ETH zürich

Network Security Group
Department of Computer Science

DNS Congestion Control in Adversarial Settings

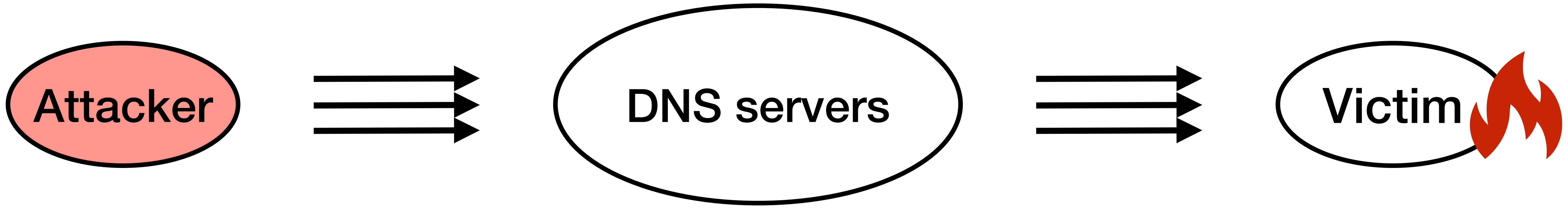
Huayi Duan, Jihye Kim, Marc Wyss, and Adrian Perrig

