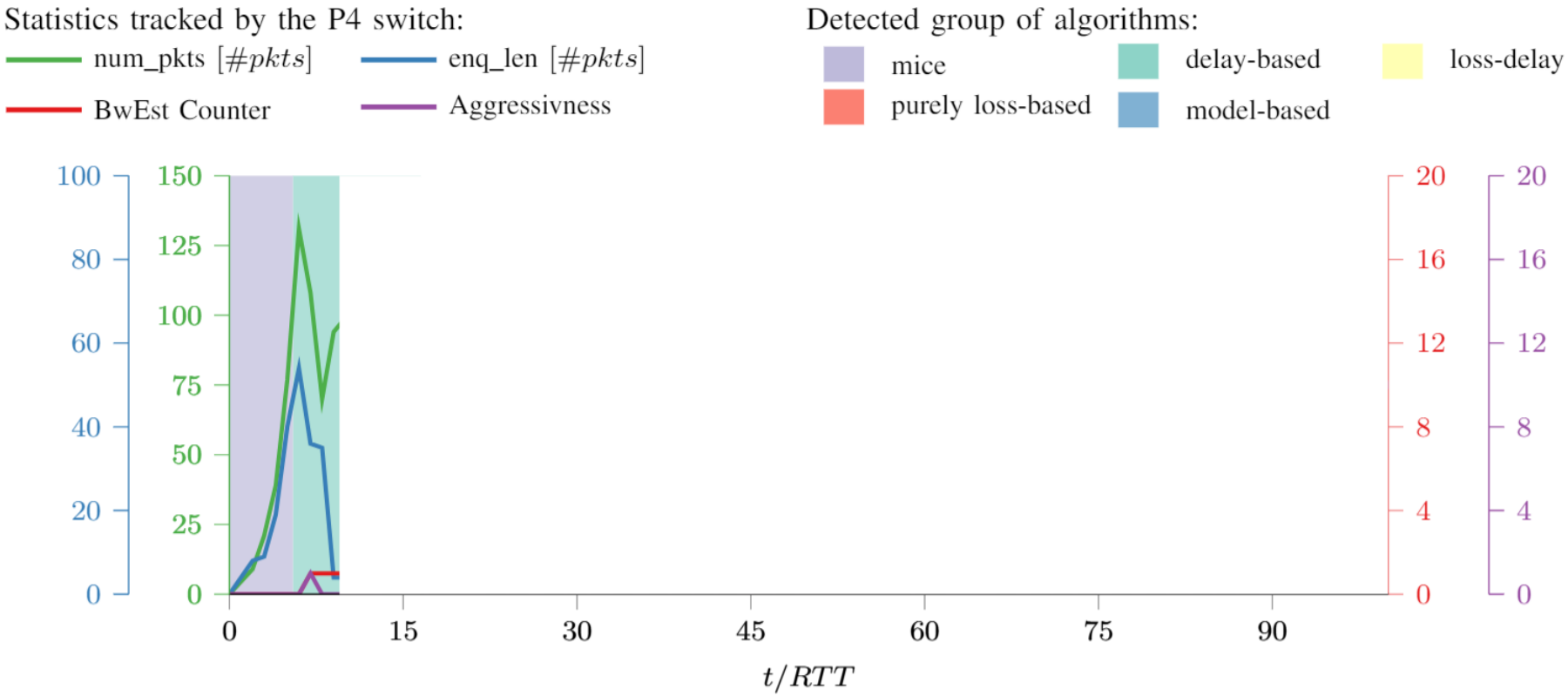
Fingerprinting - Cubic



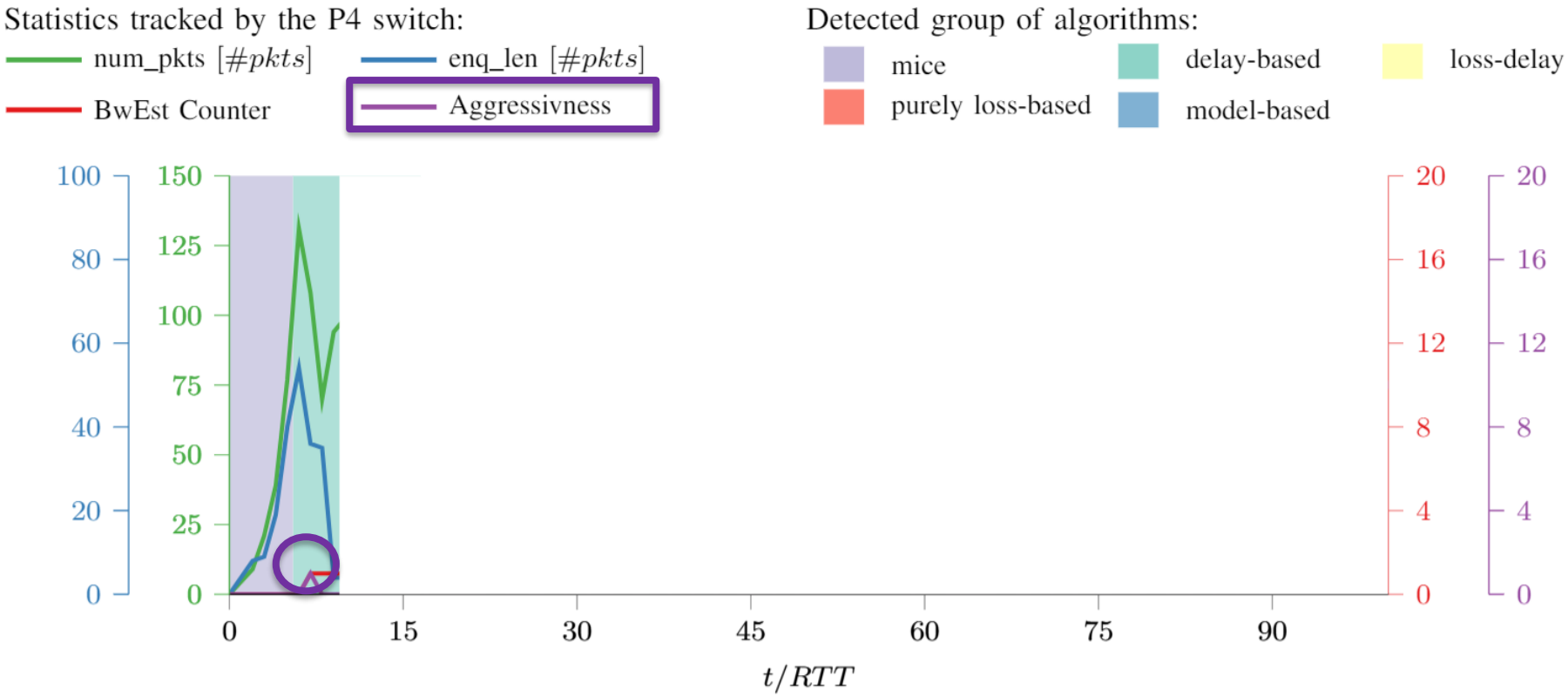
Fingerprinting - Cubic



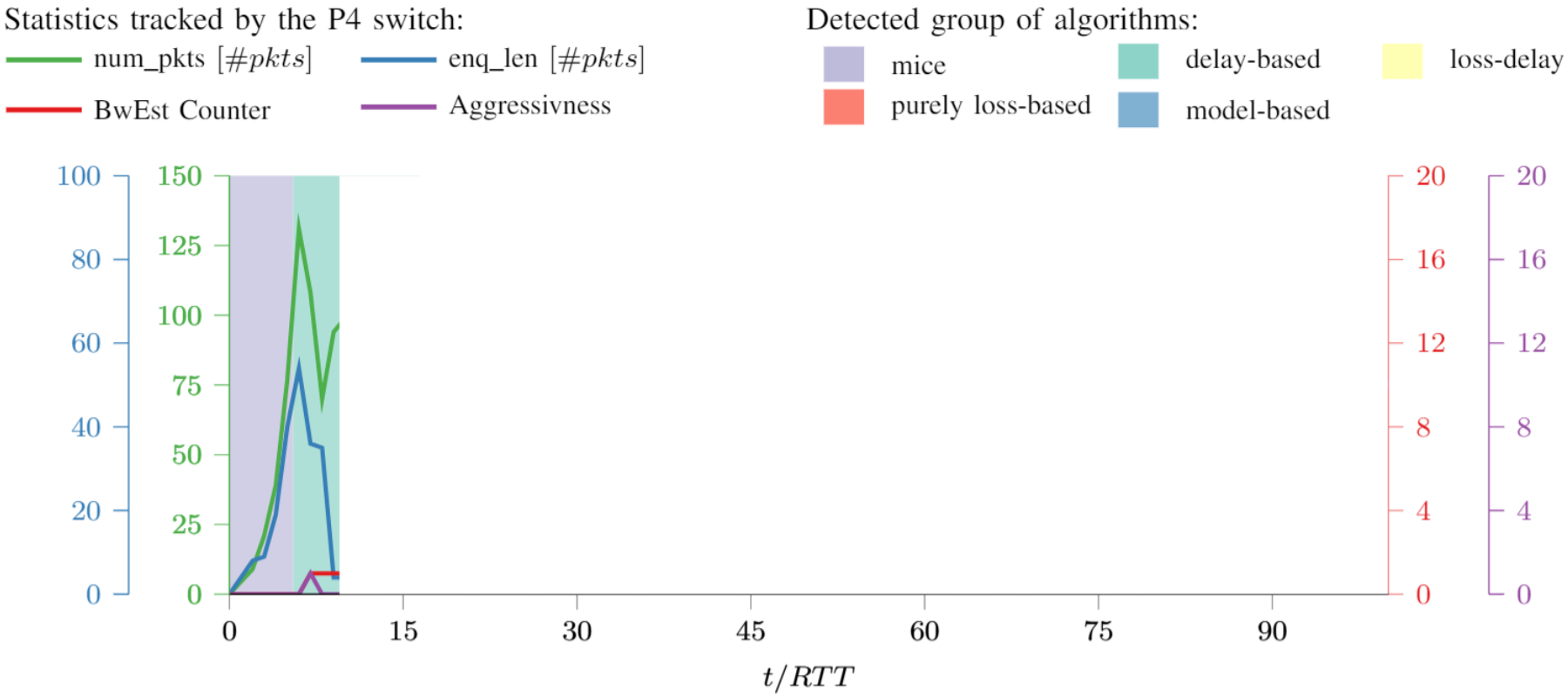
Fingerprinting - Cubic



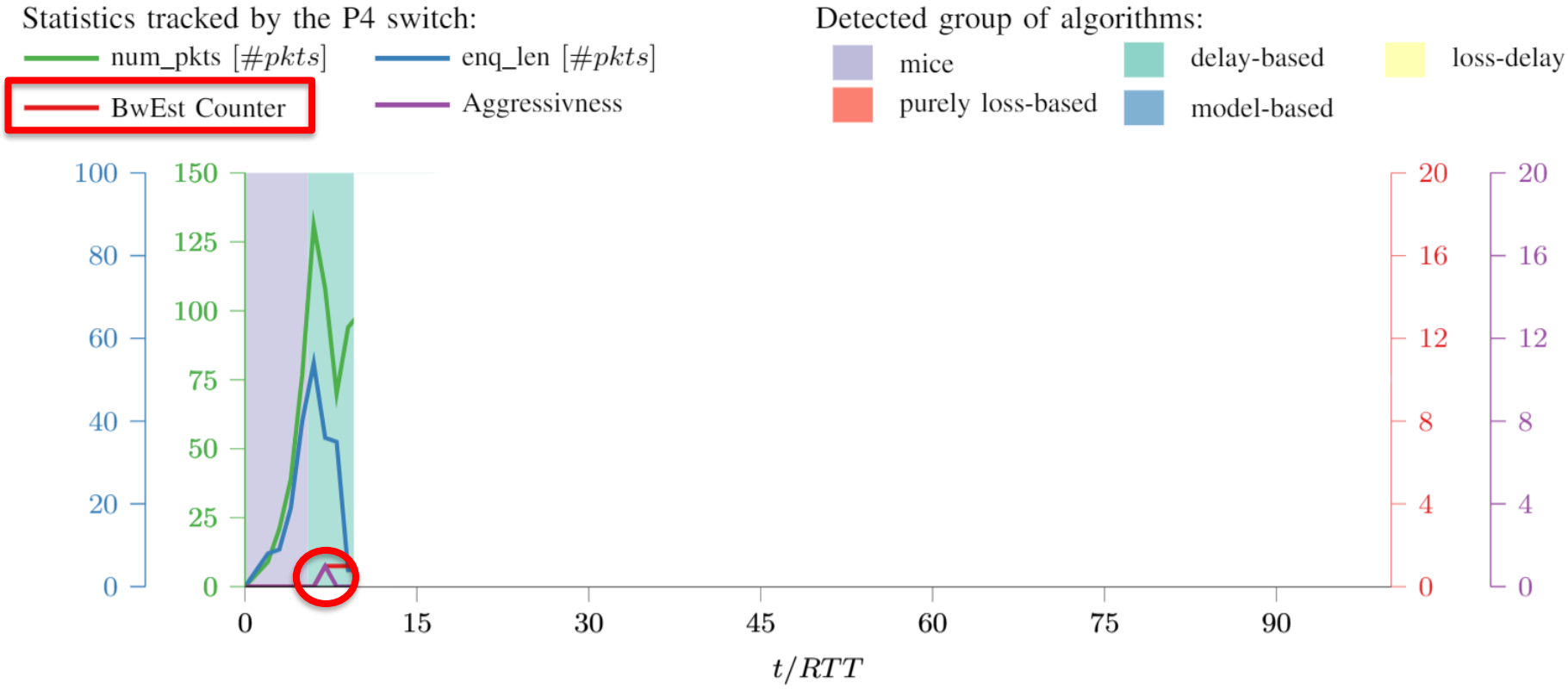
Fingerprinting - Cubic



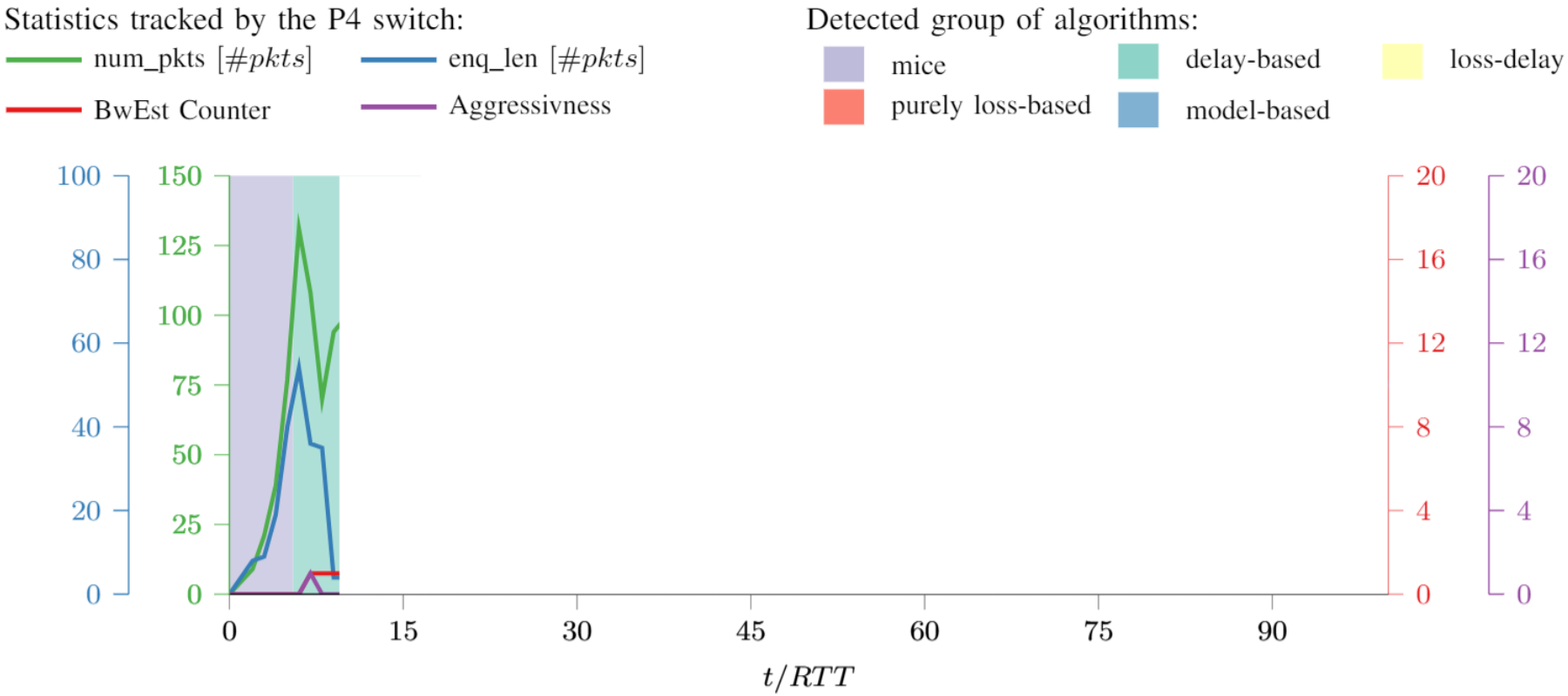
Fingerprinting - Cubic



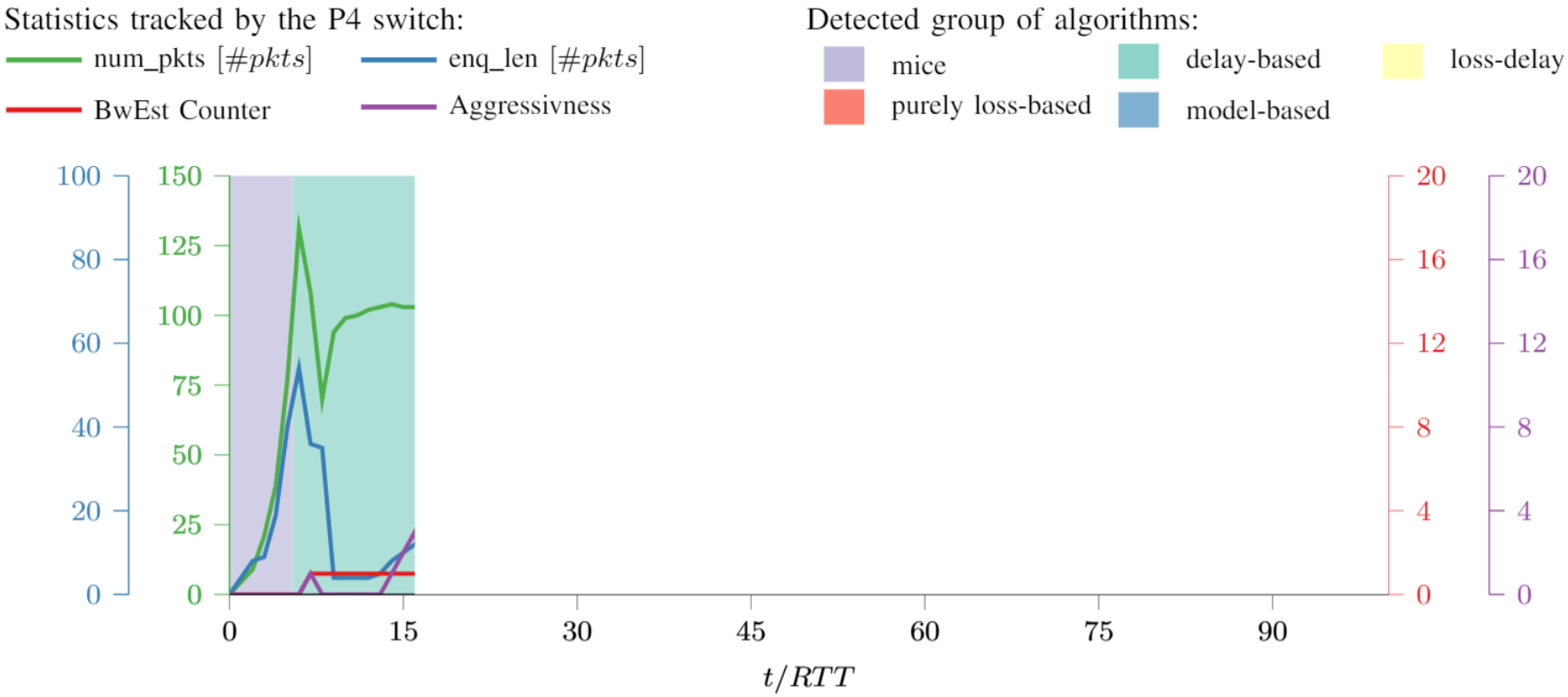
Fingerprinting - Cubic



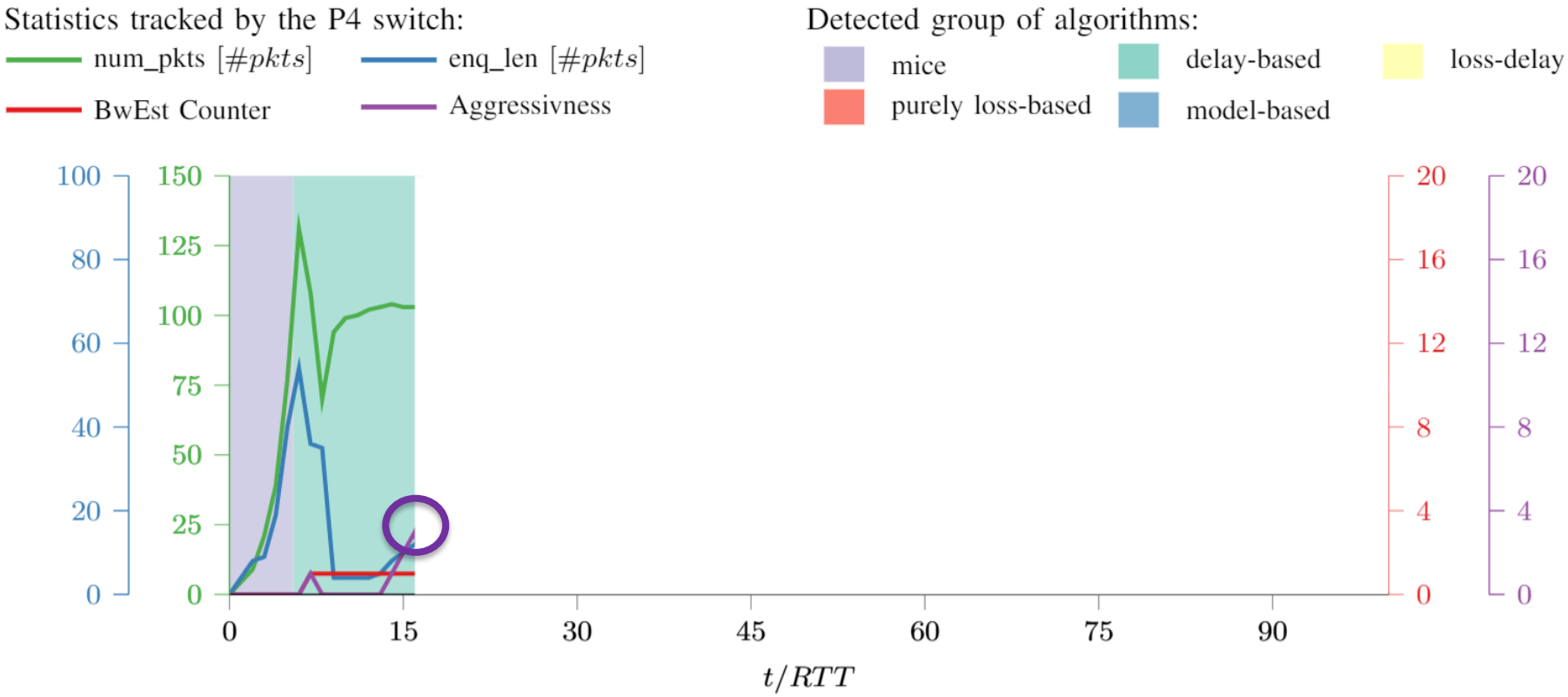
Fingerprinting - Cubic



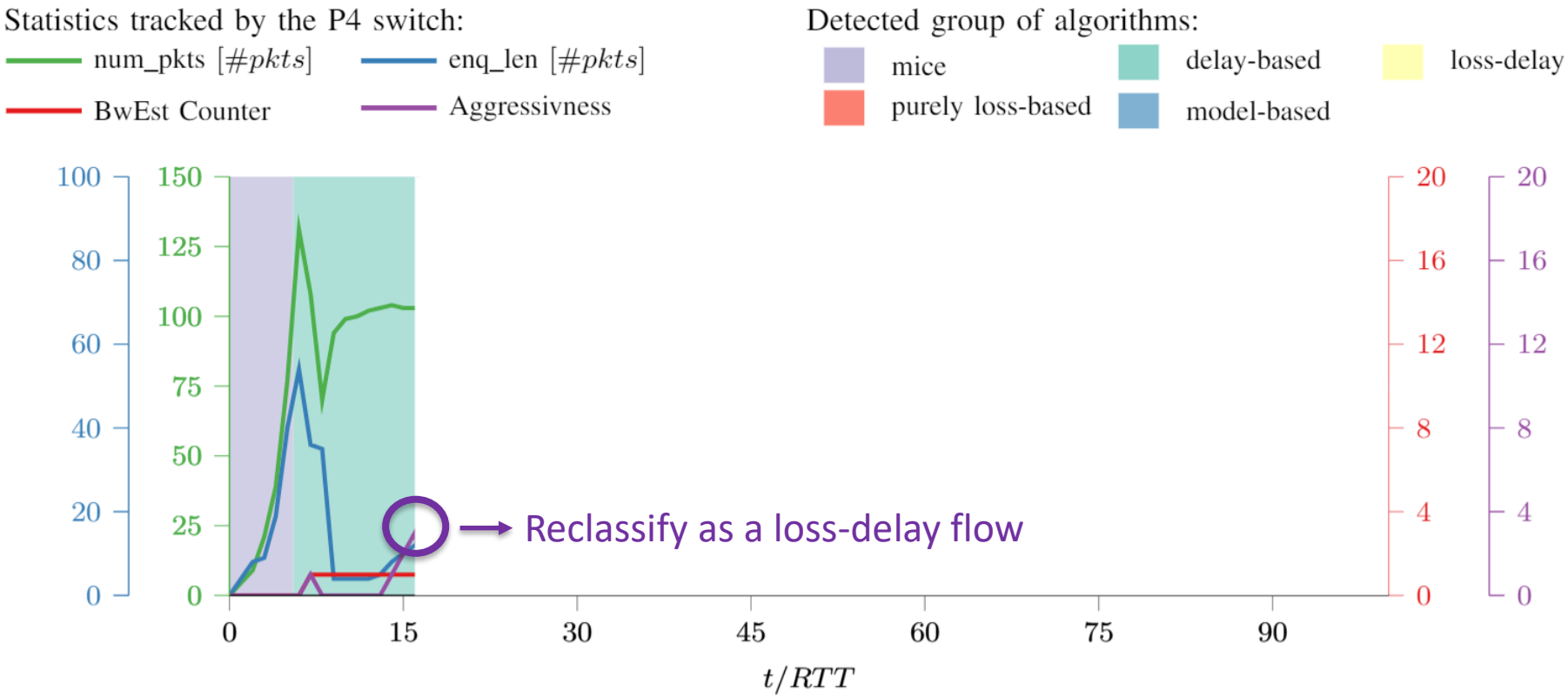