November 6, 2024
SOSP’24, Austin,TX, USA

Fast-moving DNS security landscape

DNS as tool for DoS

- Reflection



Fast-moving DNS security landscape

DNS as tool for DoS
- Reflection

DNS as target for DoS
- Pseudo-Random SubDomain

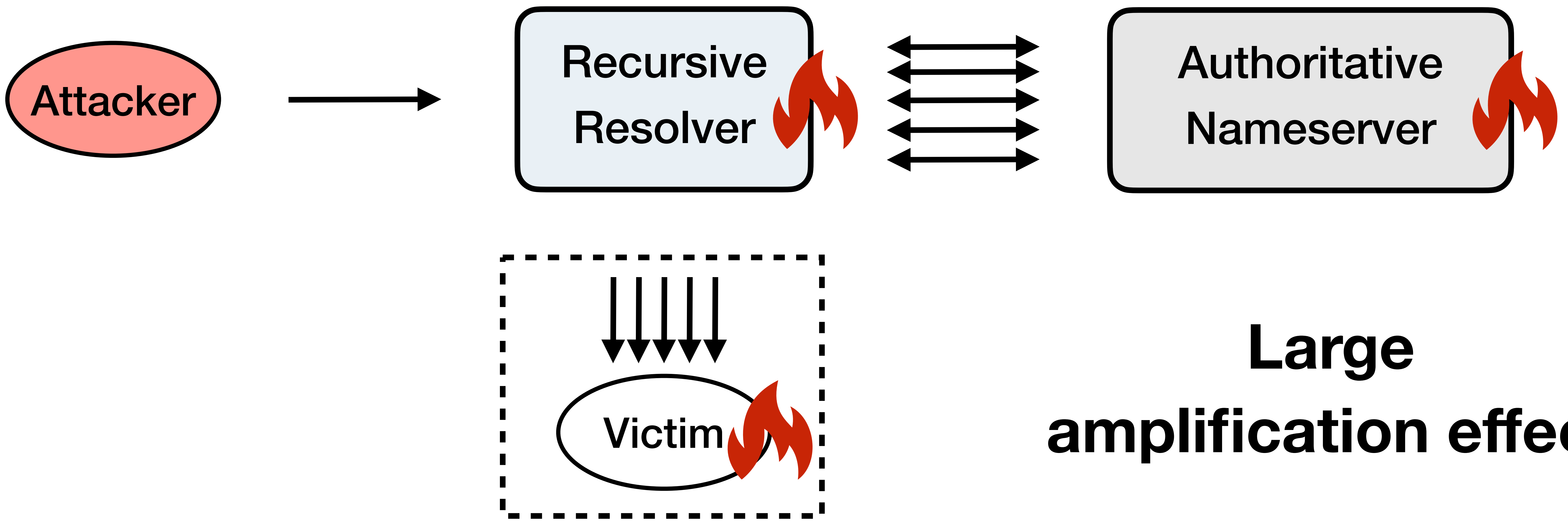
Fast-moving DNS security landscape

DNS as tool for DoS

- Reflection
- **DNSBomb, SP'24**
- **TsuKing, CCS'23**
- **CAMP, SEC'24**
- …

DNS as target for DoS

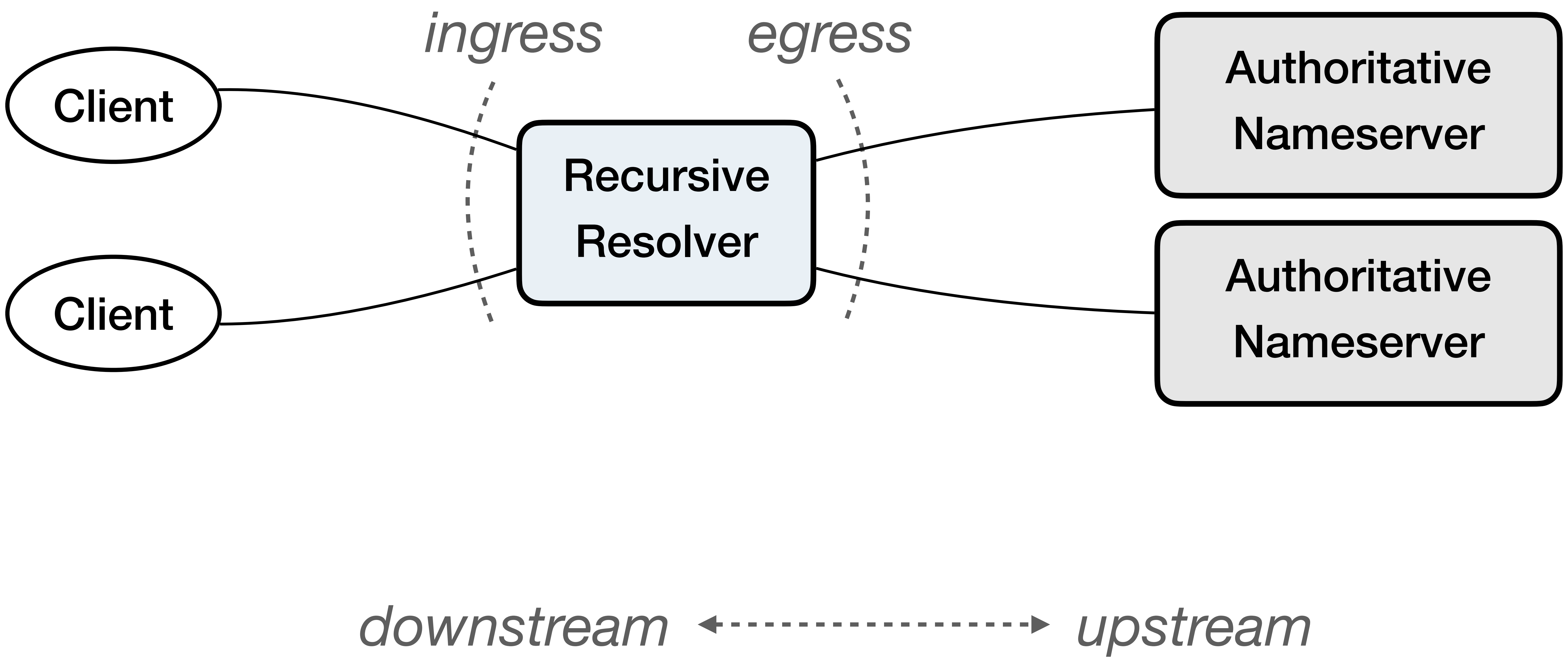
- Pseudo-Random SubDomain
- **NXNSAttack, SEC'20**
- **TsuNAME, IMC'21**
- **CAMP, SEC'24**
- …