Fingerprinting - Cubic



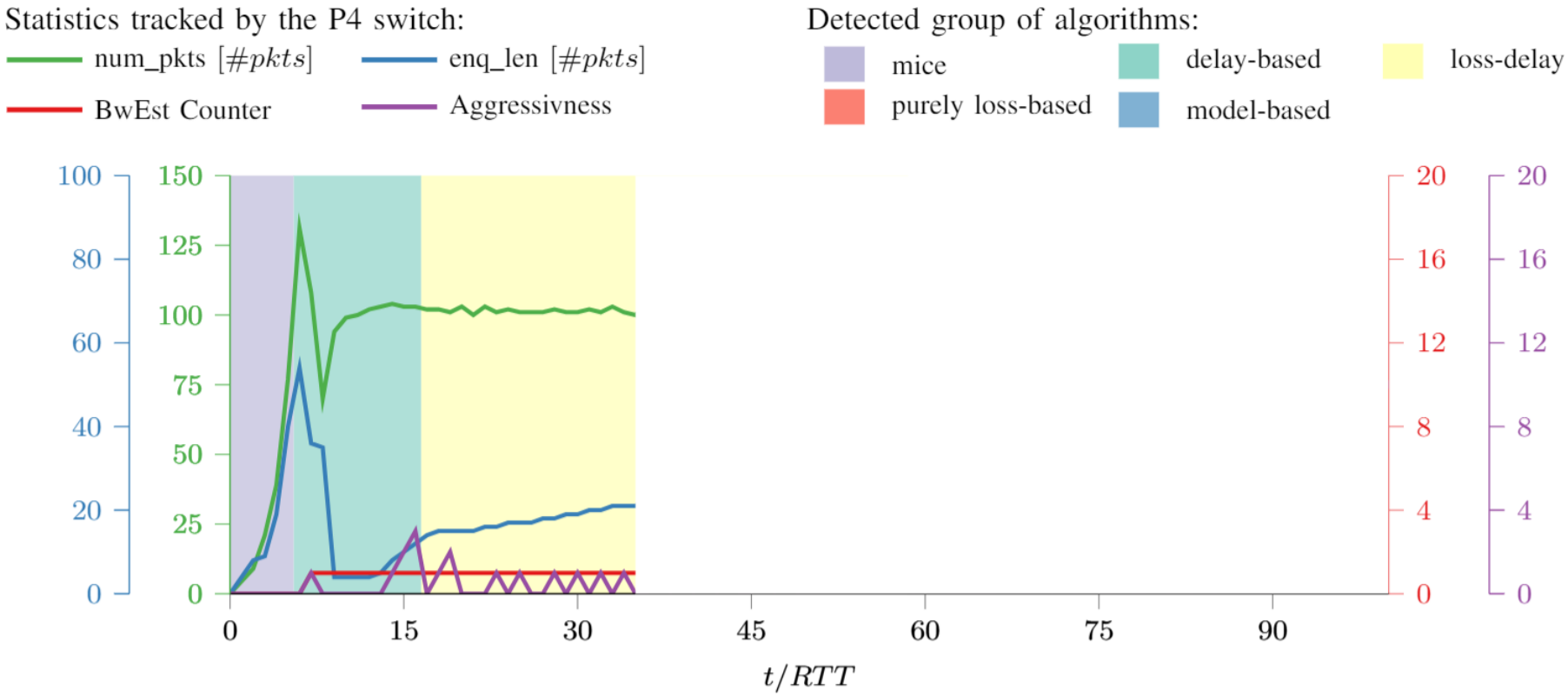
Fingerprinting - Cubic



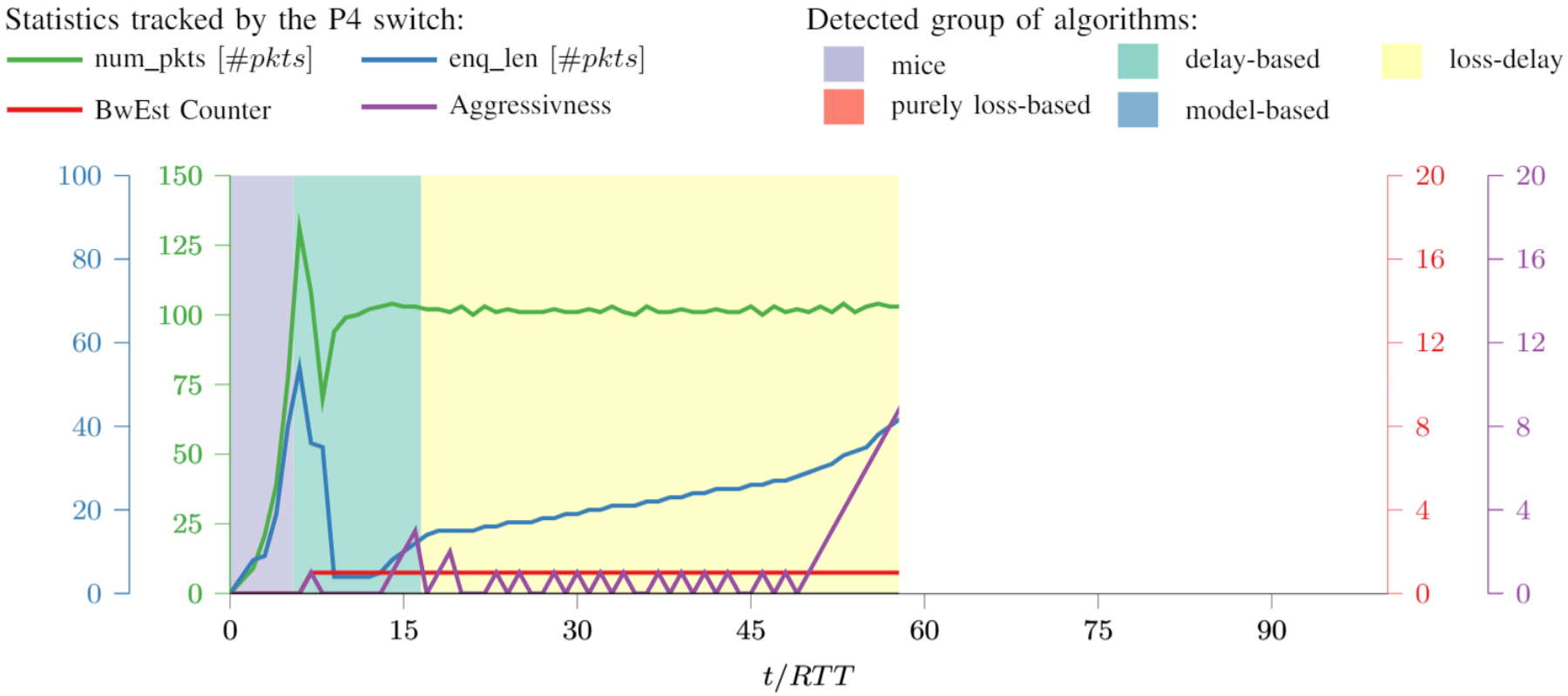
Fingerprinting - Cubic



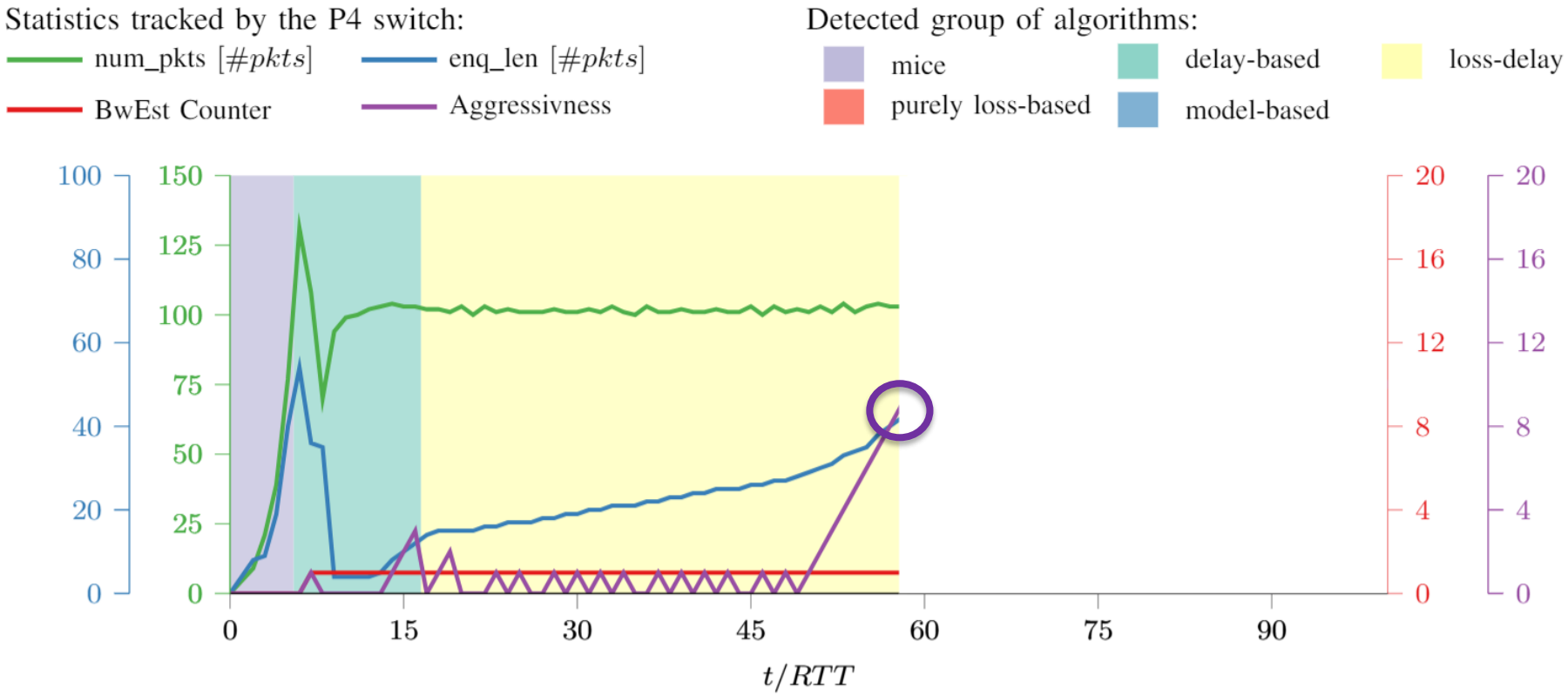
Fingerprinting - Cubic



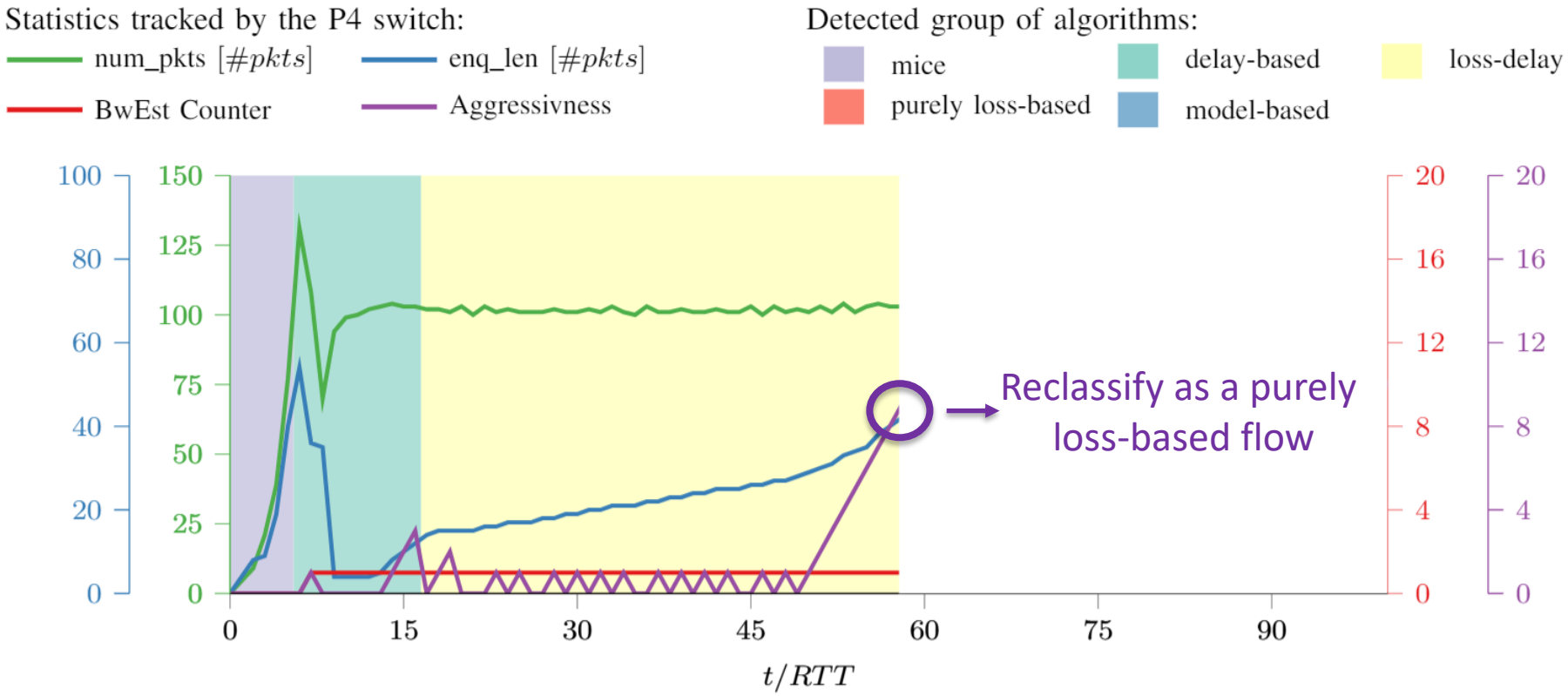
Fingerprinting - Cubic



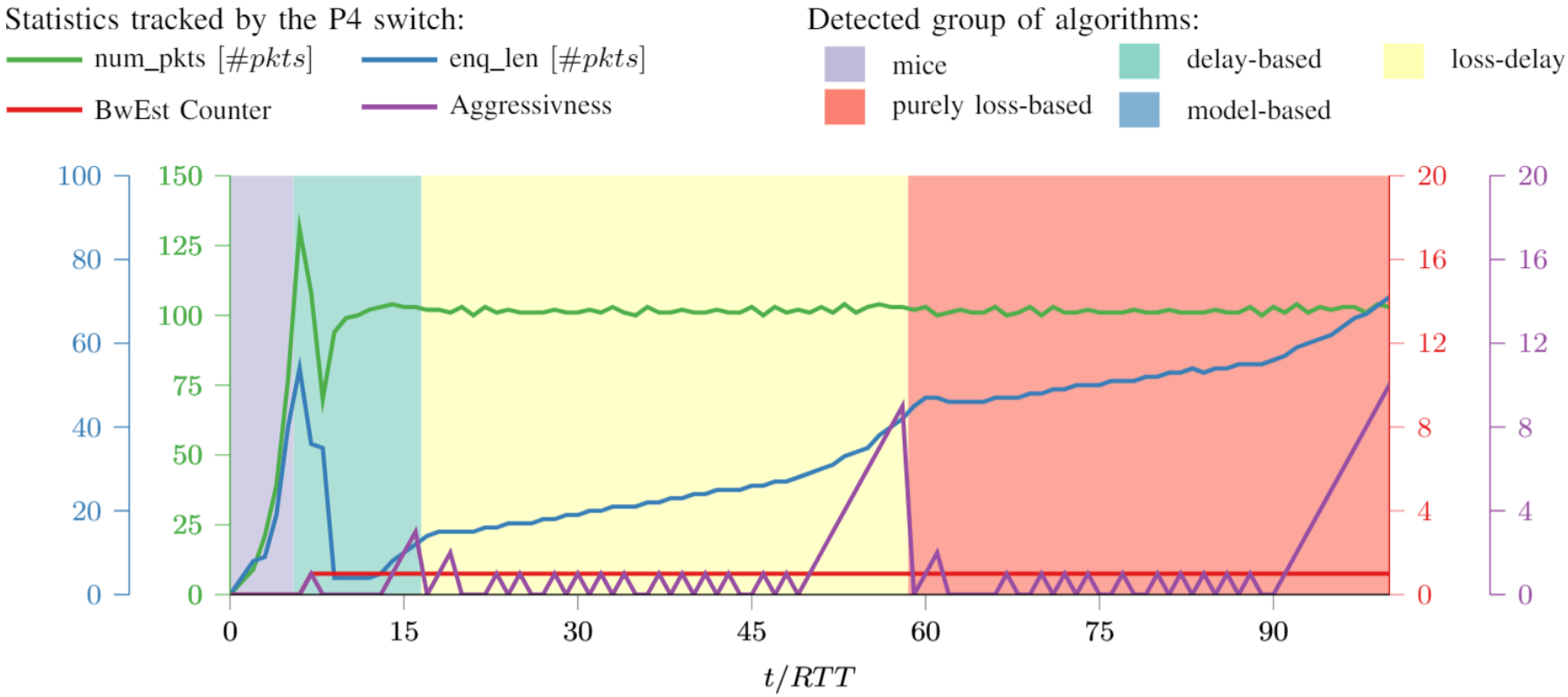
Fingerprinting - Cubic



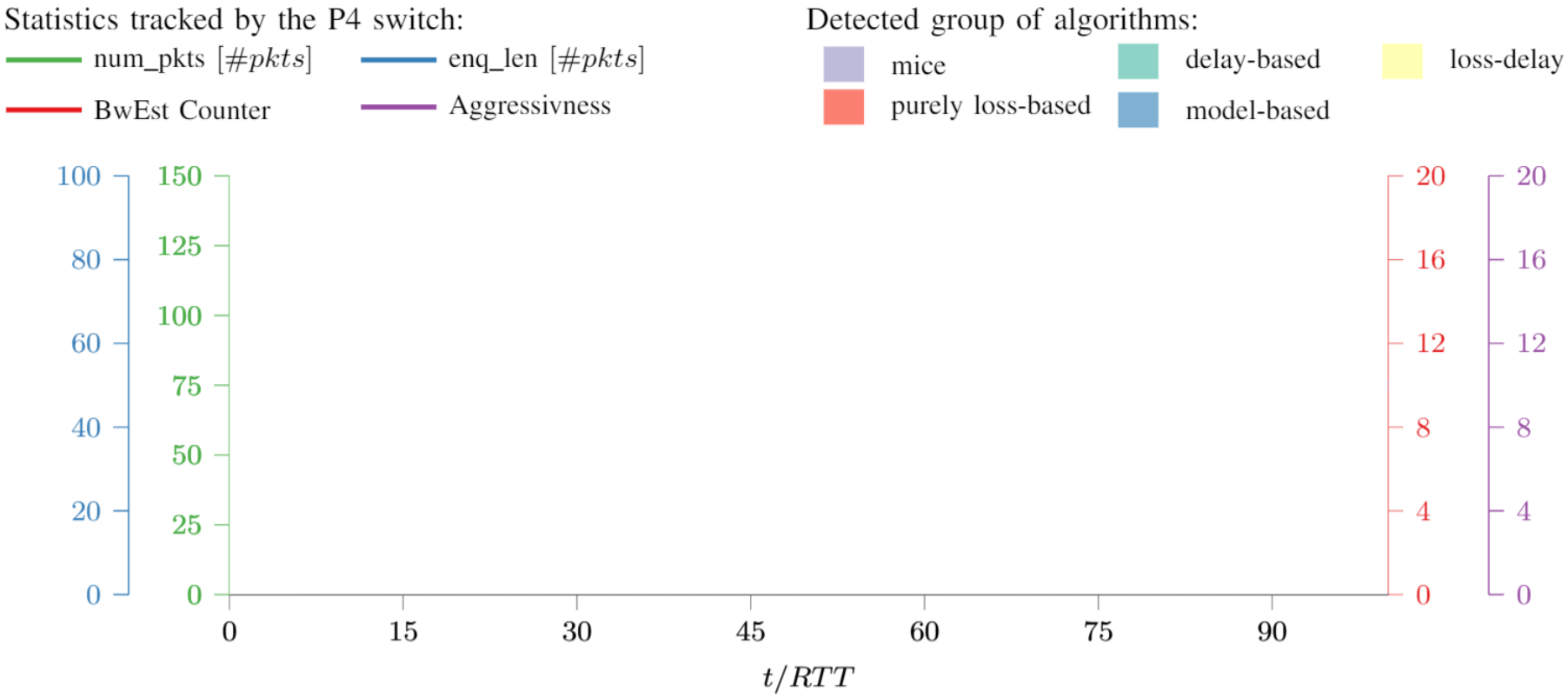
Fingerprinting - Cubic



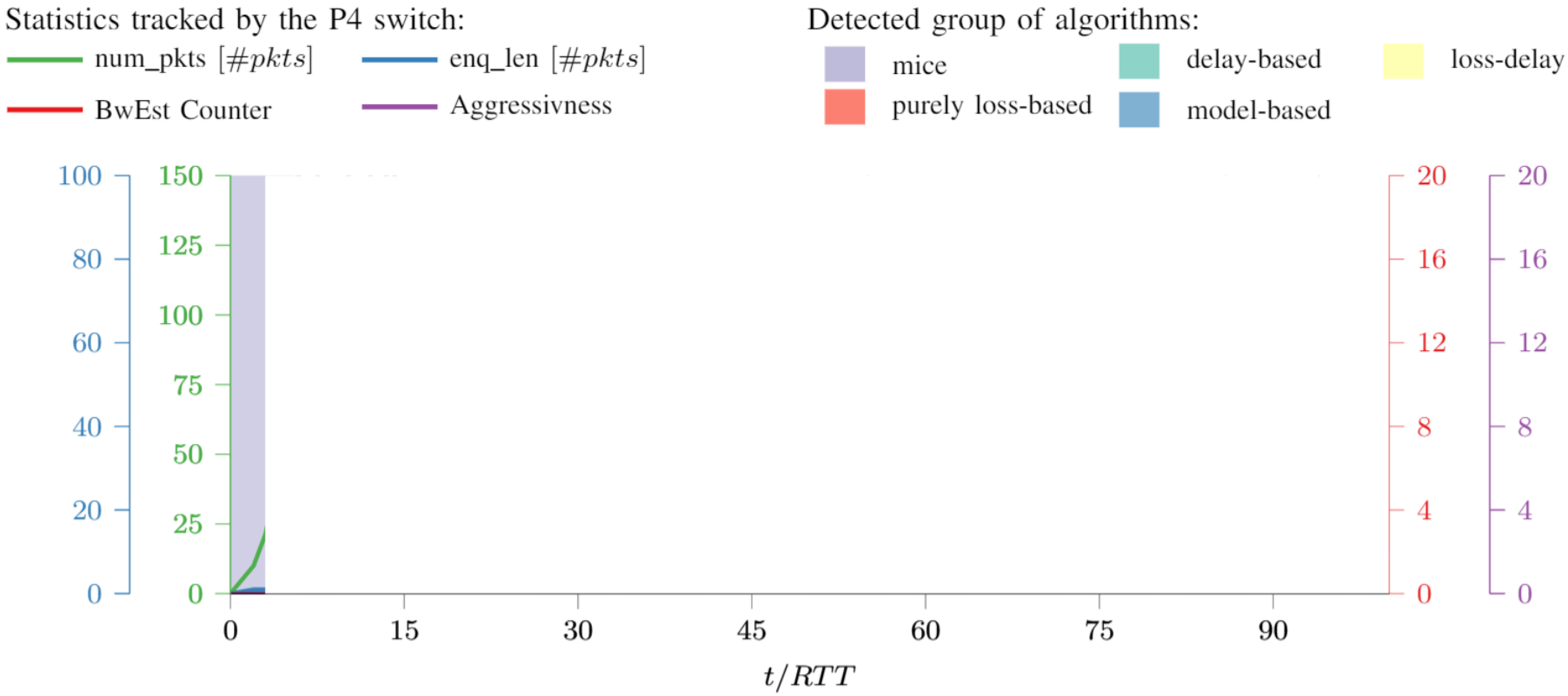