Large amplification effect

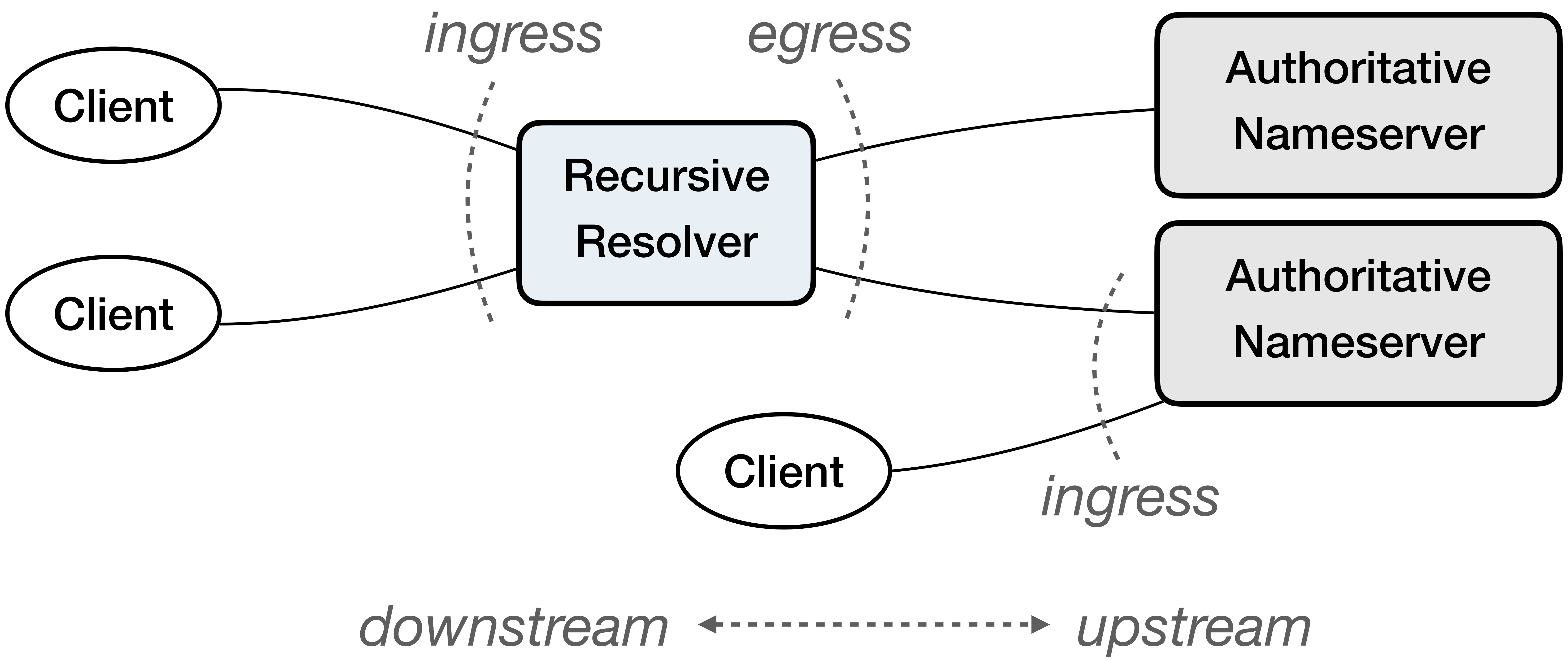
Rate limiting as a universal defense

Upper bound individual entity's impact



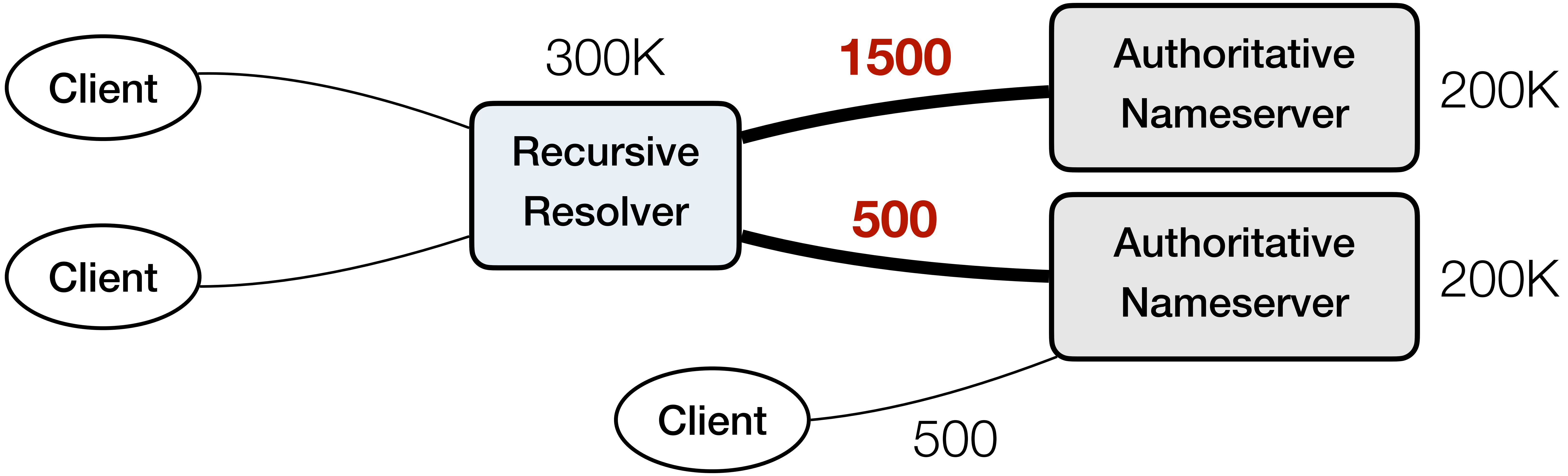
Rate limiting as a universal defense

Upper bound individual entity's impact



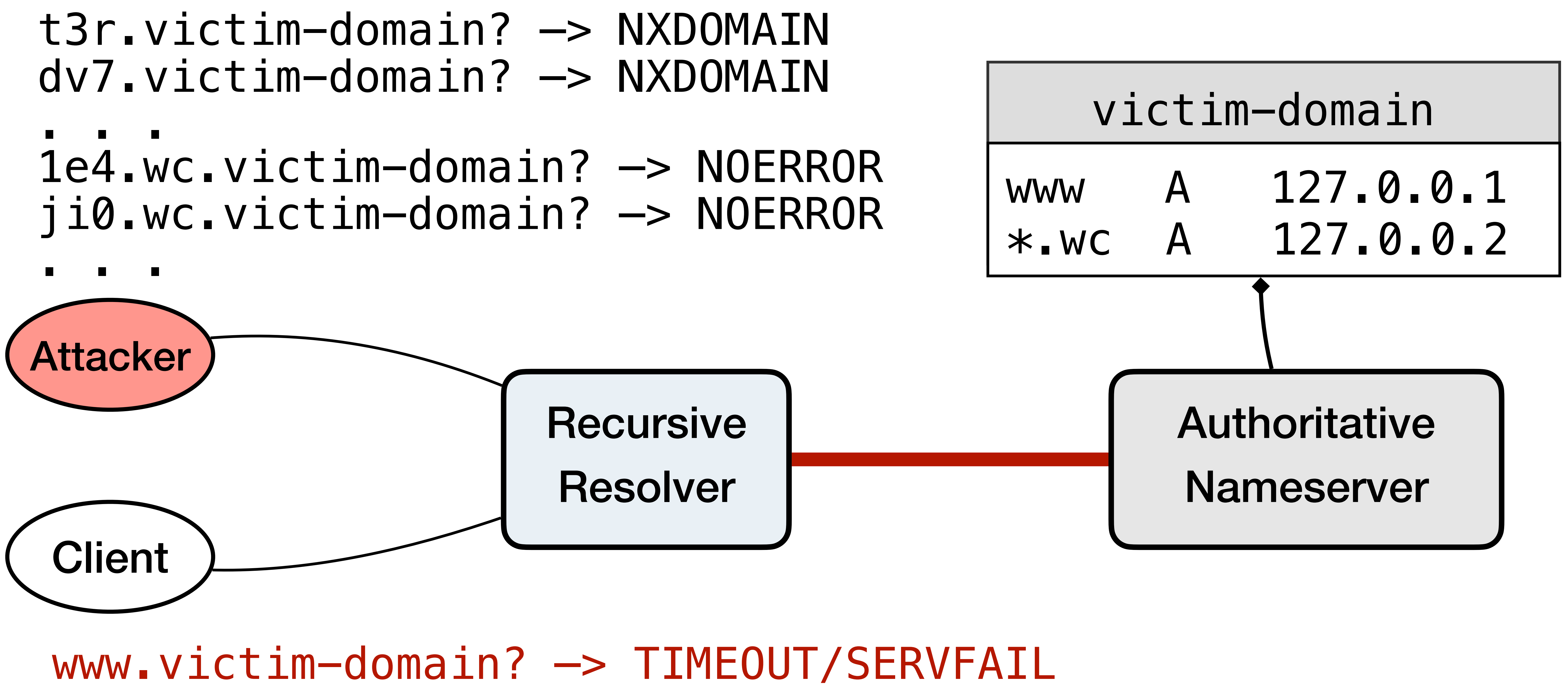
Rate limiting as a universal defense that expands DoS attack surface!