Fingerprinting - Cubic



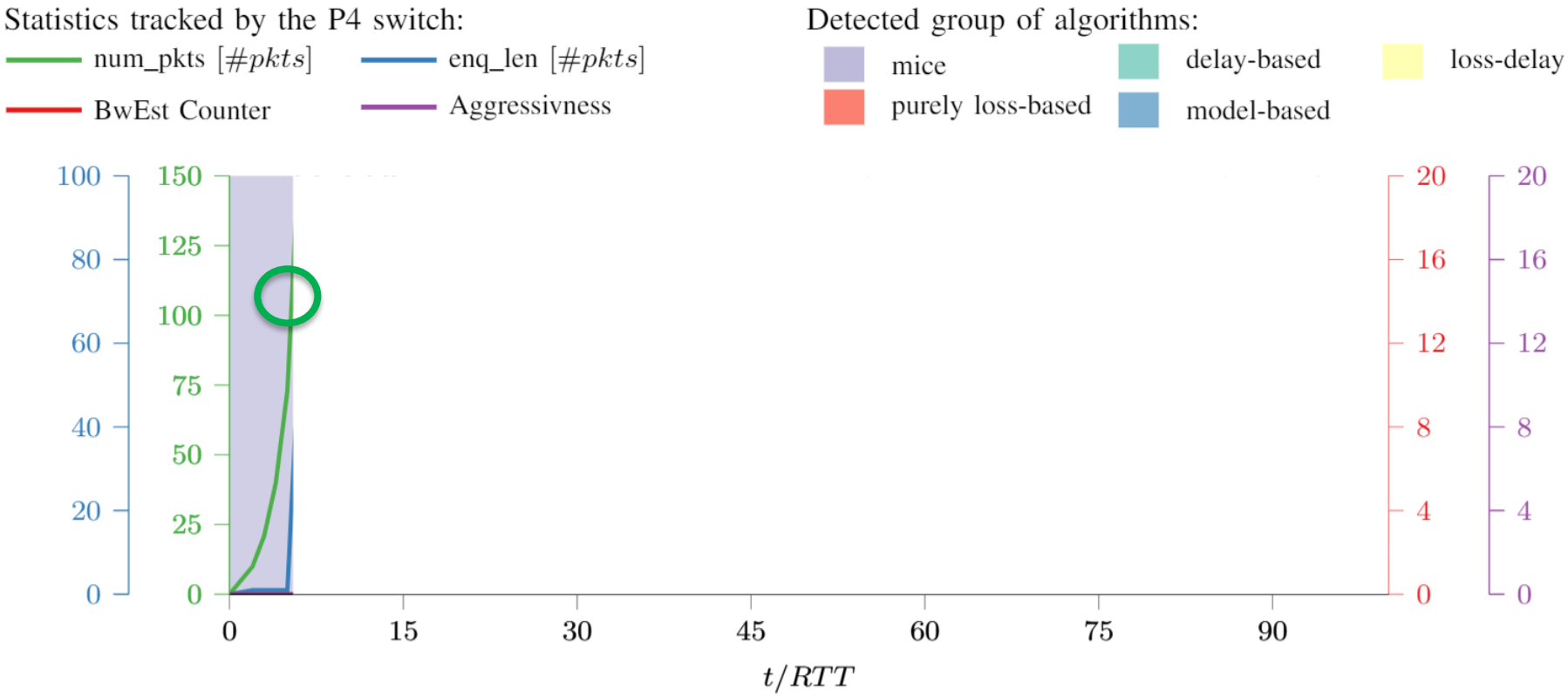
Fingerprinting - BBR



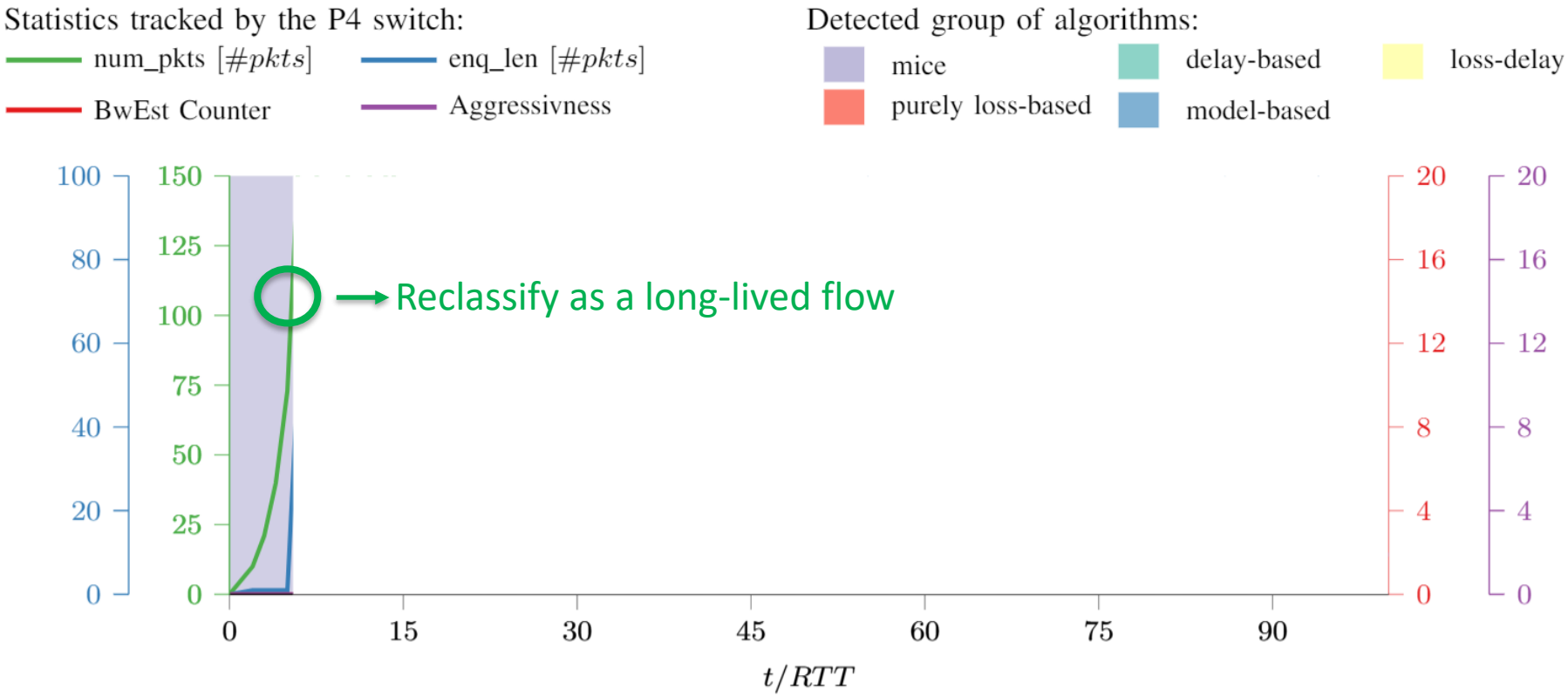
Fingerprinting - BBR



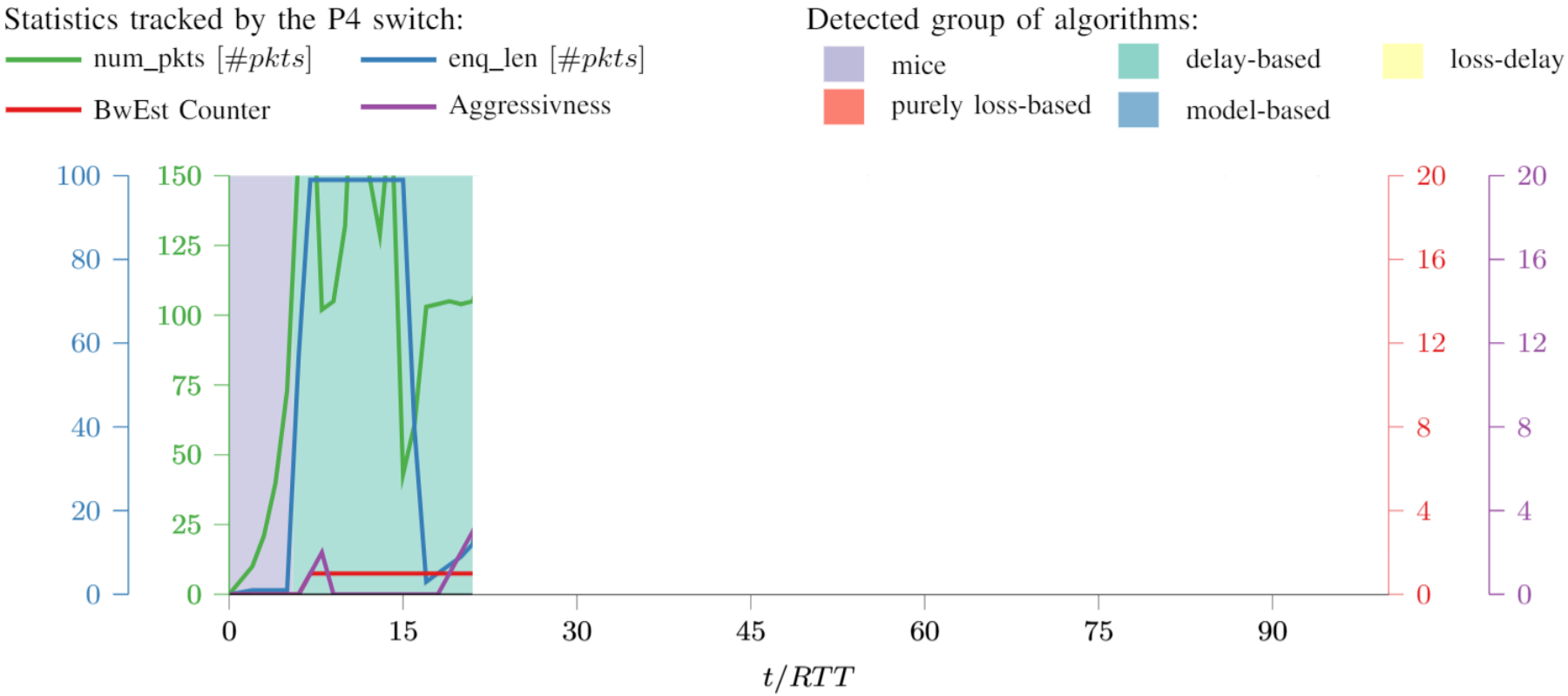
Fingerprinting - BBR



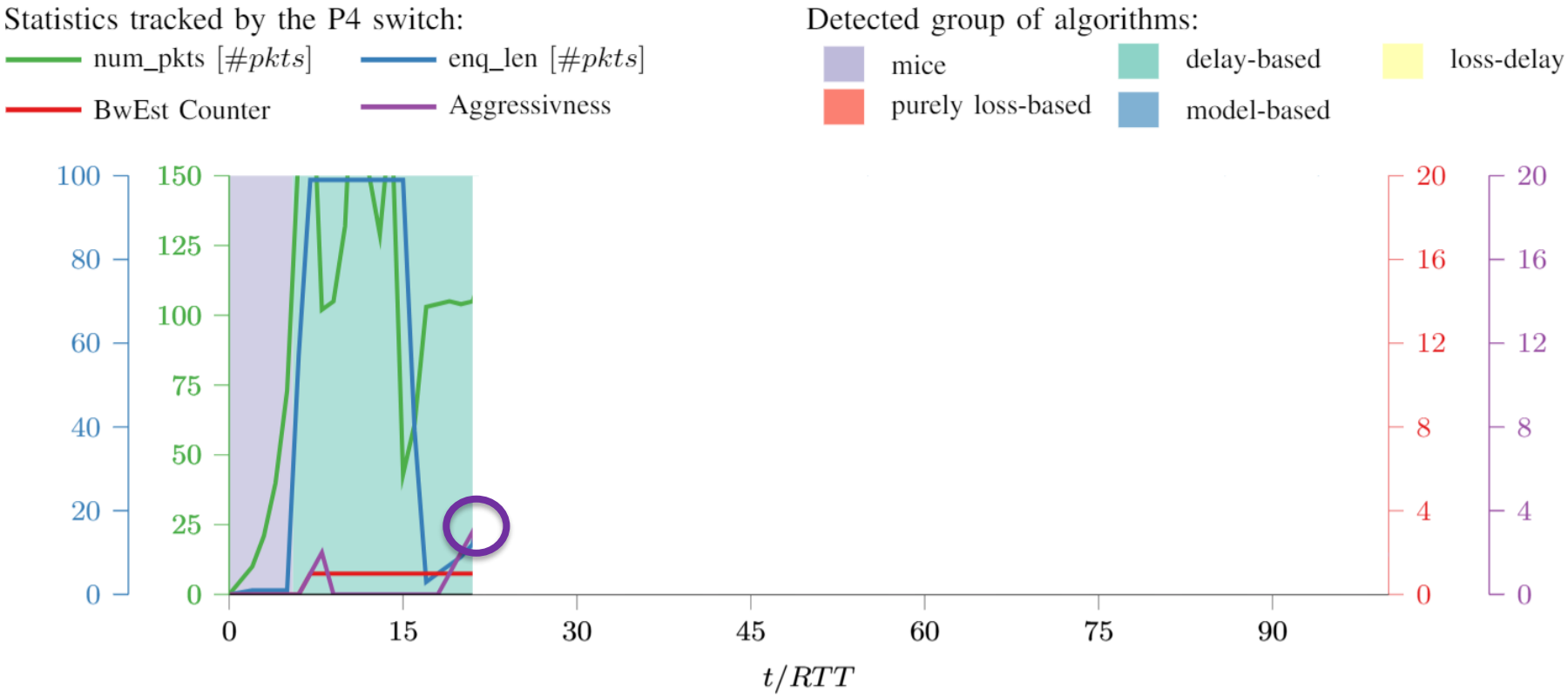
Fingerprinting - BBR



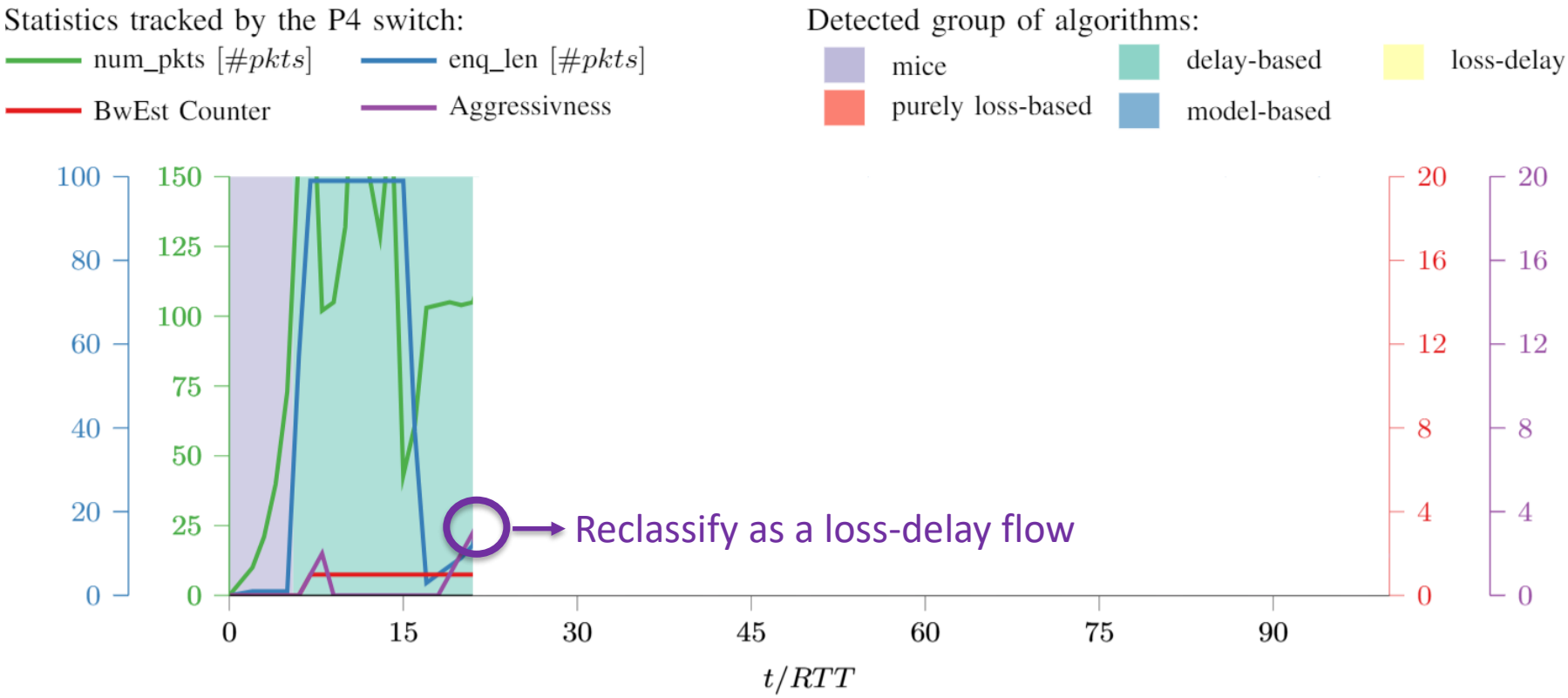
Fingerprinting - BBR



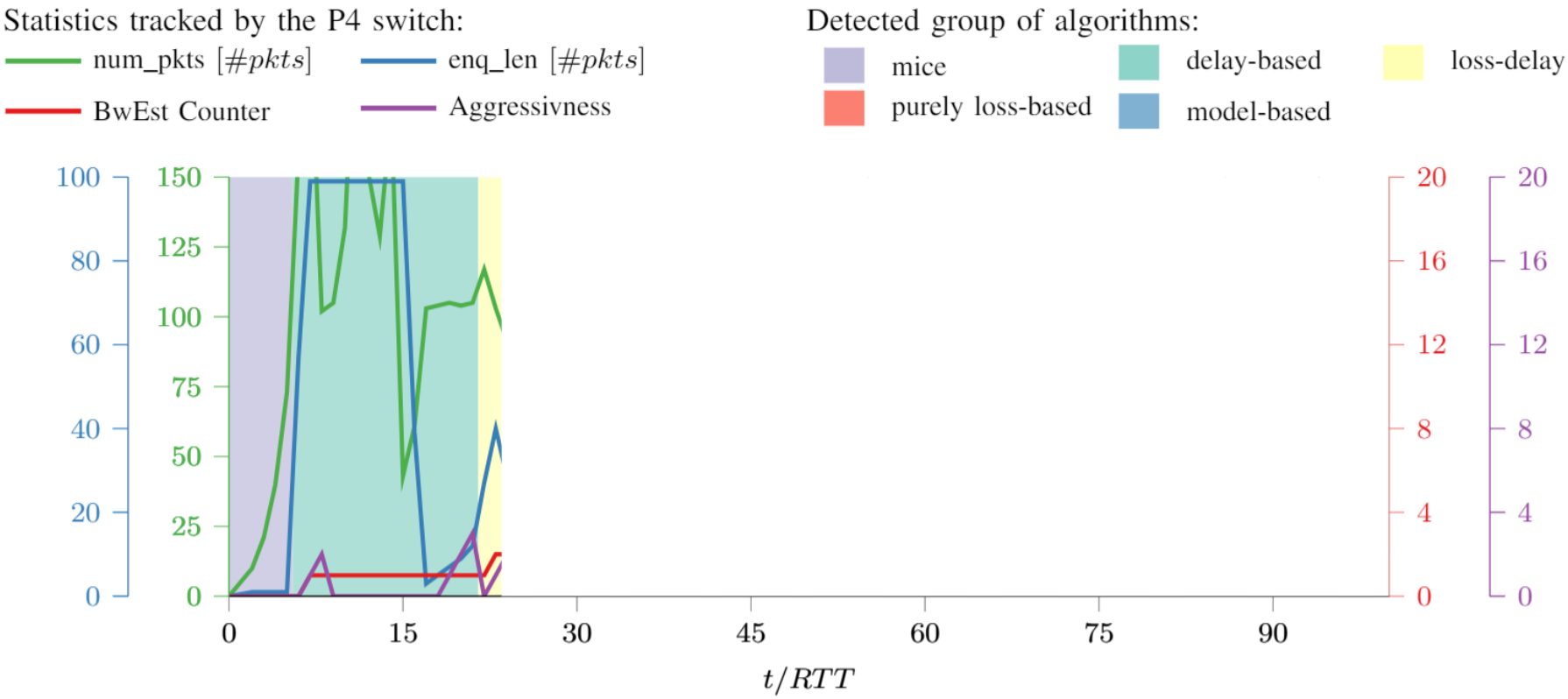
Fingerprinting - BBR



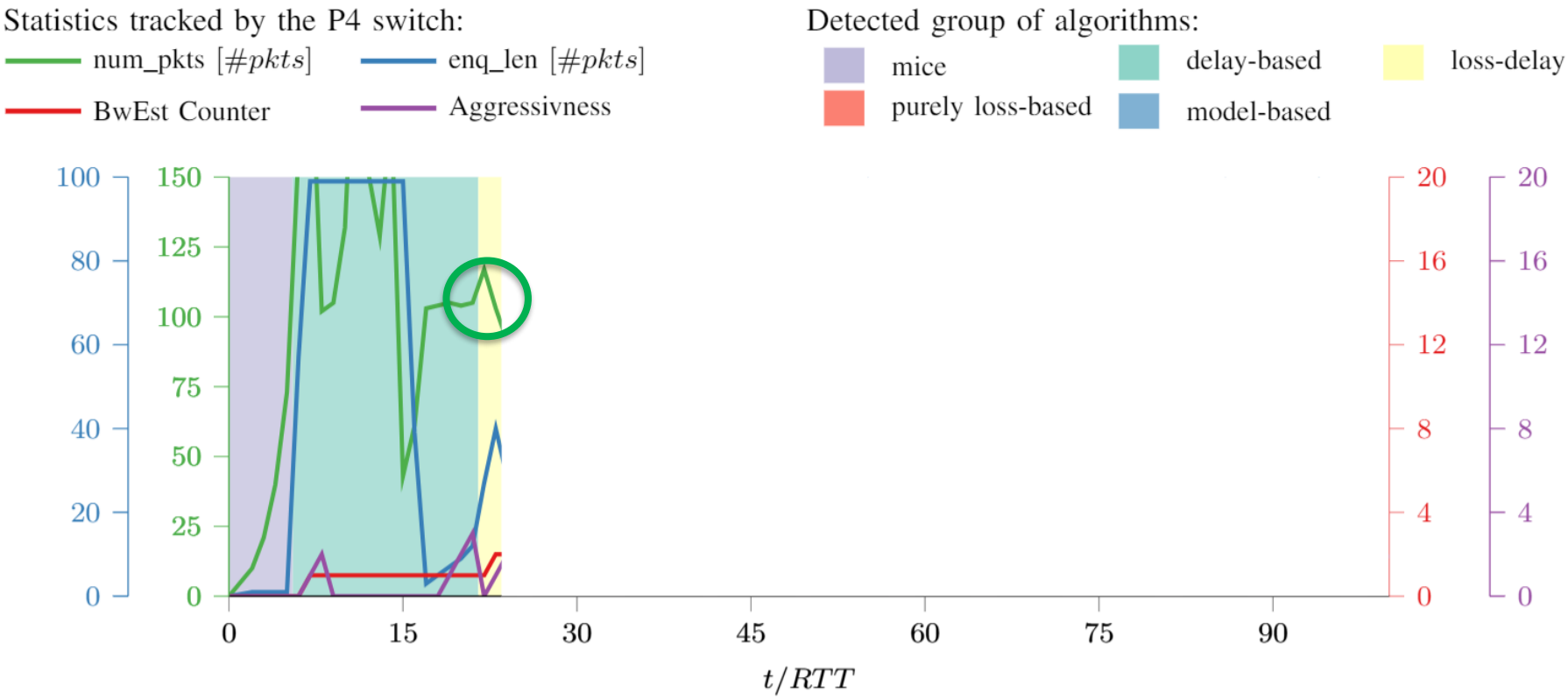
Fingerprinting - BBR



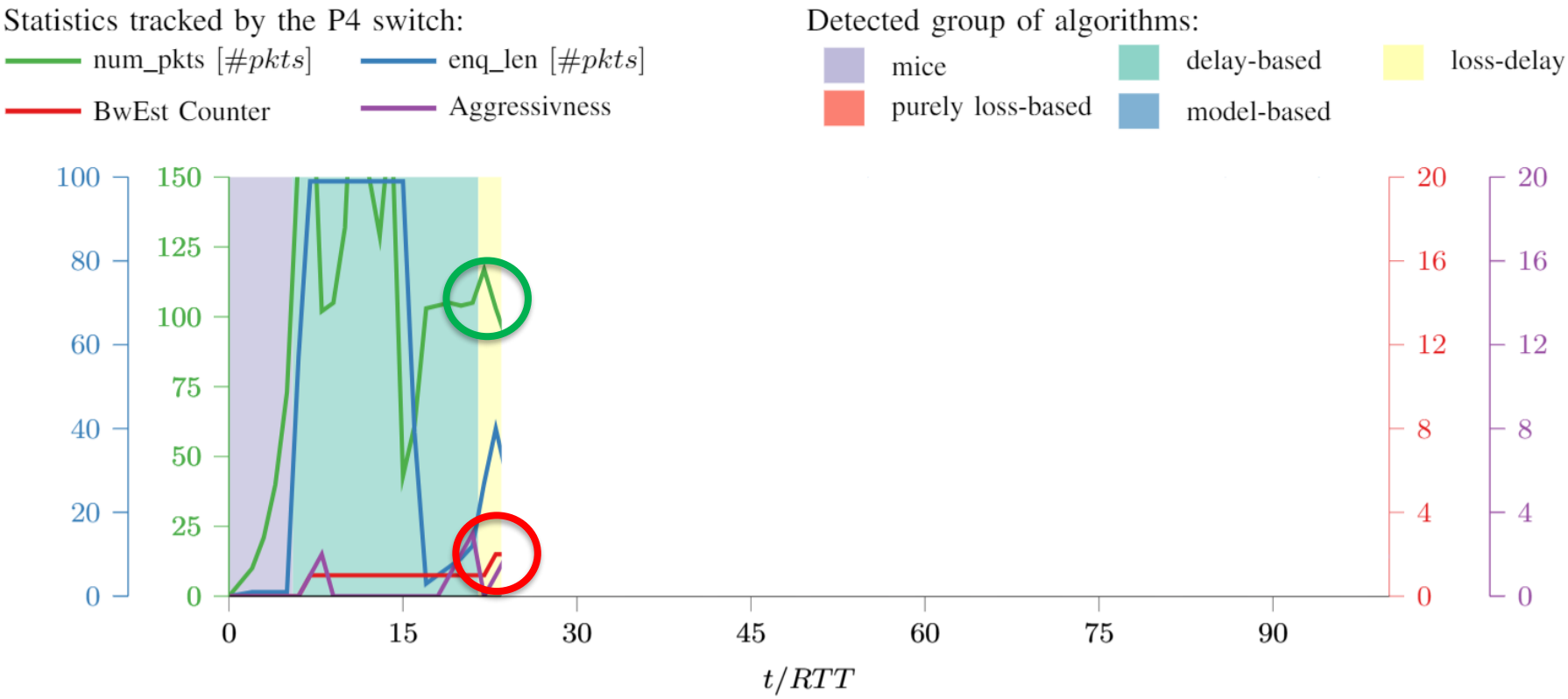
Fingerprinting - BBR



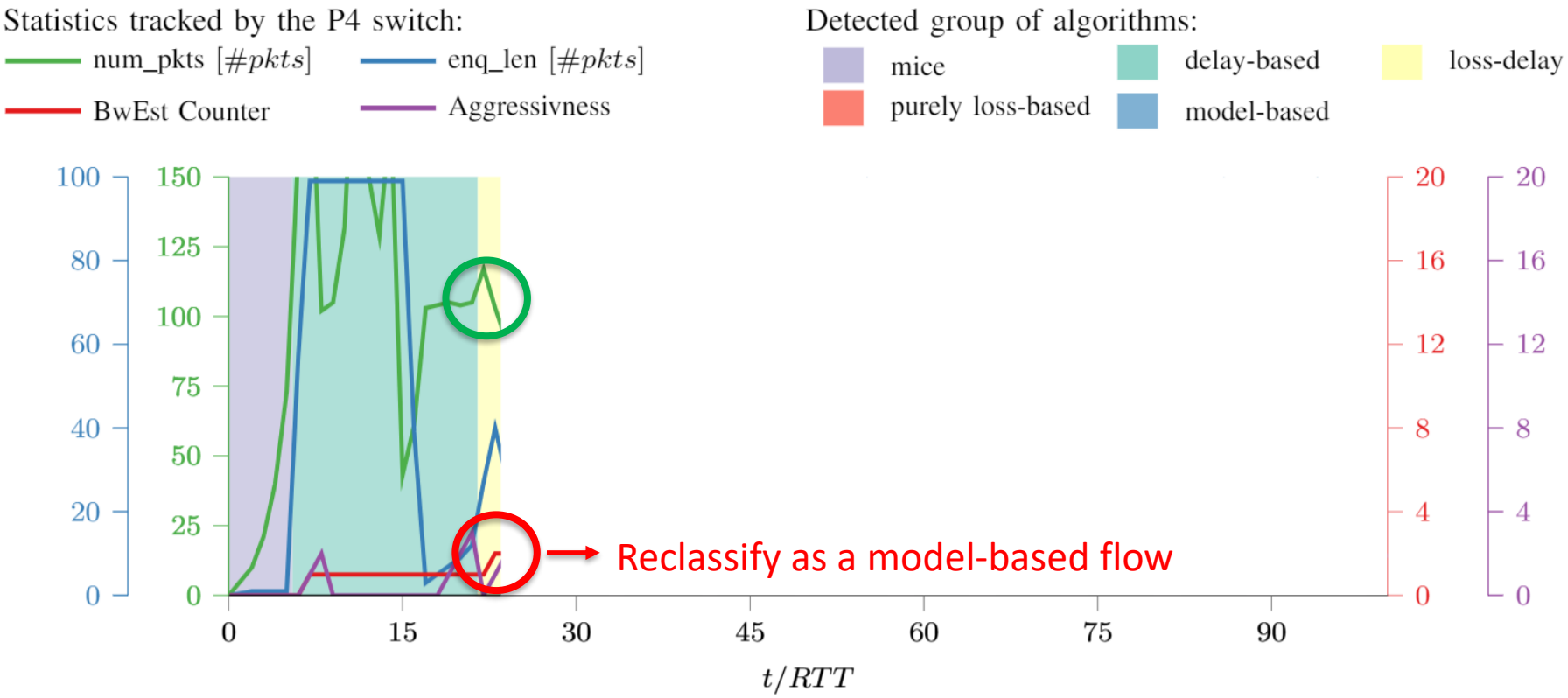
Fingerprinting - BBR



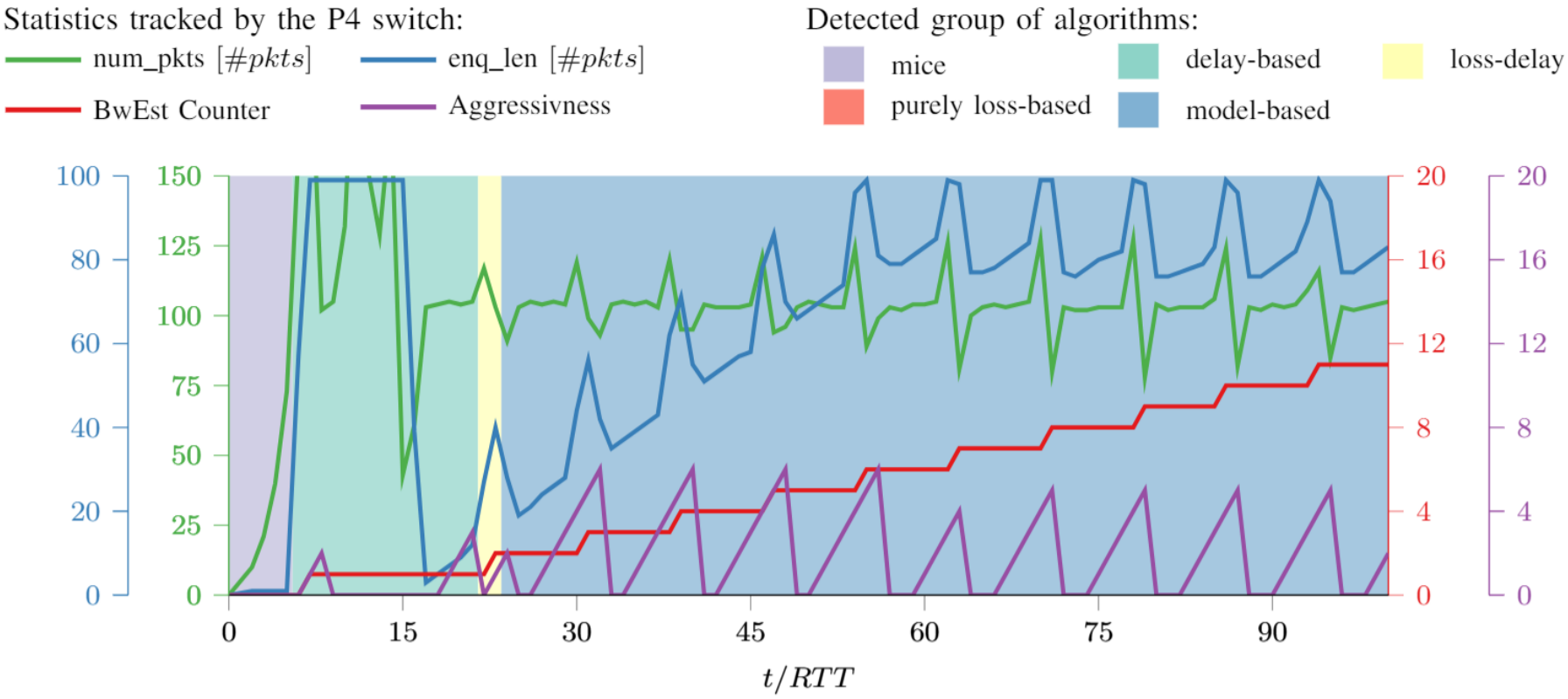
Fingerprinting - BBR



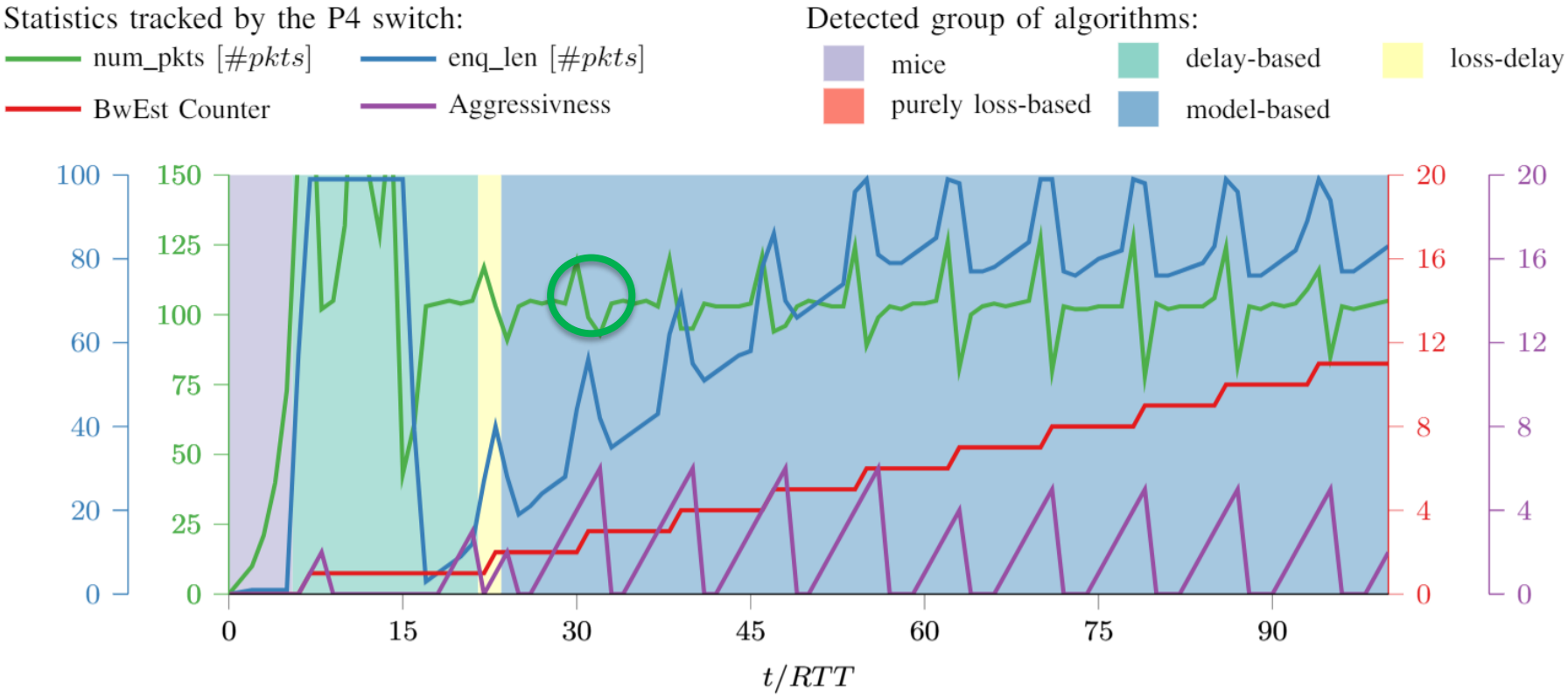
Fingerprinting - BBR



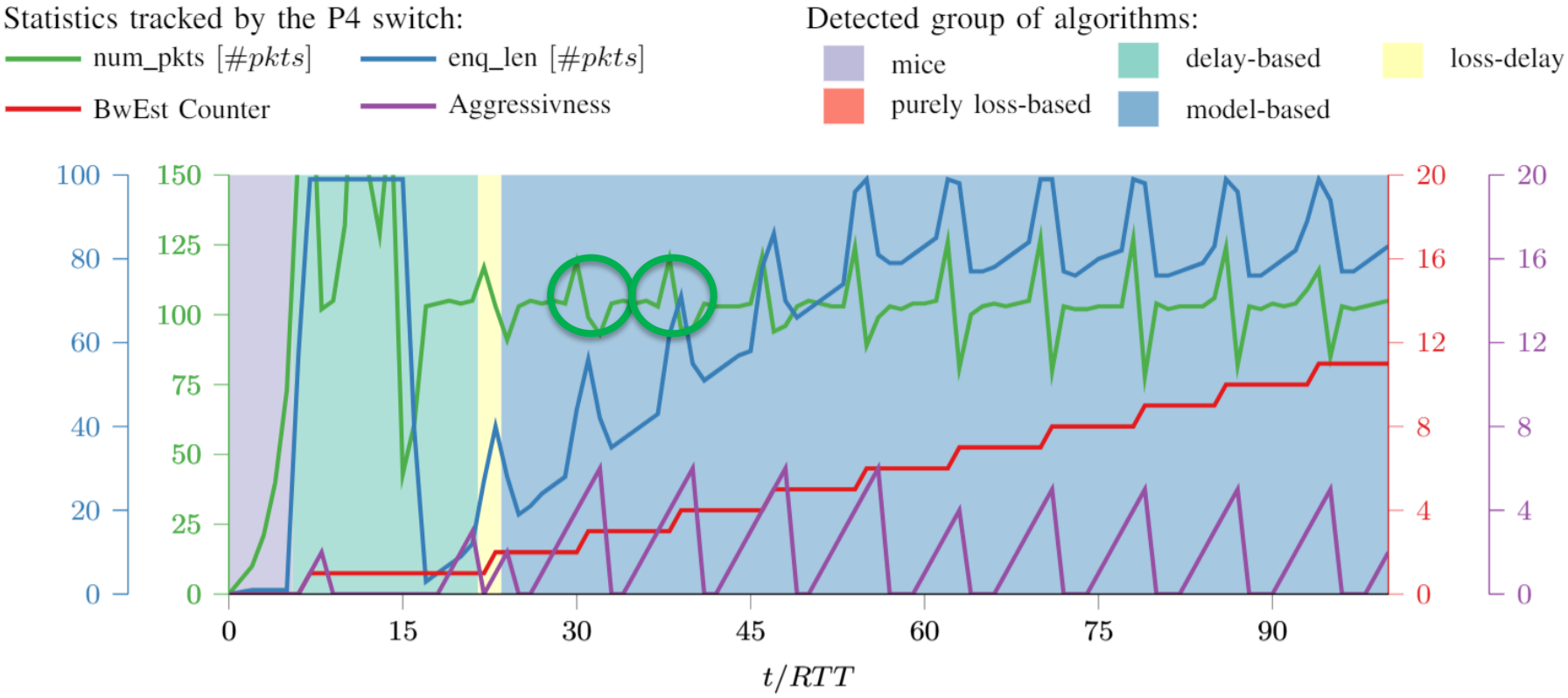
Fingerprinting - BBR



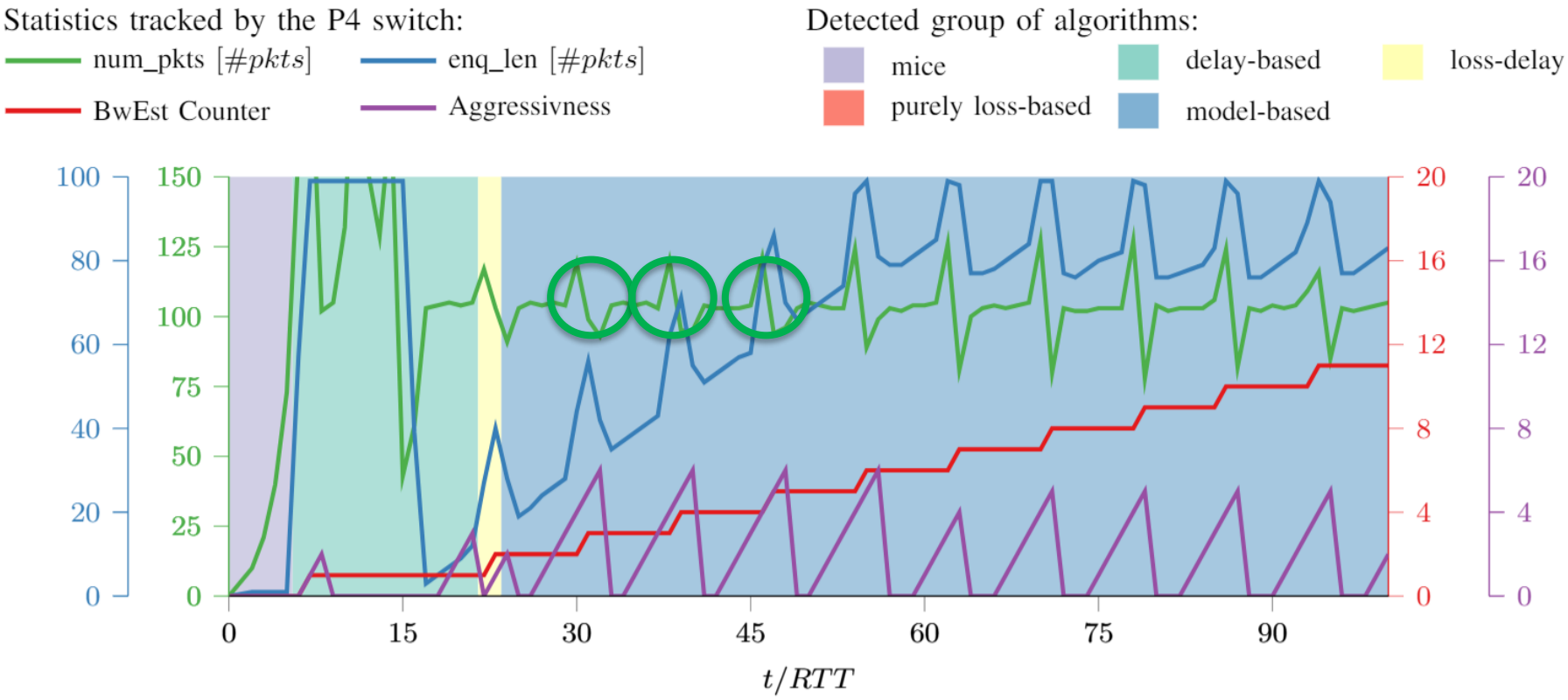
Fingerprinting - BBR



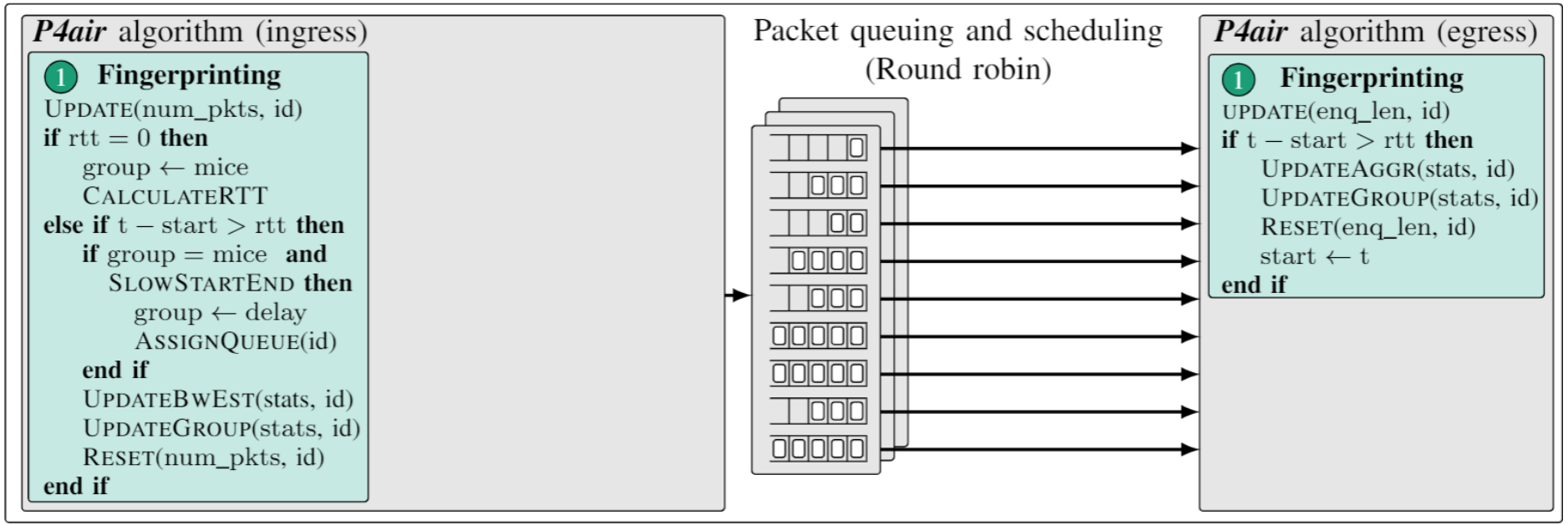
Fingerprinting - BBR



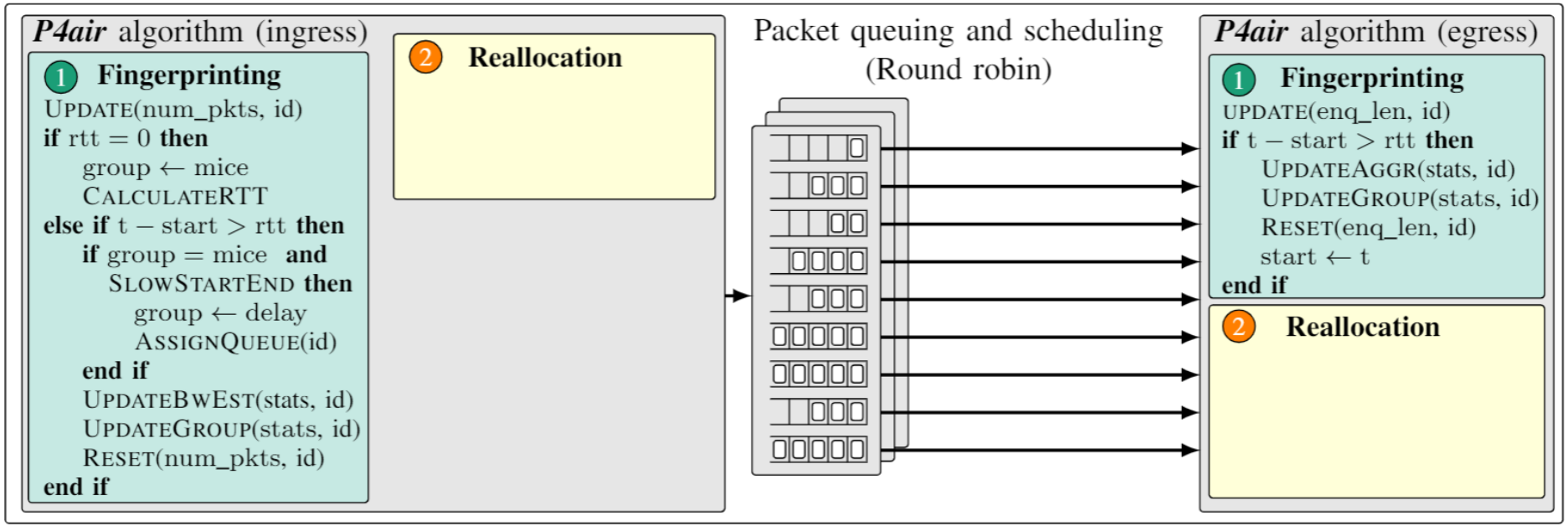
Fingerprinting - BBR