Result in ***logical inter-server channel*** with ***limited capacity***



Adversarial congestion on inter-server channels

Can disrupt access to ***victim domain*** via shared resolver

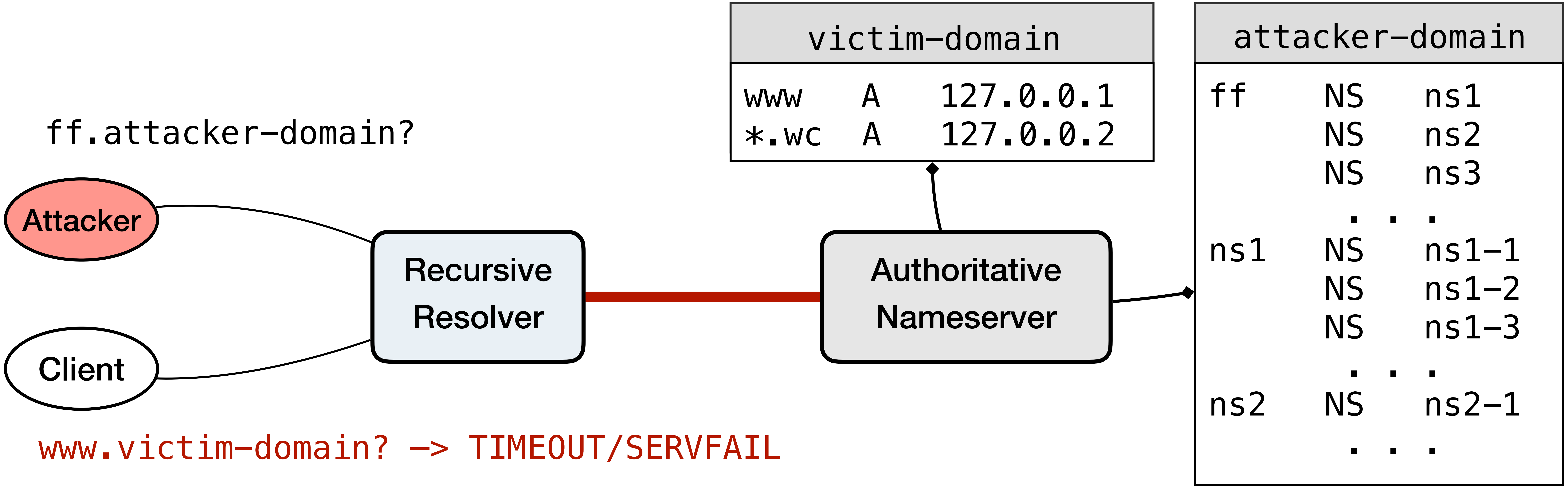


Adversarial congestion on inter-server channels

Can disrupt access to ***victim domain*** via shared resolver

Can leverage ***amplification,*** *esp. when* the attacker can access victim nameserver

89% of top-100K domains hosted by 3rd-party DNS [Kashaf et al., IMC'20]



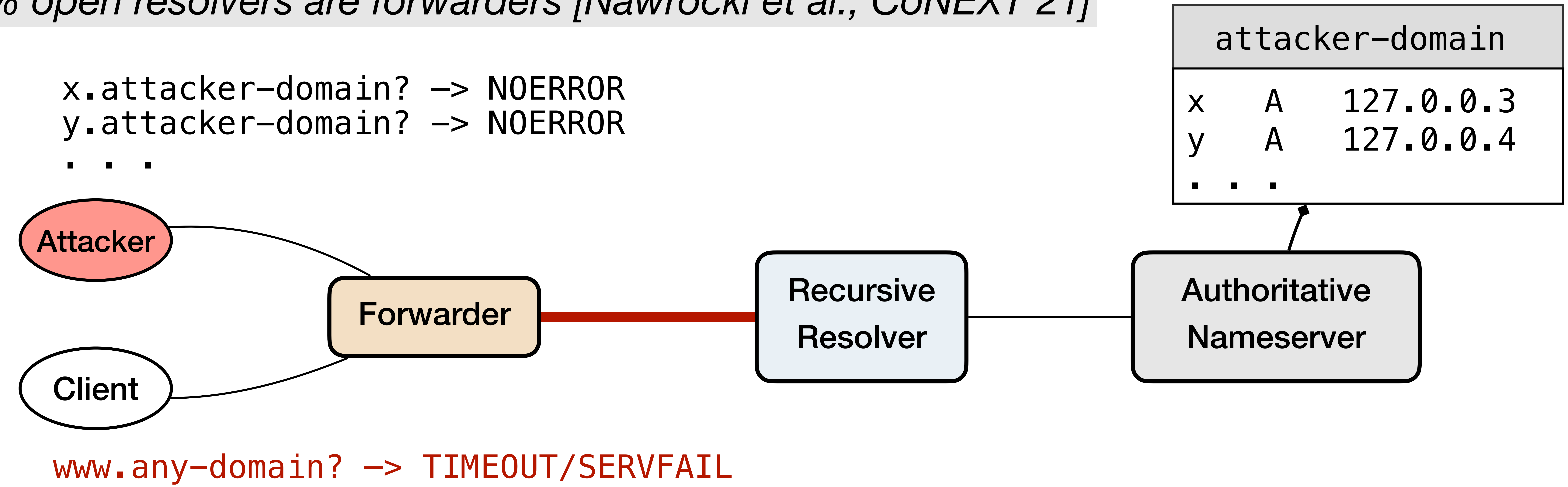
Adversarial congestion on inter-server channels

Can disrupt access to ***victim domain*** via shared resolver

Can leverage ***amplification,*** *esp. when* the attacker can access victim nameserver