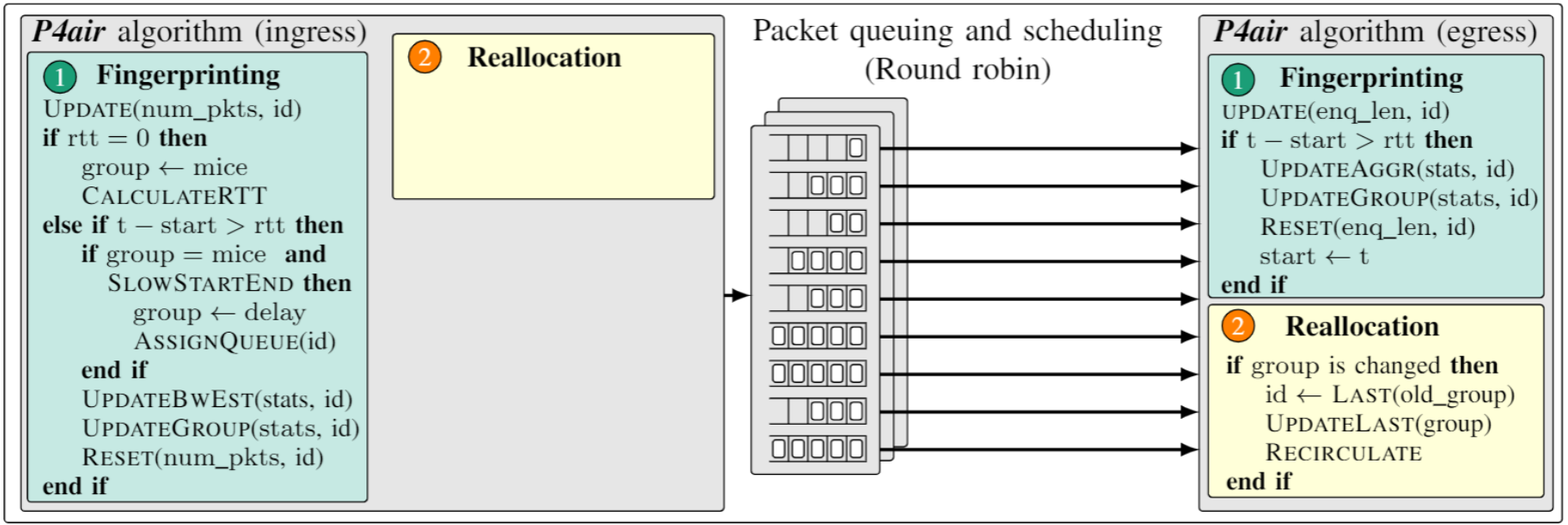
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if

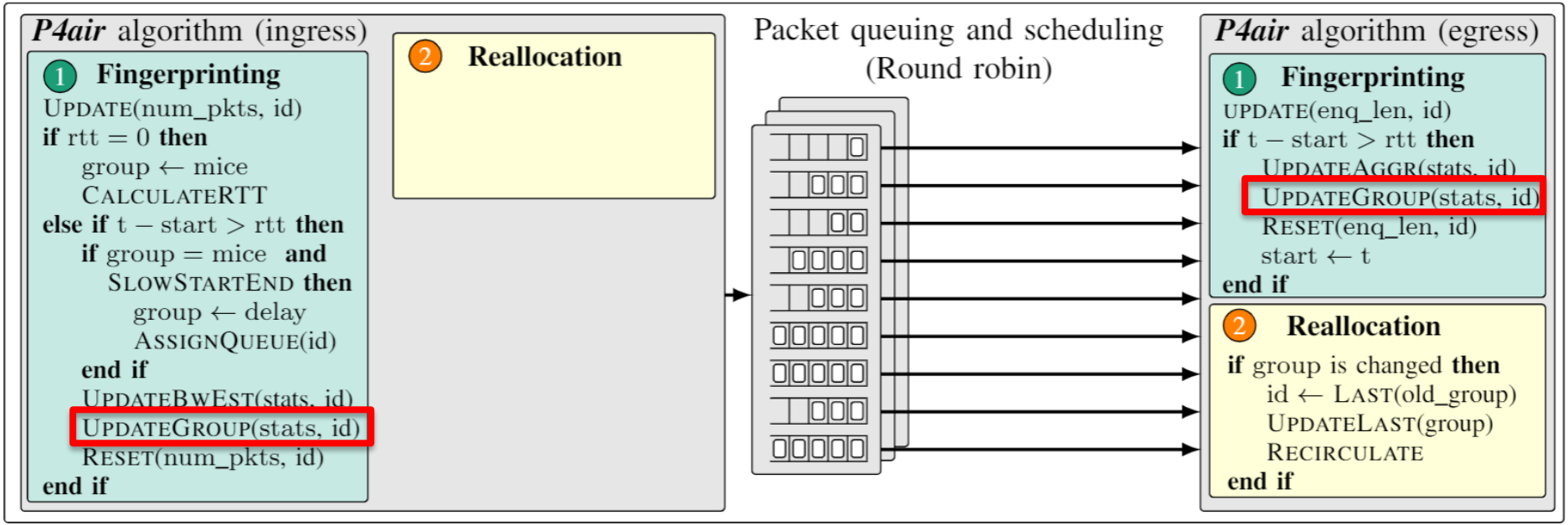
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation

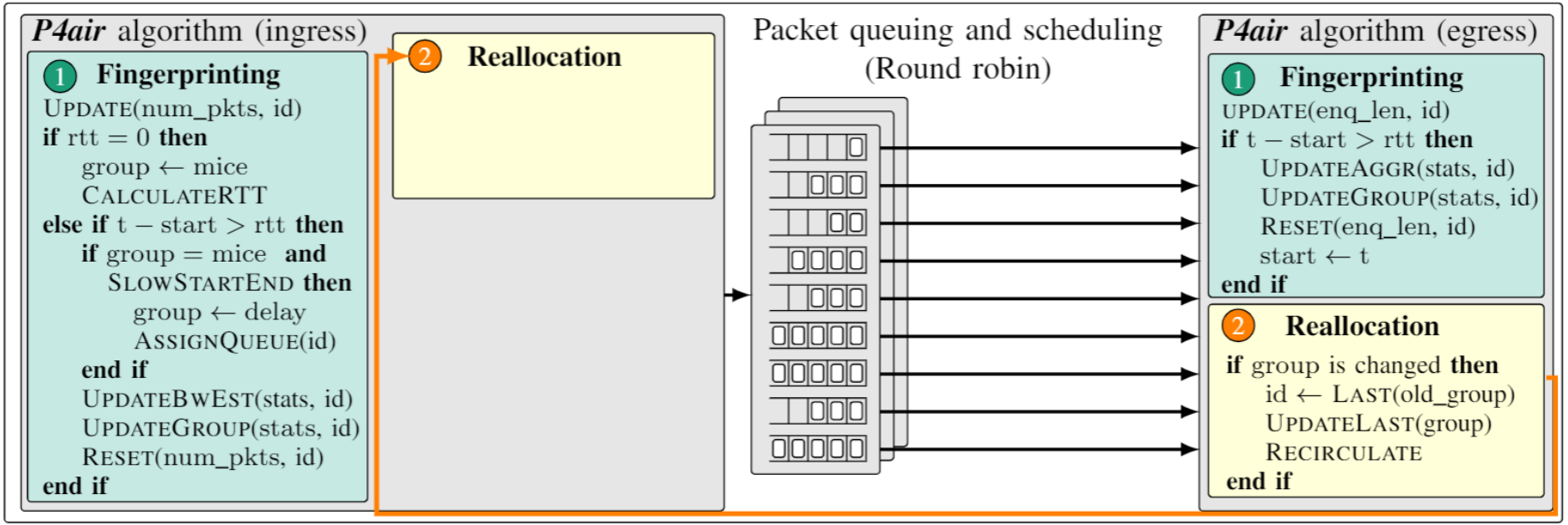
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

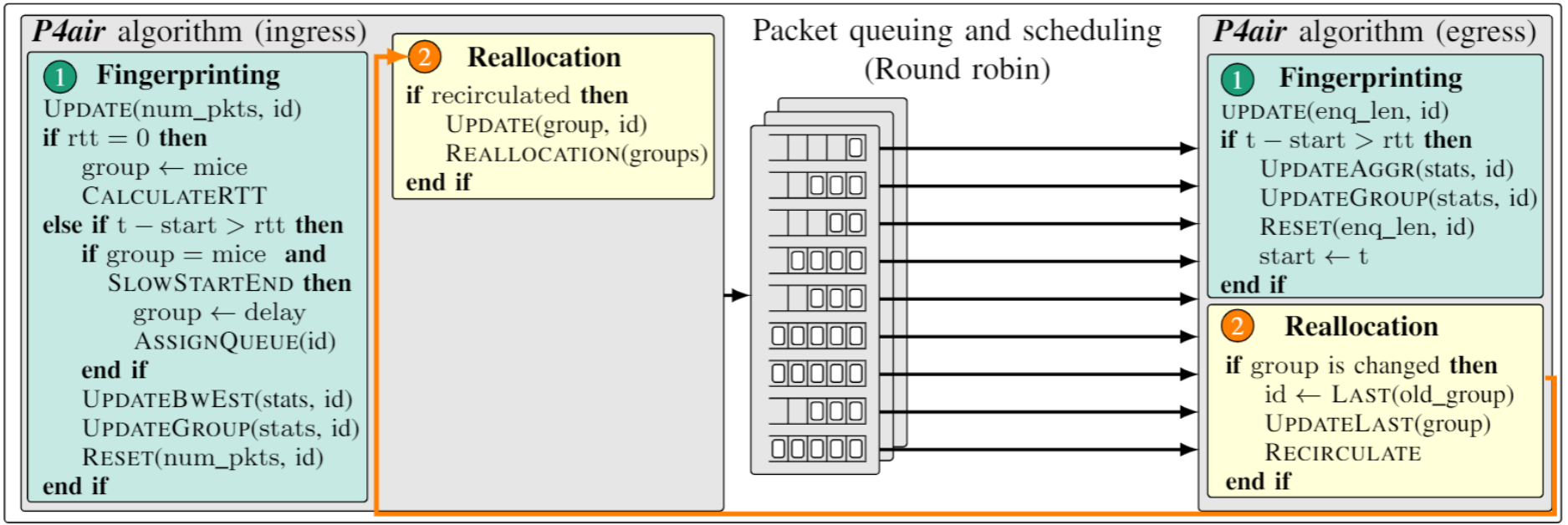
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

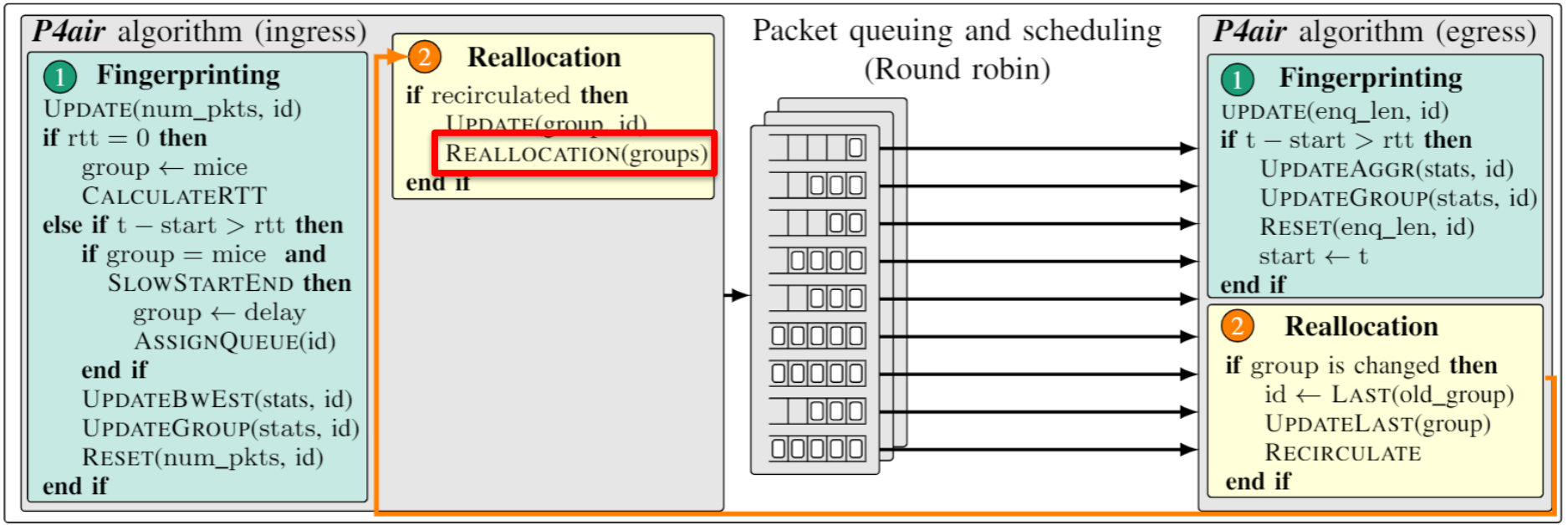
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