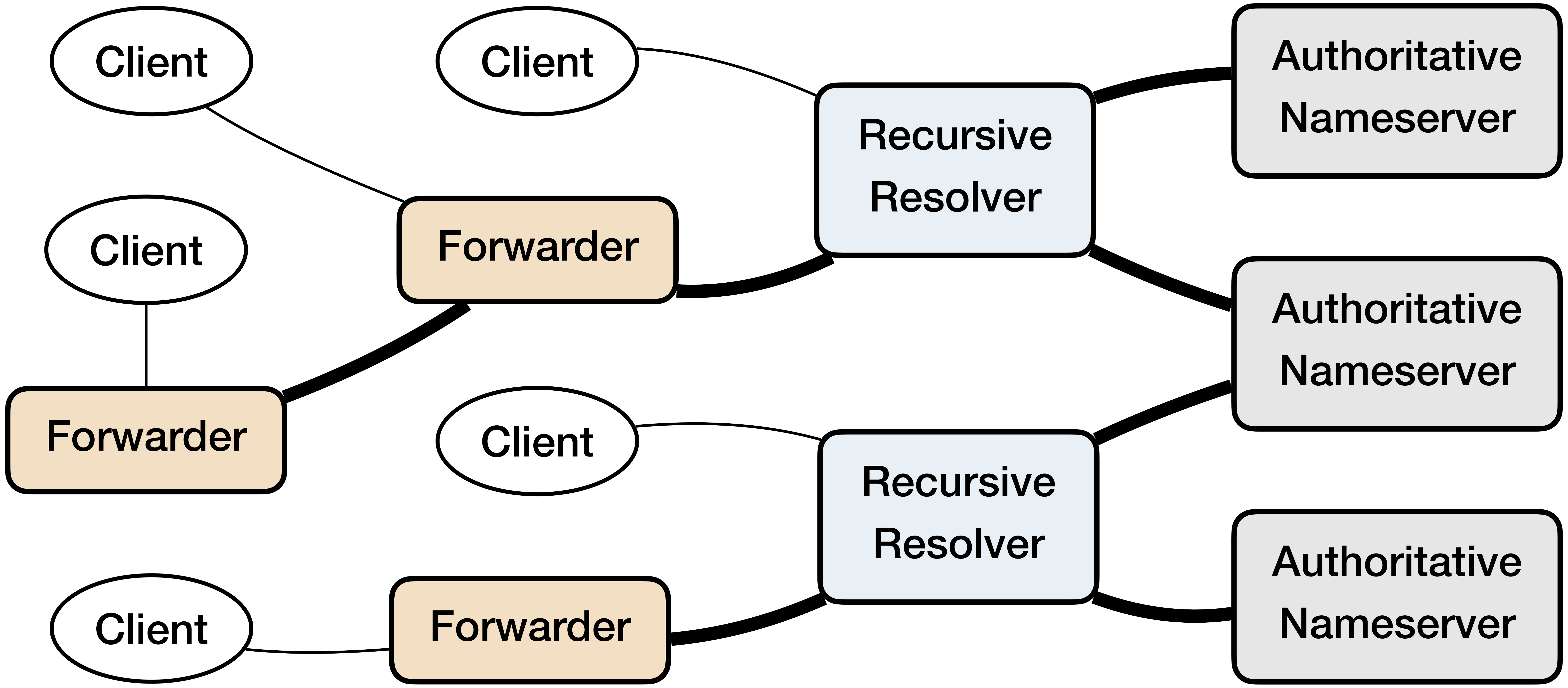
Can disrupt access to ***all domains*** via shared forwarder

>90% open resolvers are forwarders [Nawrocki et al., CoNEXT'21]