P4air

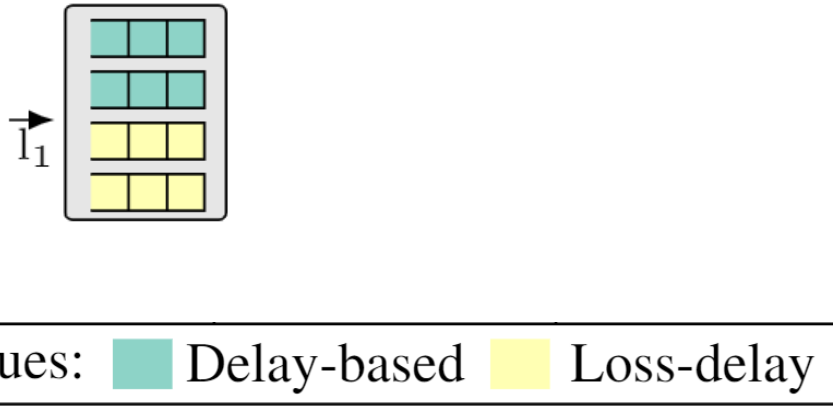

P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

P4air

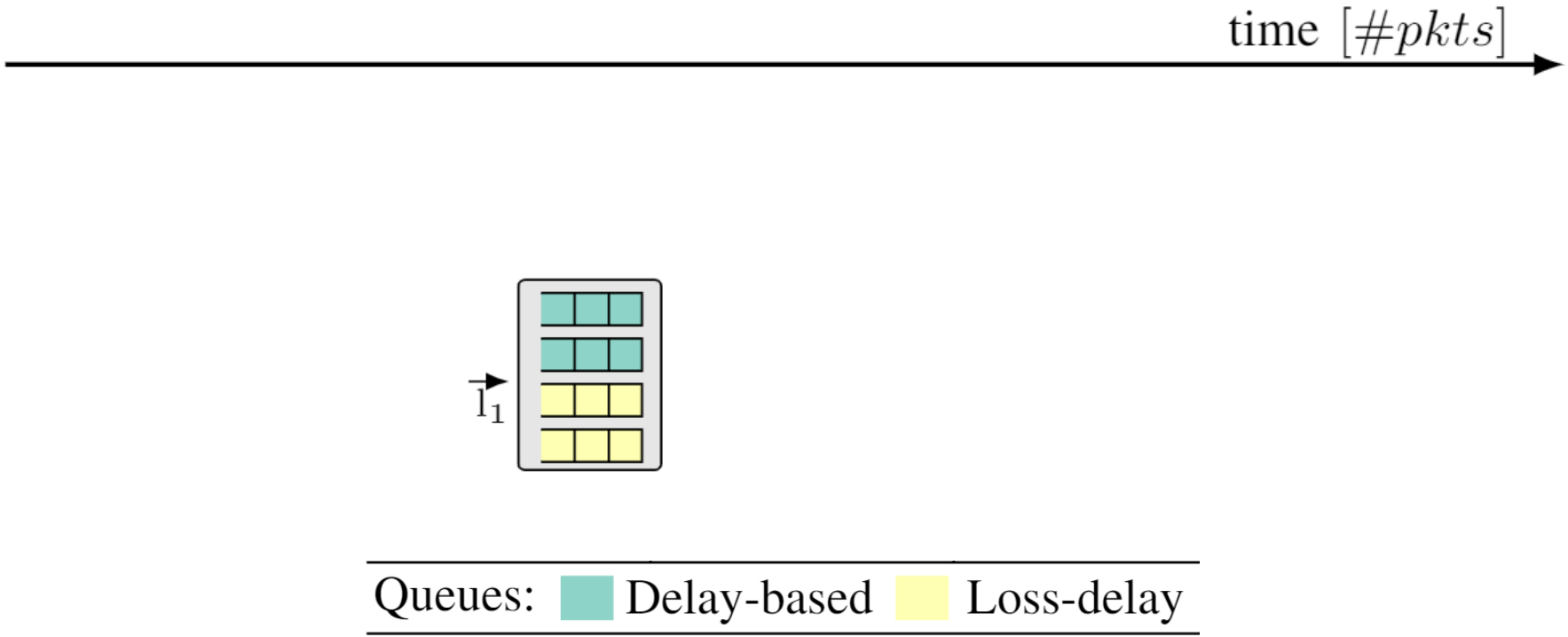

P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

Reallocation

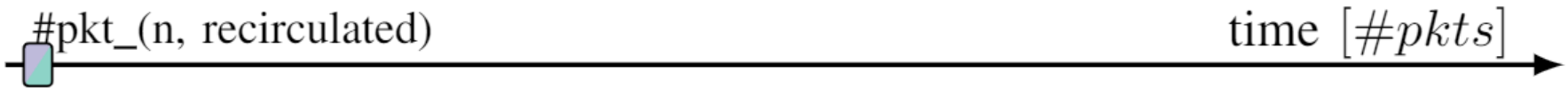
Reallocation

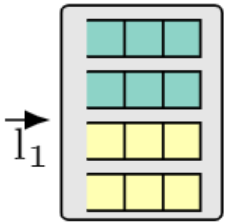


Reallocation



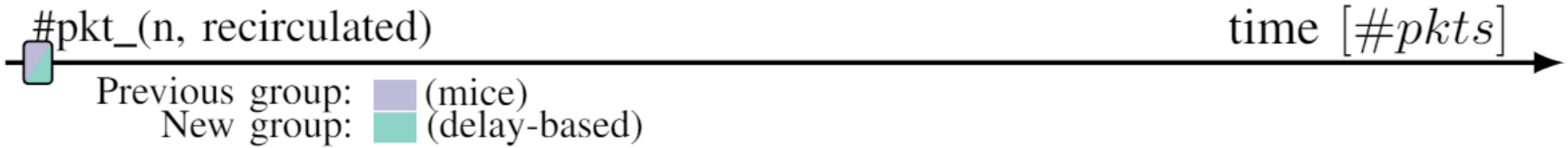
Reallocation

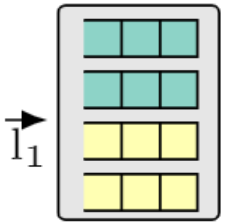


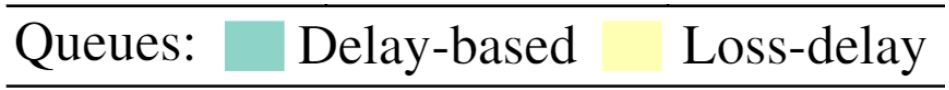


Queues: Delay-based Loss-delay

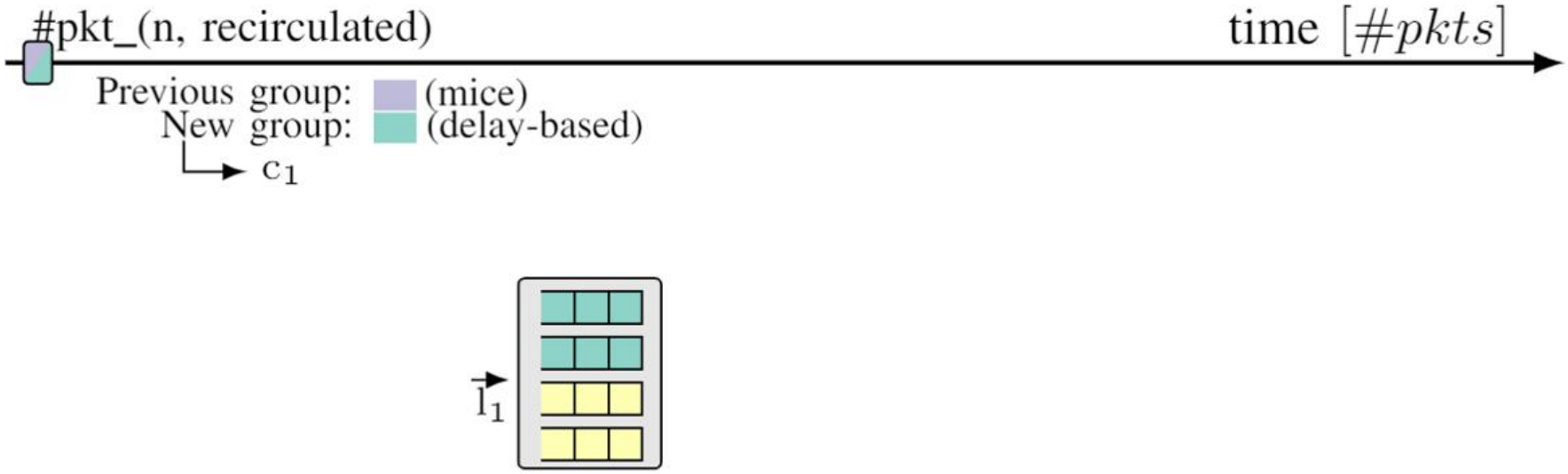
Reallocation



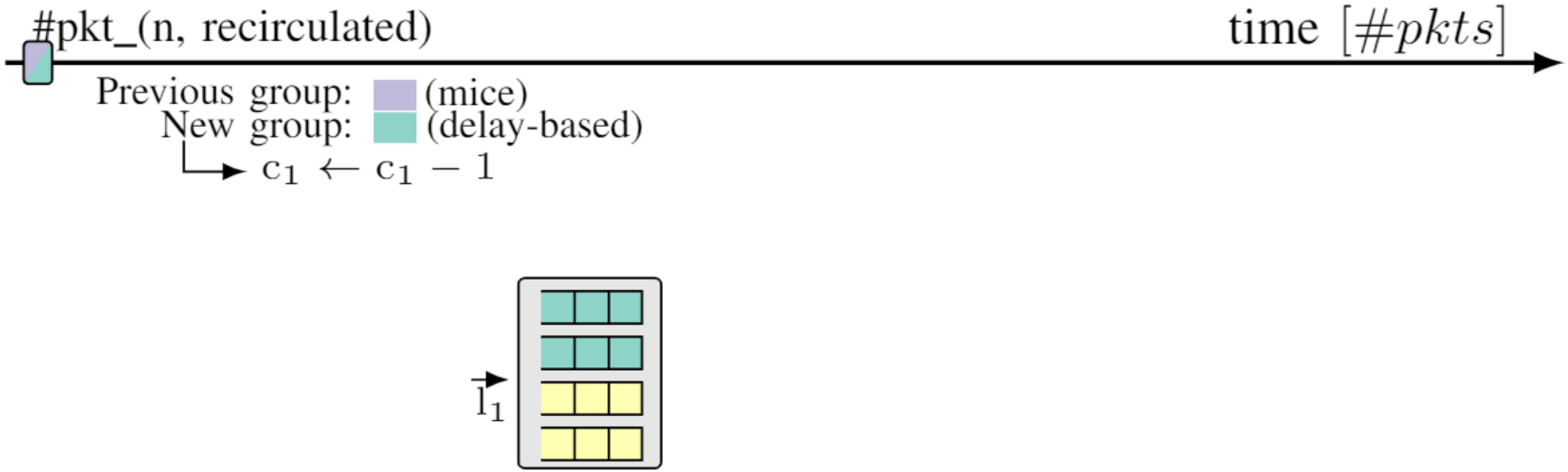




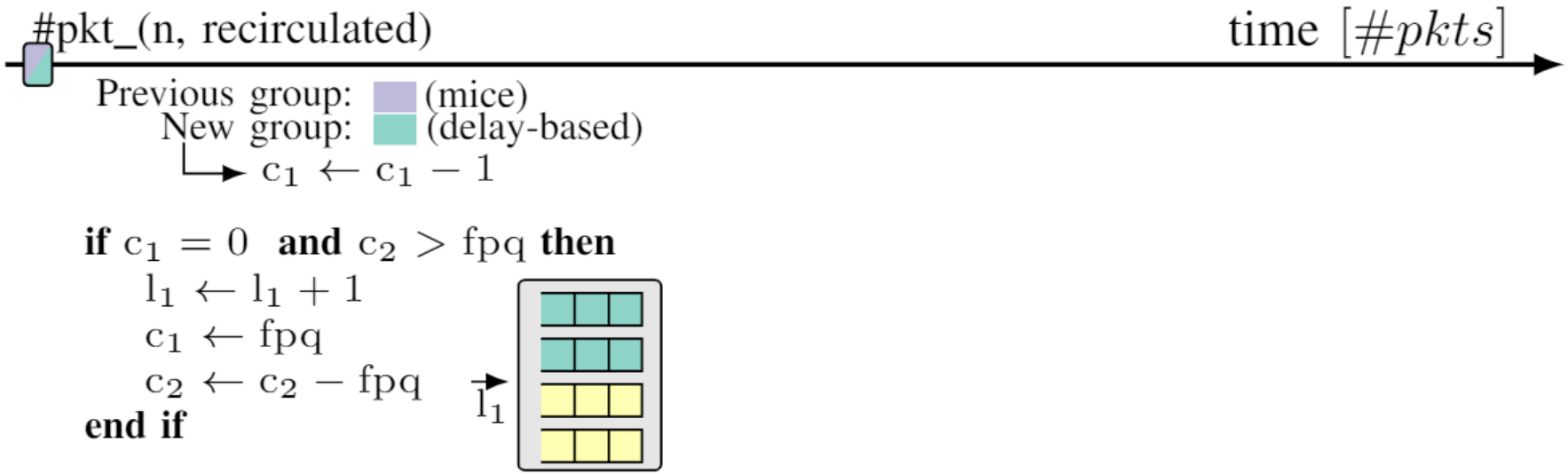
Reallocation



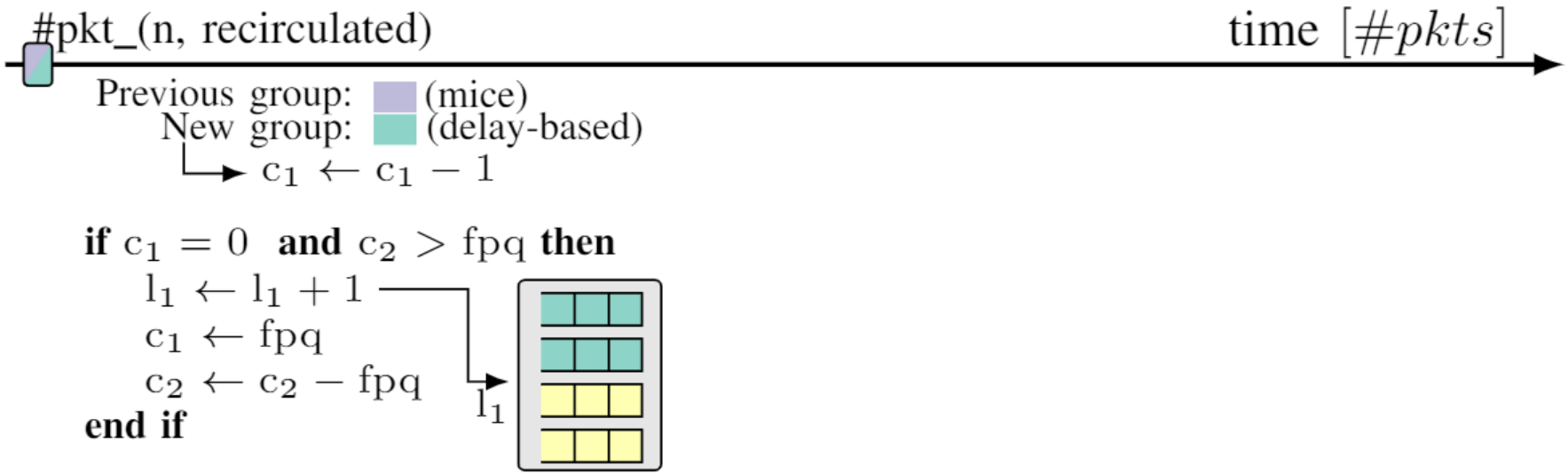
Reallocation



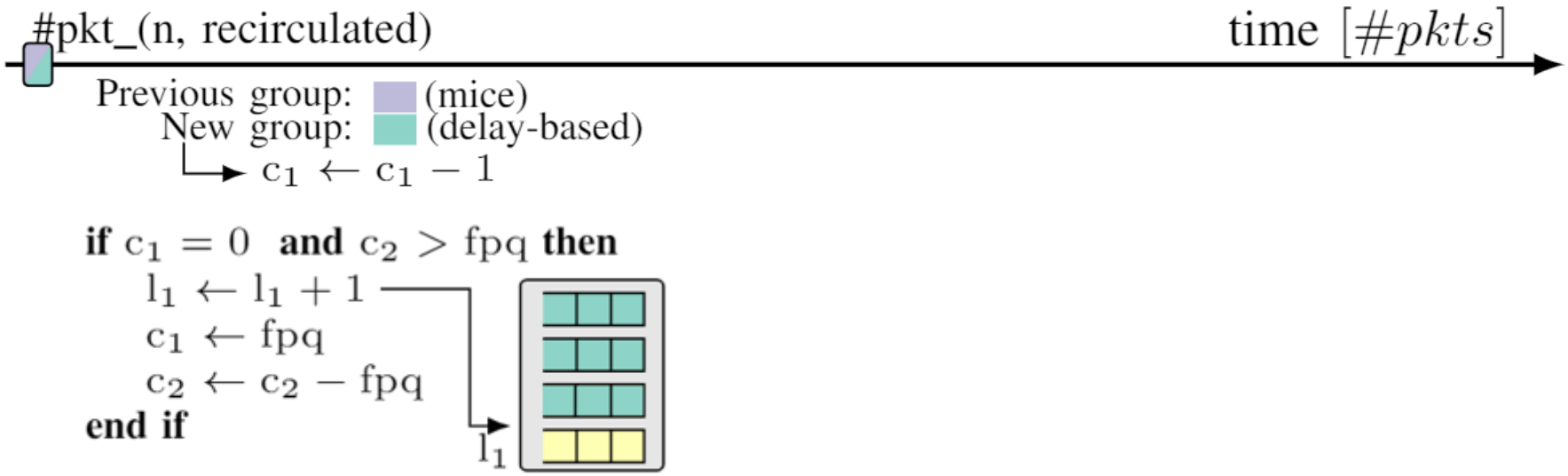
Reallocation



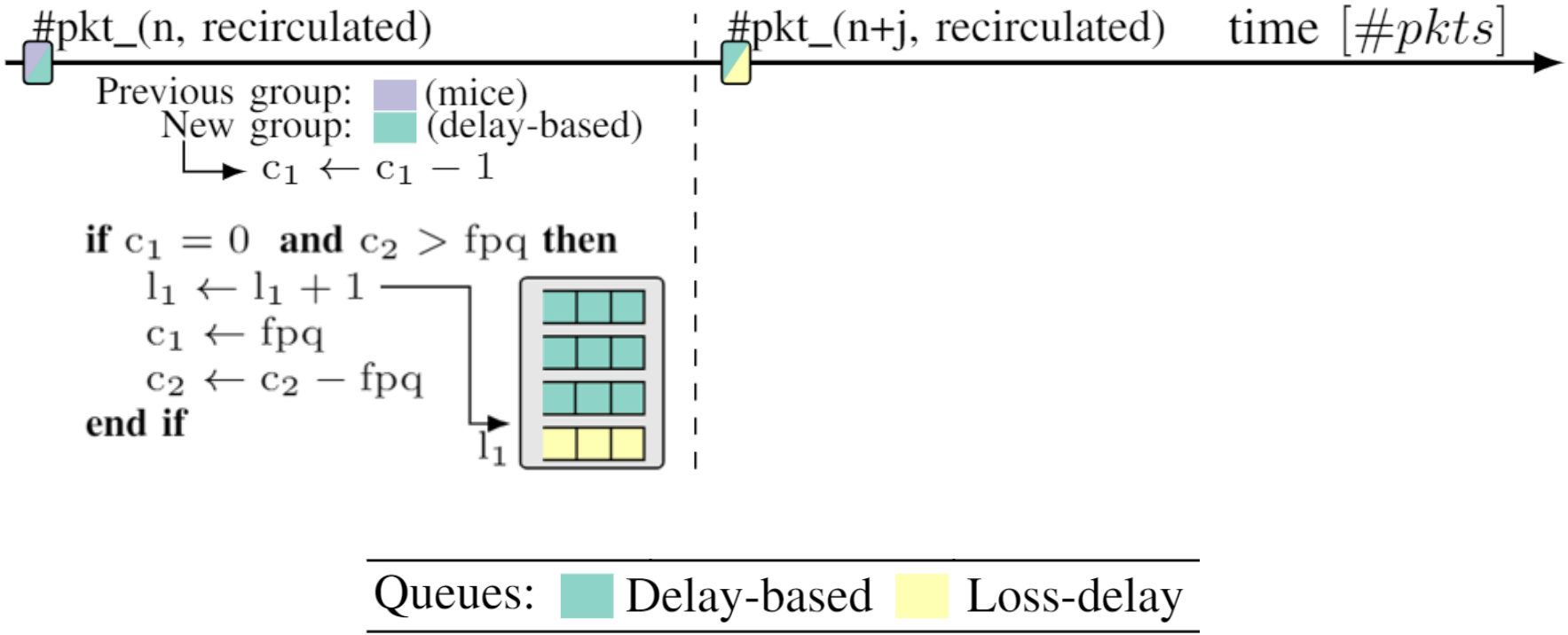
Reallocation



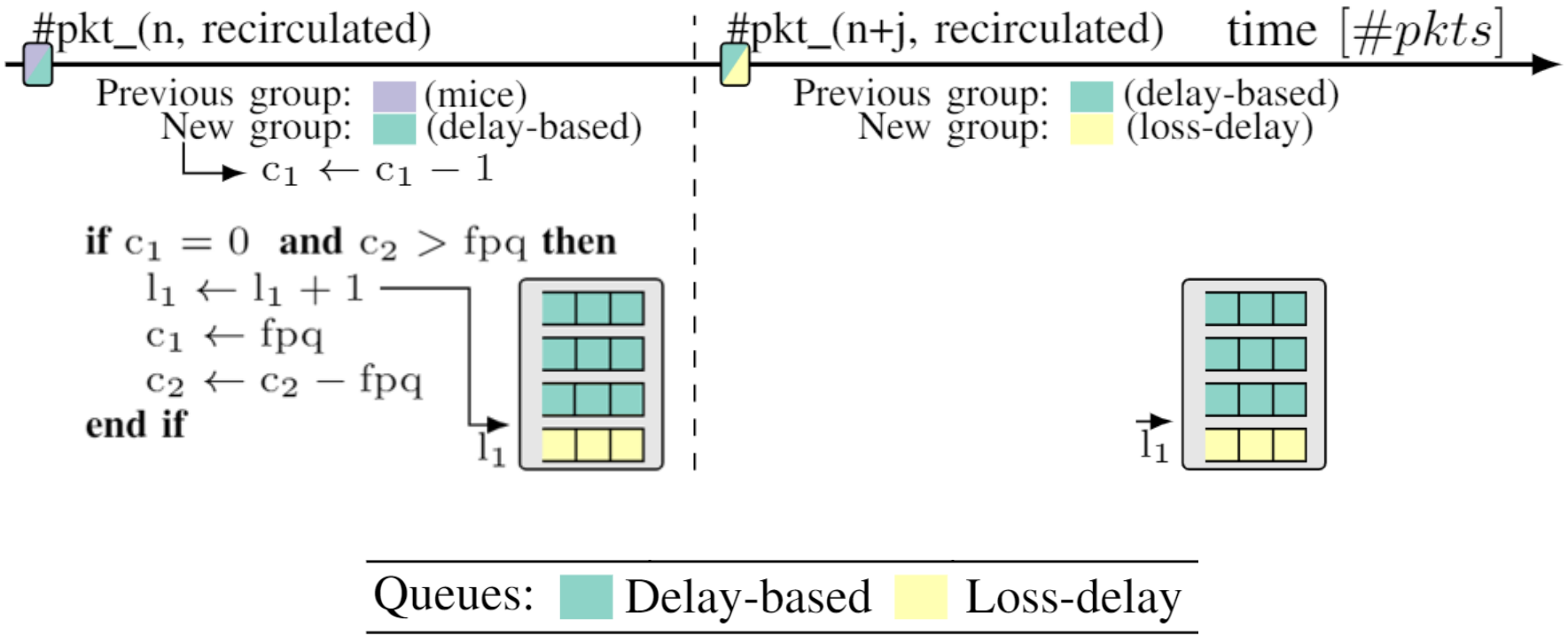
Reallocation



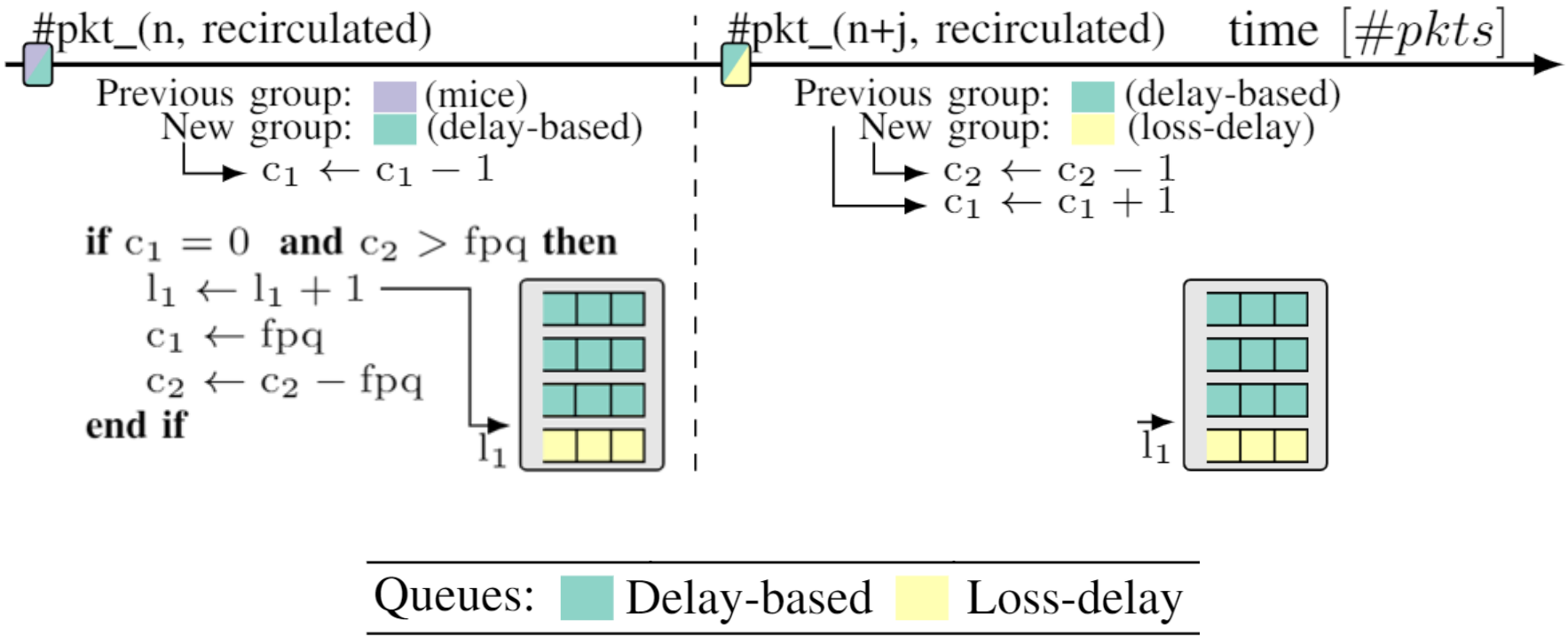
Reallocation



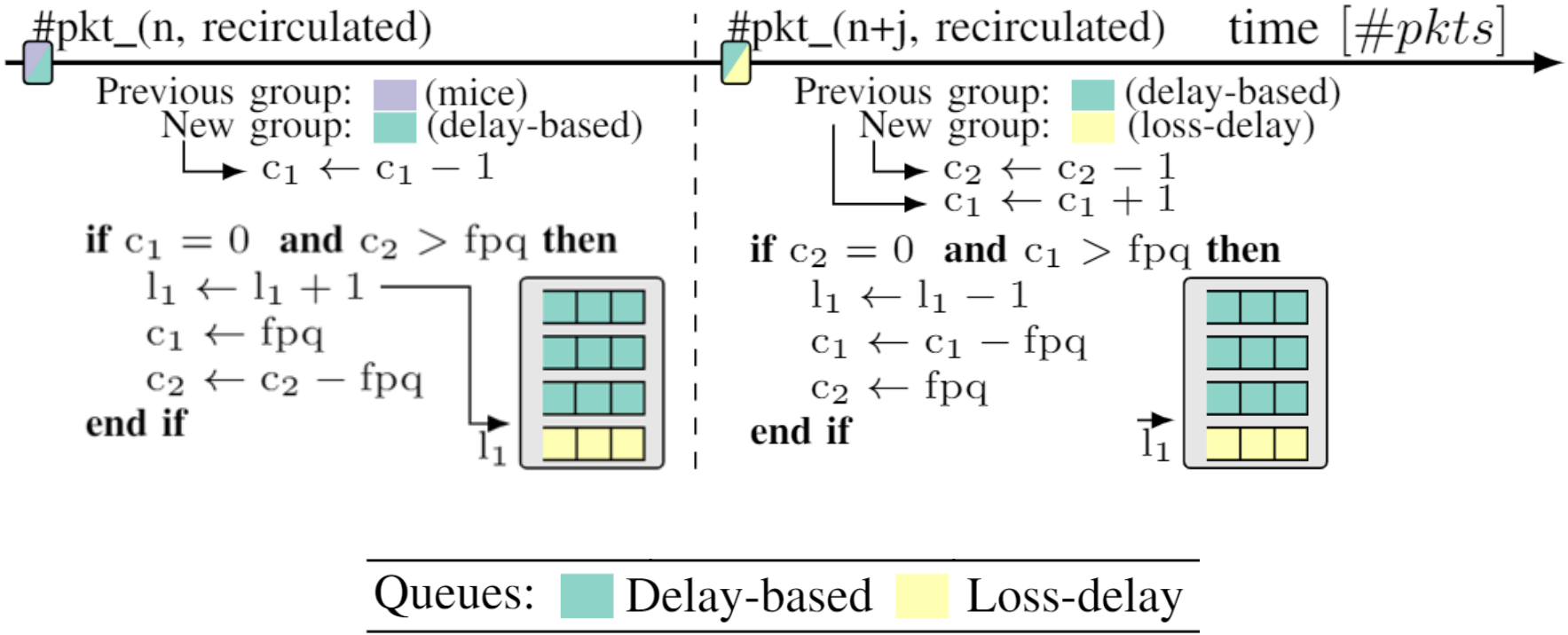
Reallocation



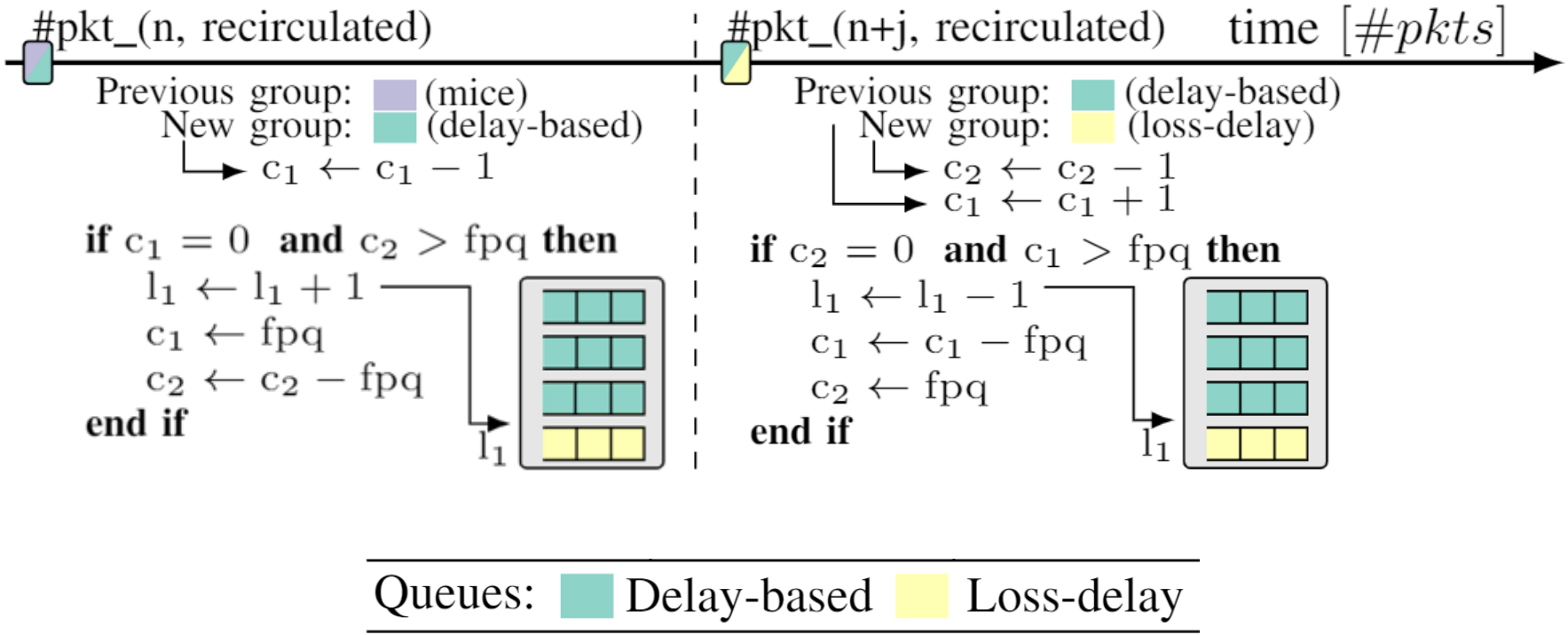
Reallocation



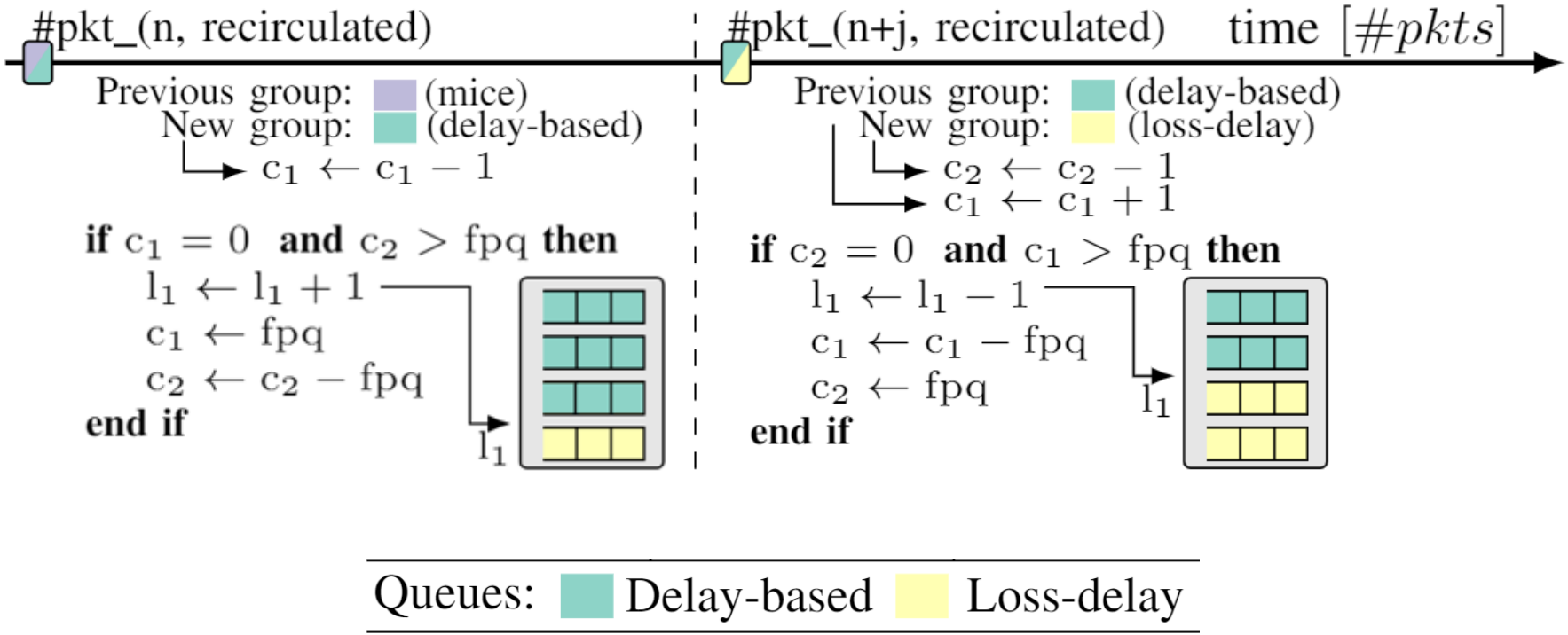
Reallocation



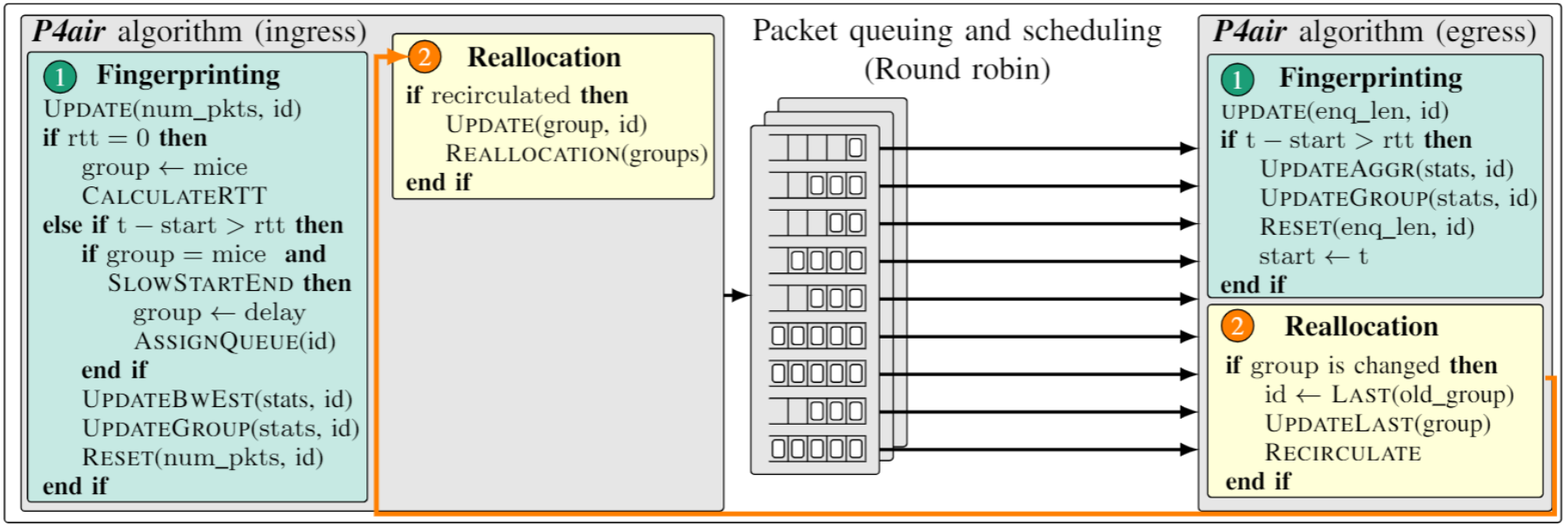
Reallocation



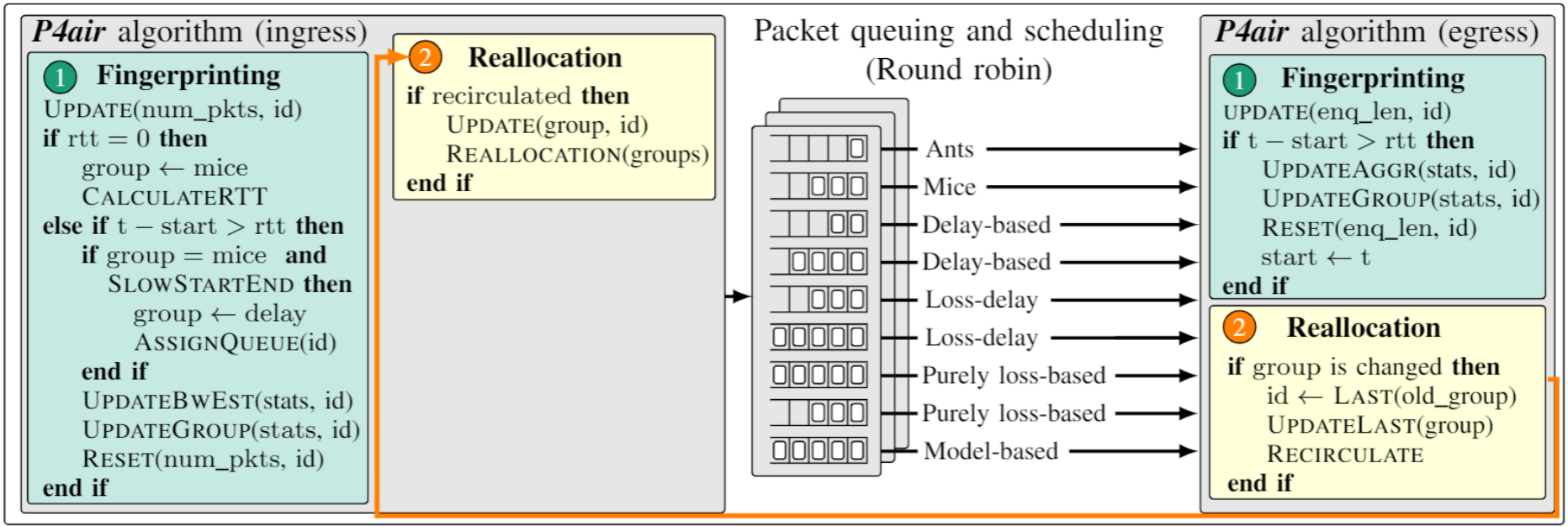
Reallocation



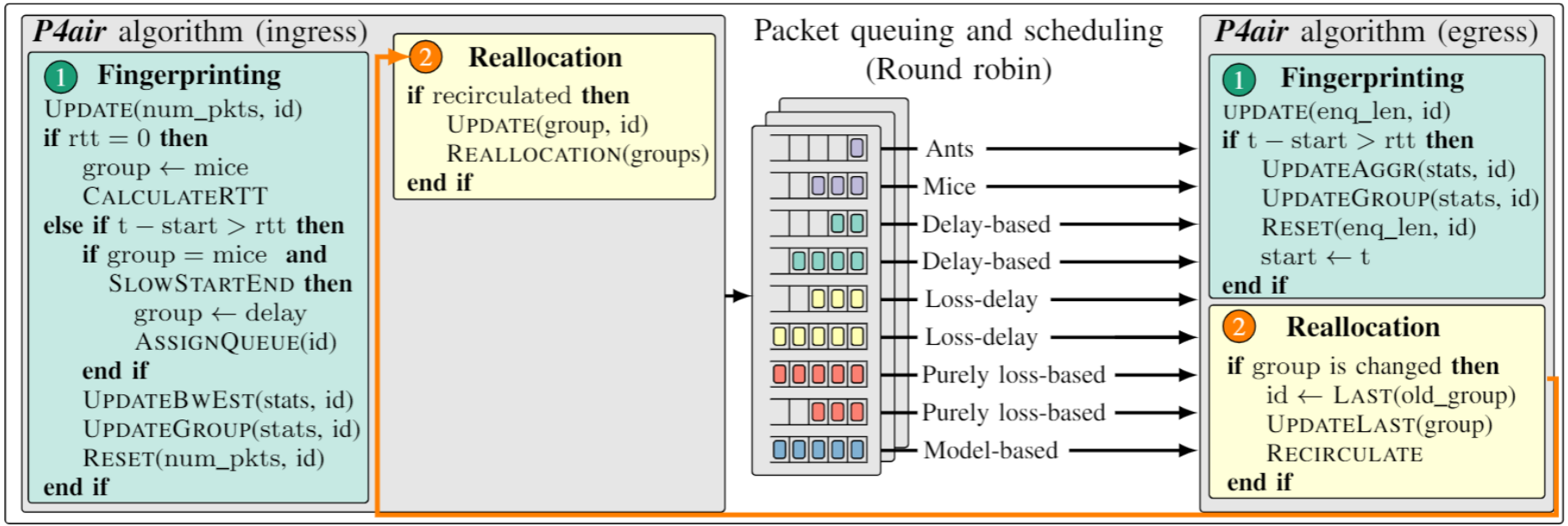
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
Packet queuing and scheduling
(Round robin)
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

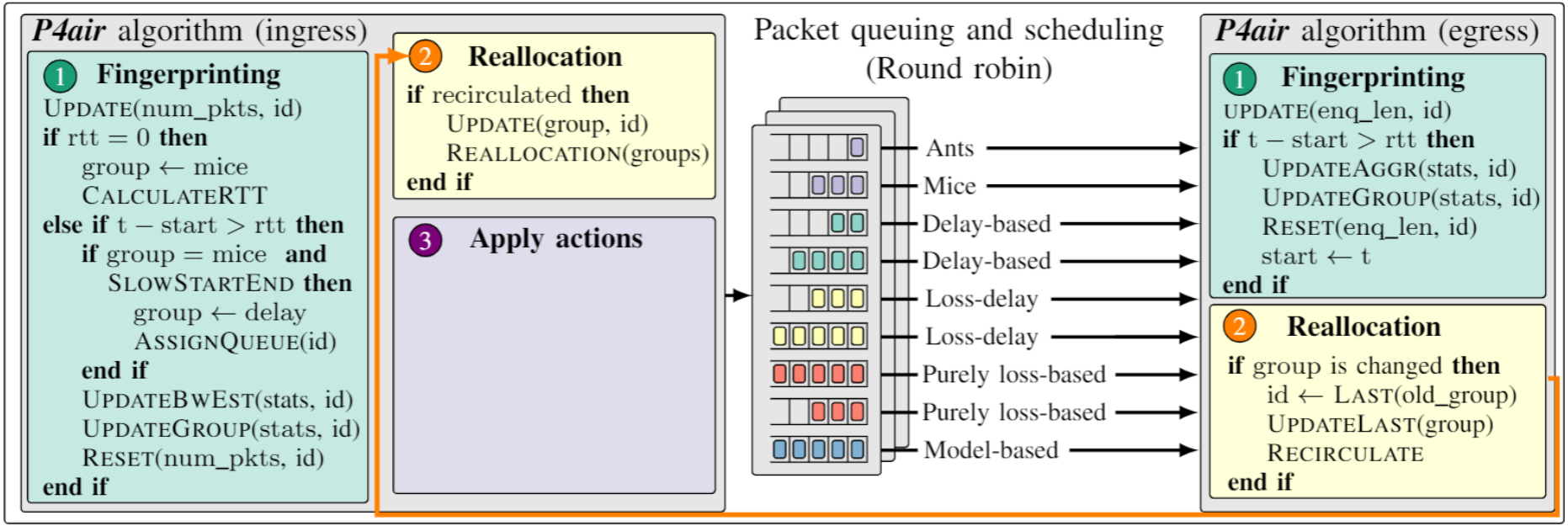
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
Packet queuing and scheduling
(Round robin)
Ants
Mice
Delay-based
Delay-based
Loss-delay
Loss-delay
Purely loss-based
Purely loss-based
Model-based
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

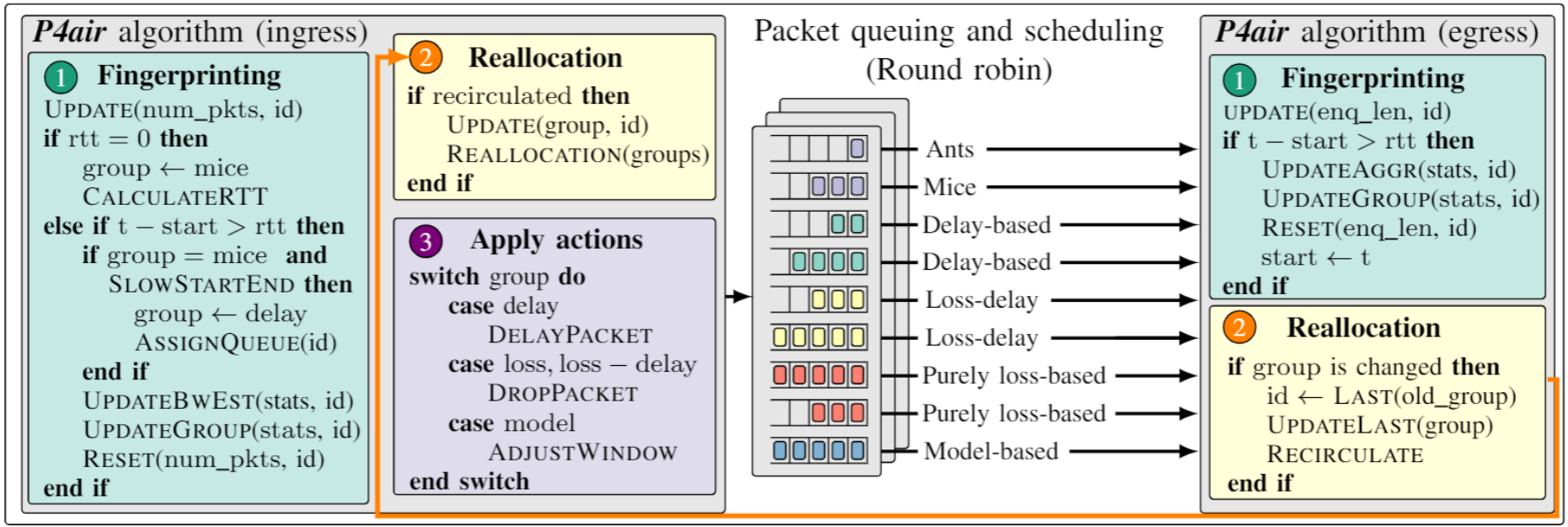
P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
Packet queuing and scheduling
(Round robin)
Ants
Mice
Delay-based
Delay-based
Loss-delay
Loss-delay
Purely loss-based
Purely loss-based
Model-based
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
3 Apply actions
Packet queuing and scheduling
(Round robin)
Ants
Mice
Delay-based
Delay-based
Loss-delay
Loss-delay
Purely loss-based
Purely loss-based
Model-based
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

P4air


P4air algorithm (ingress)
1 Fingerprinting
UPDATE(num_pkts, id)
if rtt = 0 then
group ← mice
CALCULATERTT
else if t − start > rtt then
if group = mice and
SLOWSTARTEND then
group ← delay
ASSIGNQUEUE(id)
end if
UPDATEBWEST(stats, id)
UPDATEGROUP(stats, id)
RESET(num_pkts, id)
end if
2 Reallocation
if recirculated then
UPDATE(group, id)
REALLOCATION(groups)
end if
3 Apply actions
switch group do
case delay
DELAYPACKET
case loss, loss − delay
DROPPACKET
case model
ADJUSTWINDOW
end switch
Packet queuing and scheduling
(Round robin)
Ants
Mice
Delay-based
Delay-based
Loss-delay
Loss-delay
Purely loss-based
Purely loss-based
Model-based
P4air algorithm (egress)
1 Fingerprinting
UPDATE(enq_len, id)
if t − start > rtt then
UPDATEAGGR(stats, id)
UPDATEGROUP(stats, id)
RESET(enq_len, id)
start ← t
end if
2 Reallocation
if group is changed then
id ← LAST(old_group)
UPDATELAST(group)
RECIRCULATE
end if

Evaluation

Evaluation

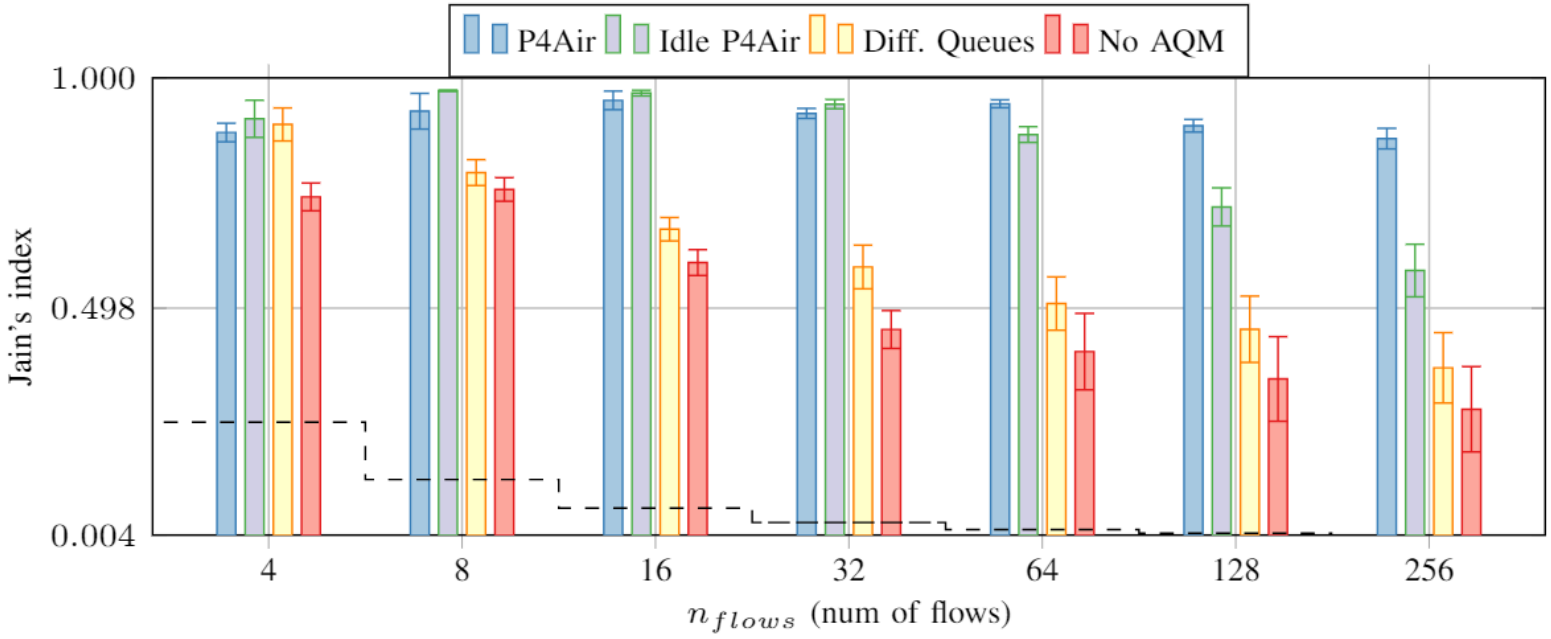
- Using Mininet
- Using a Barefoot switch