Adversarial congestion on inter-server channels

Is an ***inherent vulnerability in DNS architecture!***

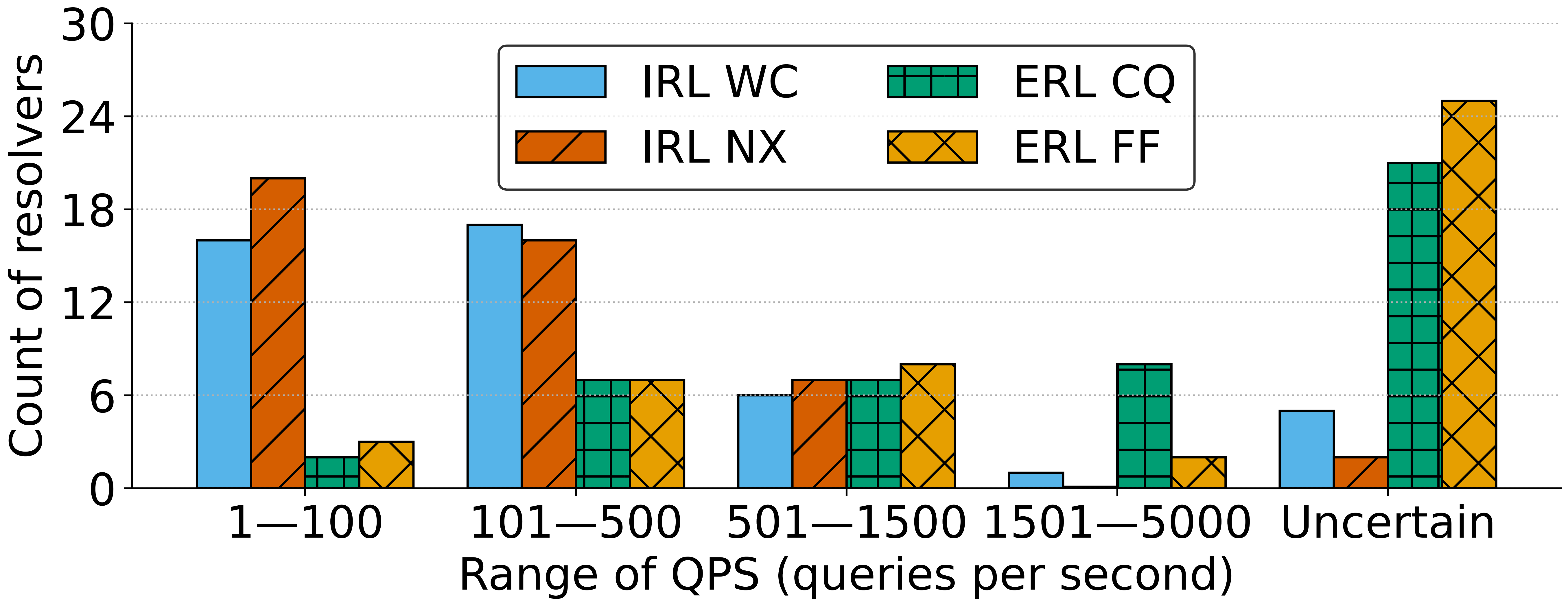


Real-world risk of adversarial congestion is high

Ingress/egress rate limiting (RL) measurement on 45 open resolvers

40 resolvers with IRL <= 1500 (default by 8.8.8.8)

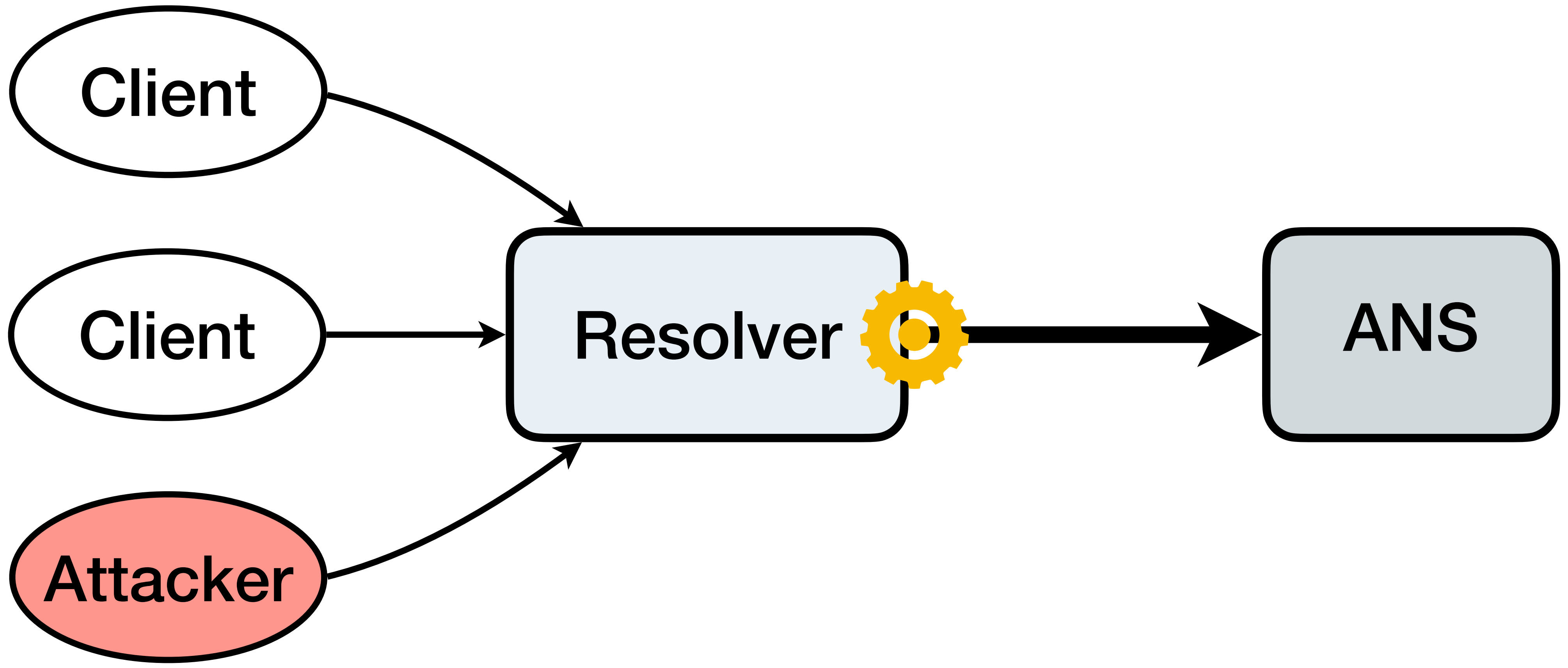
Generally higher ERL, but more uncertain cases (best-effort estimates)