Evaluation

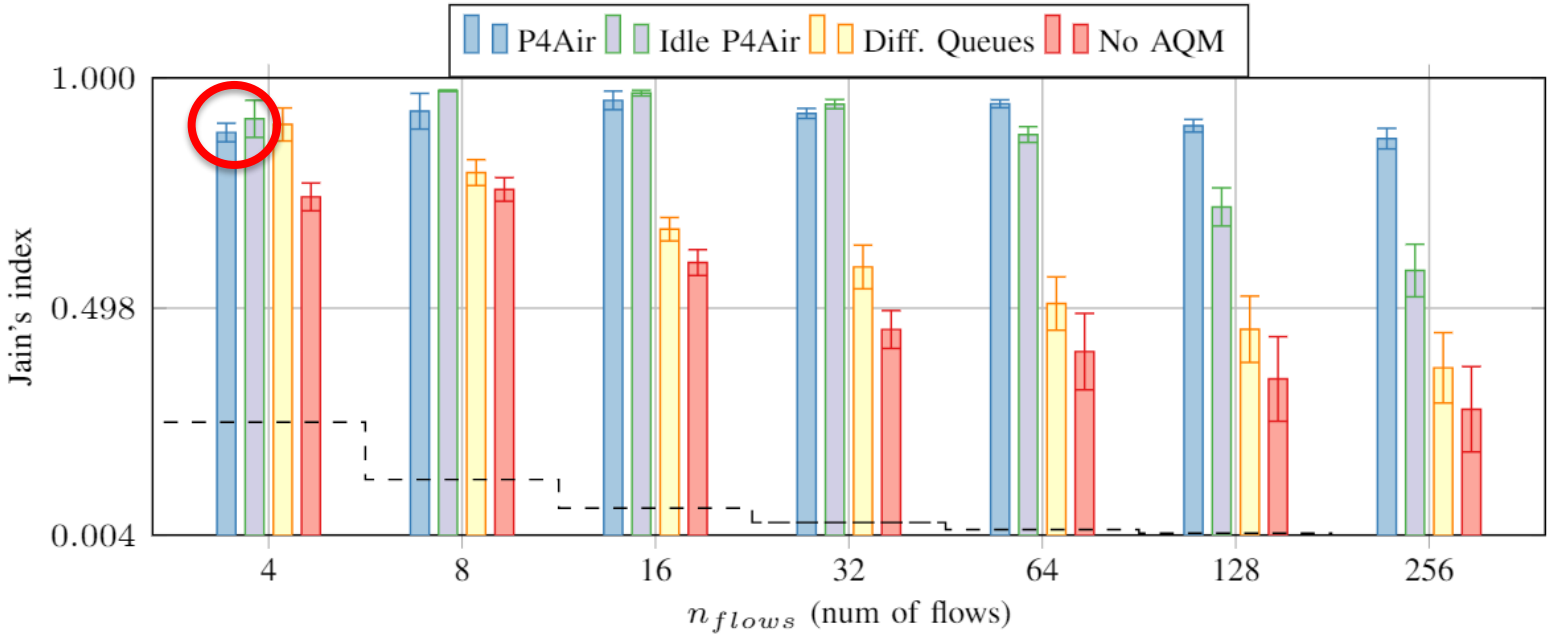
- Using Mininet
- Using a Barefoot switch

- Details on tuning in the paper!

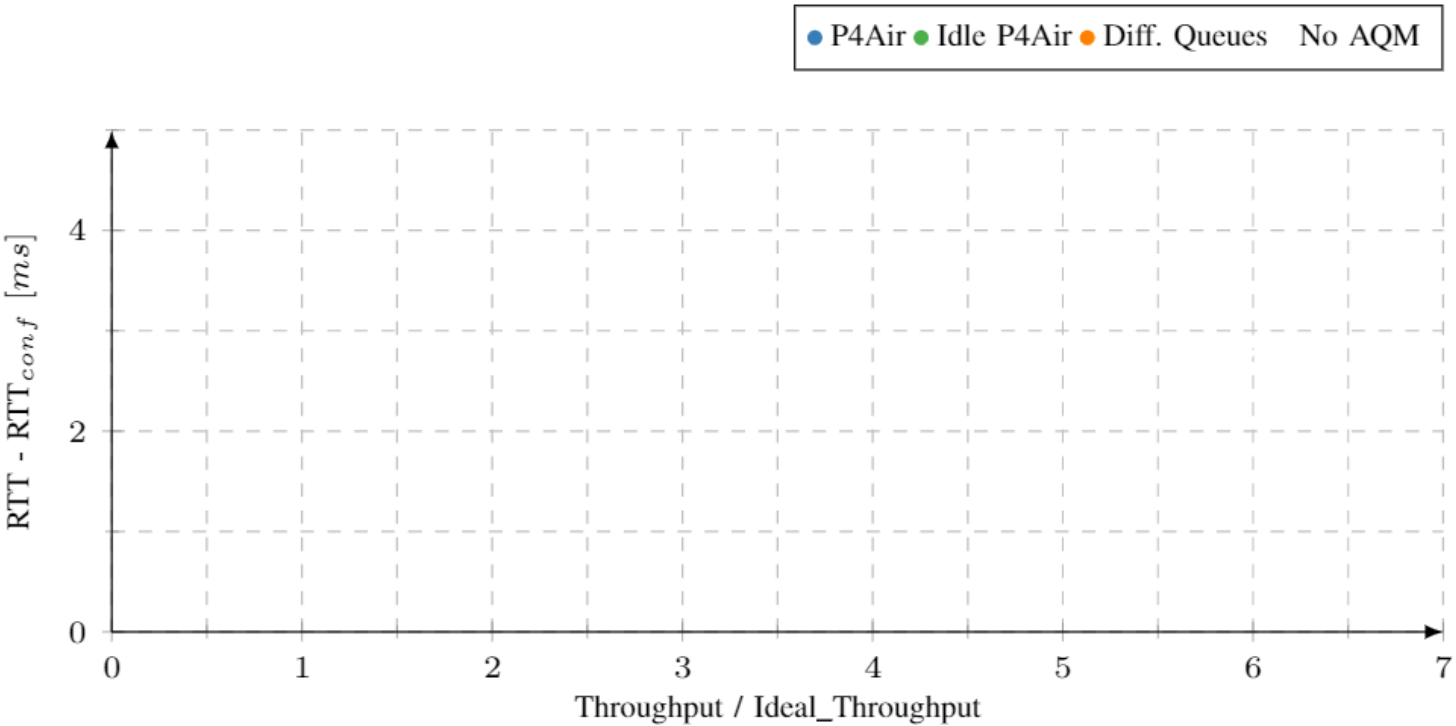
Evaluation – inter- and intra-fairness



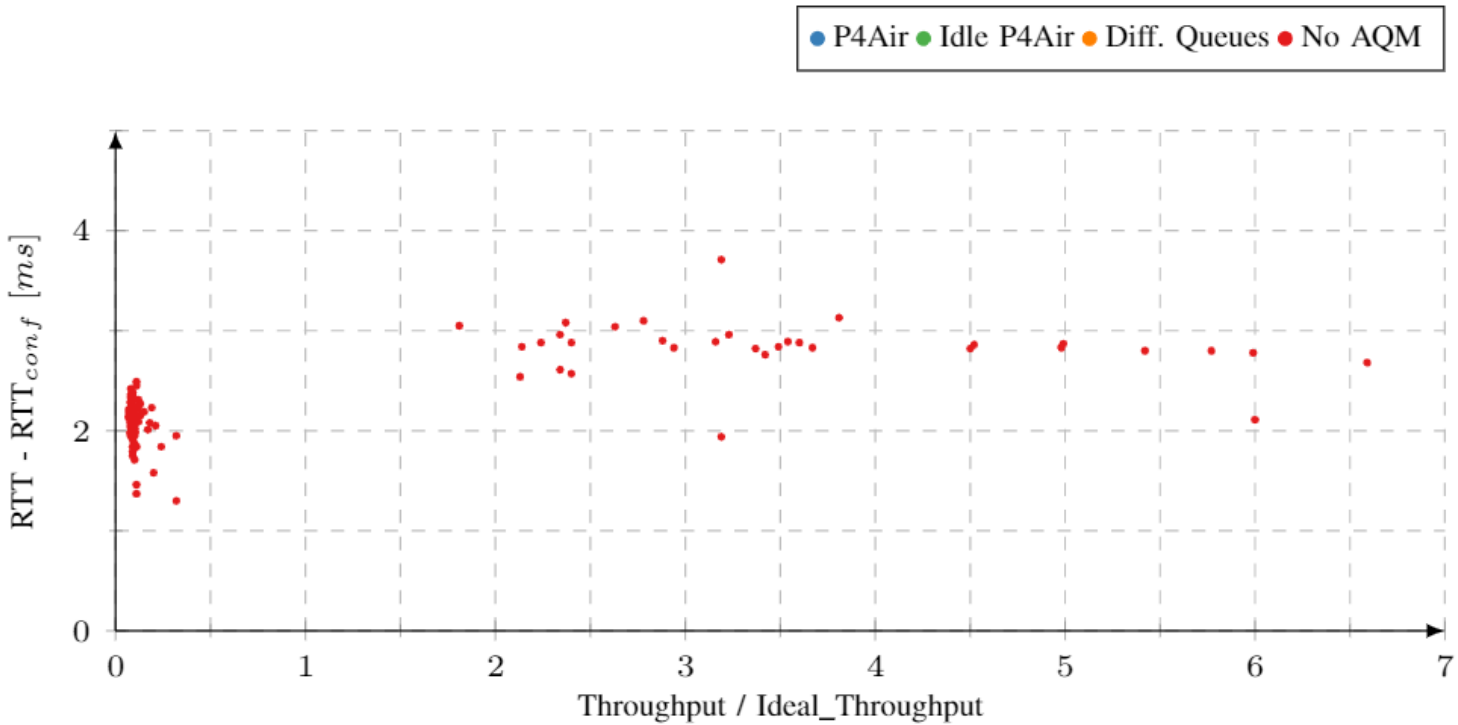
Evaluation – inter- and intra-fairness



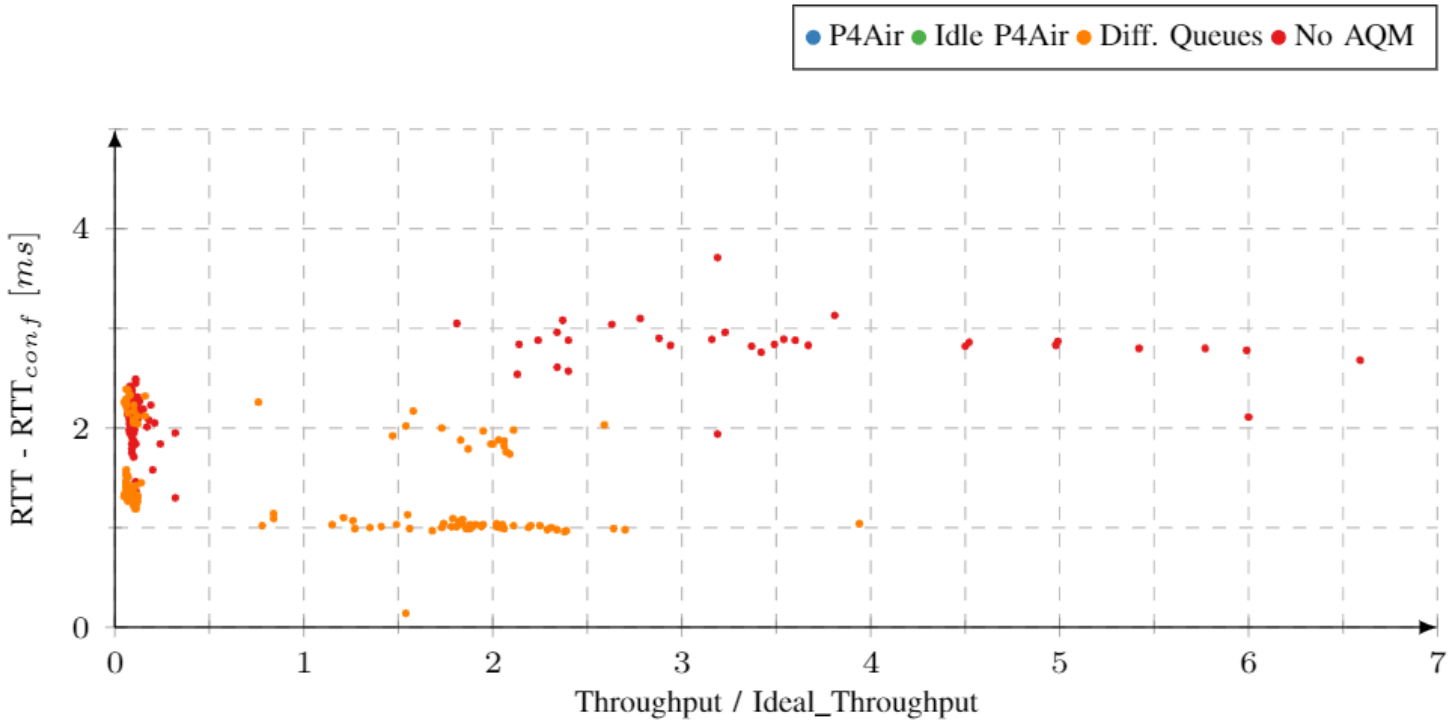
Evaluation – inter- and intra-fairness



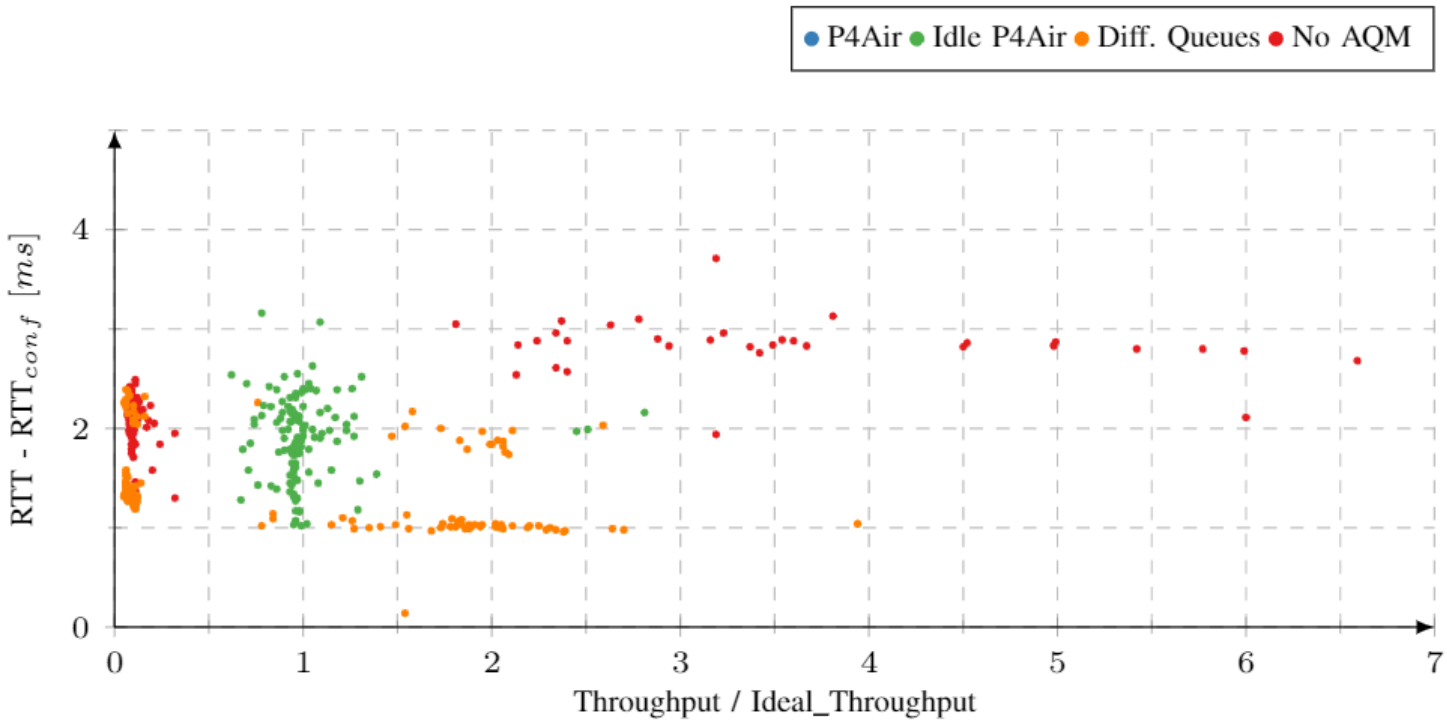
Evaluation – inter- and intra-fairness



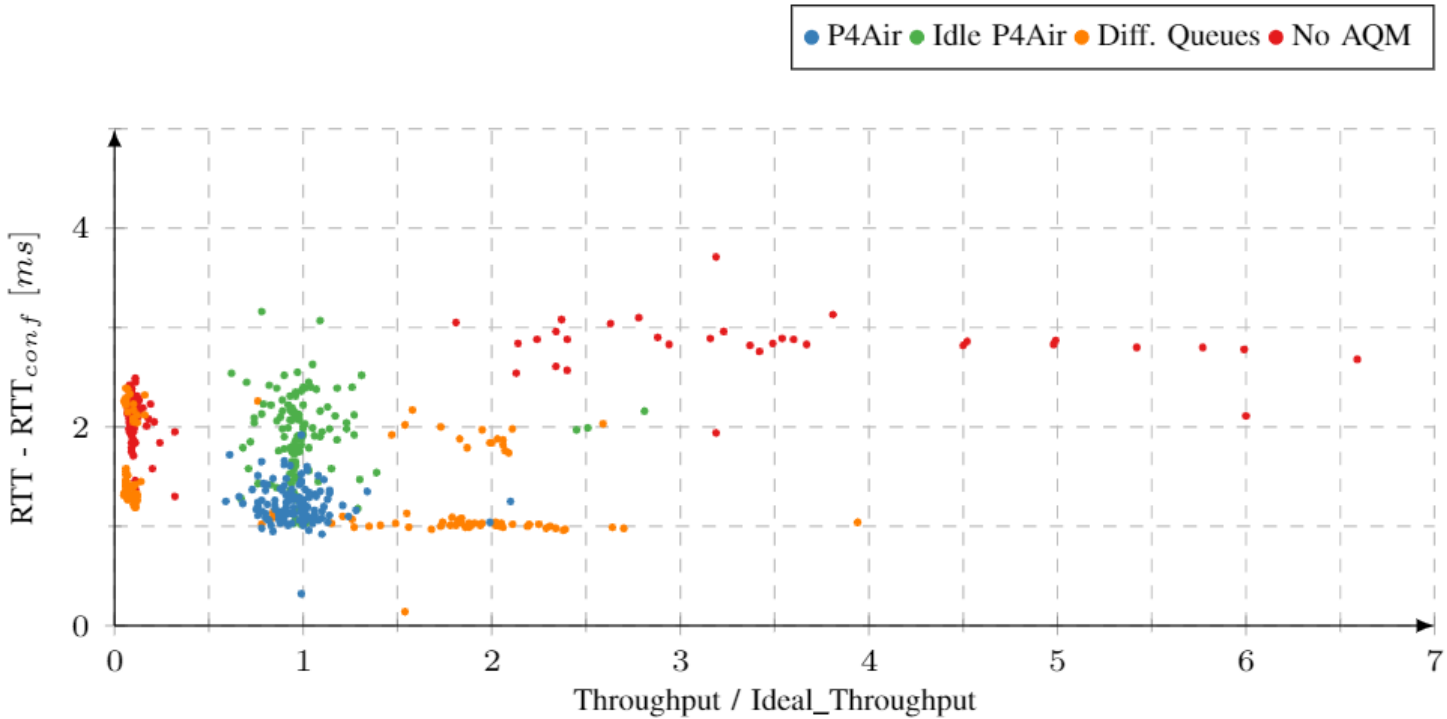
Evaluation – inter- and intra-fairness



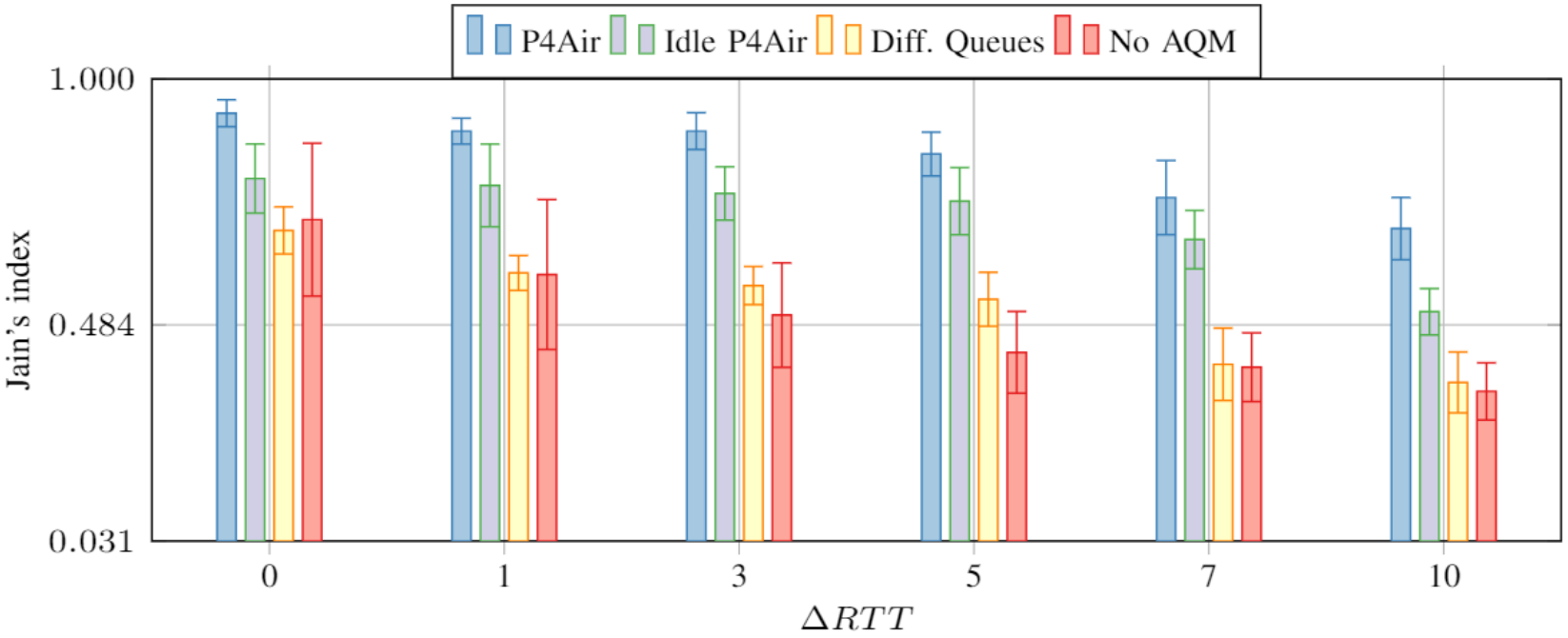
Evaluation – inter- and intra-fairness



Evaluation – inter- and intra-fairness



Evaluation – RTT-fairness



Conclusion

Conclusion

- Distributing flows to queues based on their congestion control group can significantly improves fairness in network resource utilization