100Ks of authoritative nameservers with IRL <= 500 [Deccio et al., 2019]

Design intuitions for mitigation

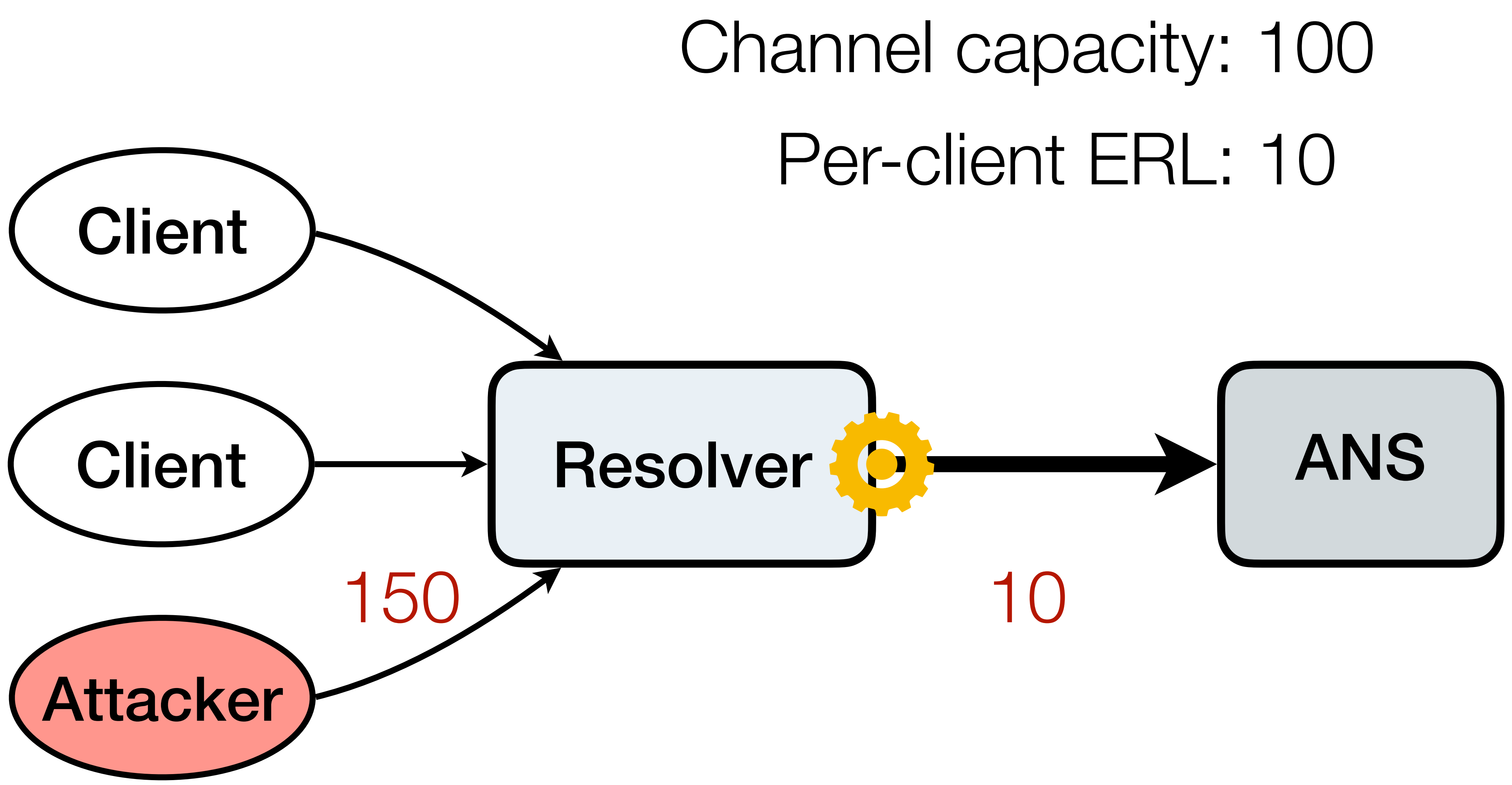
Congestion control at downstream



Design intuitions for mitigation

Congestion control at downstream

Per-client egress query RL?

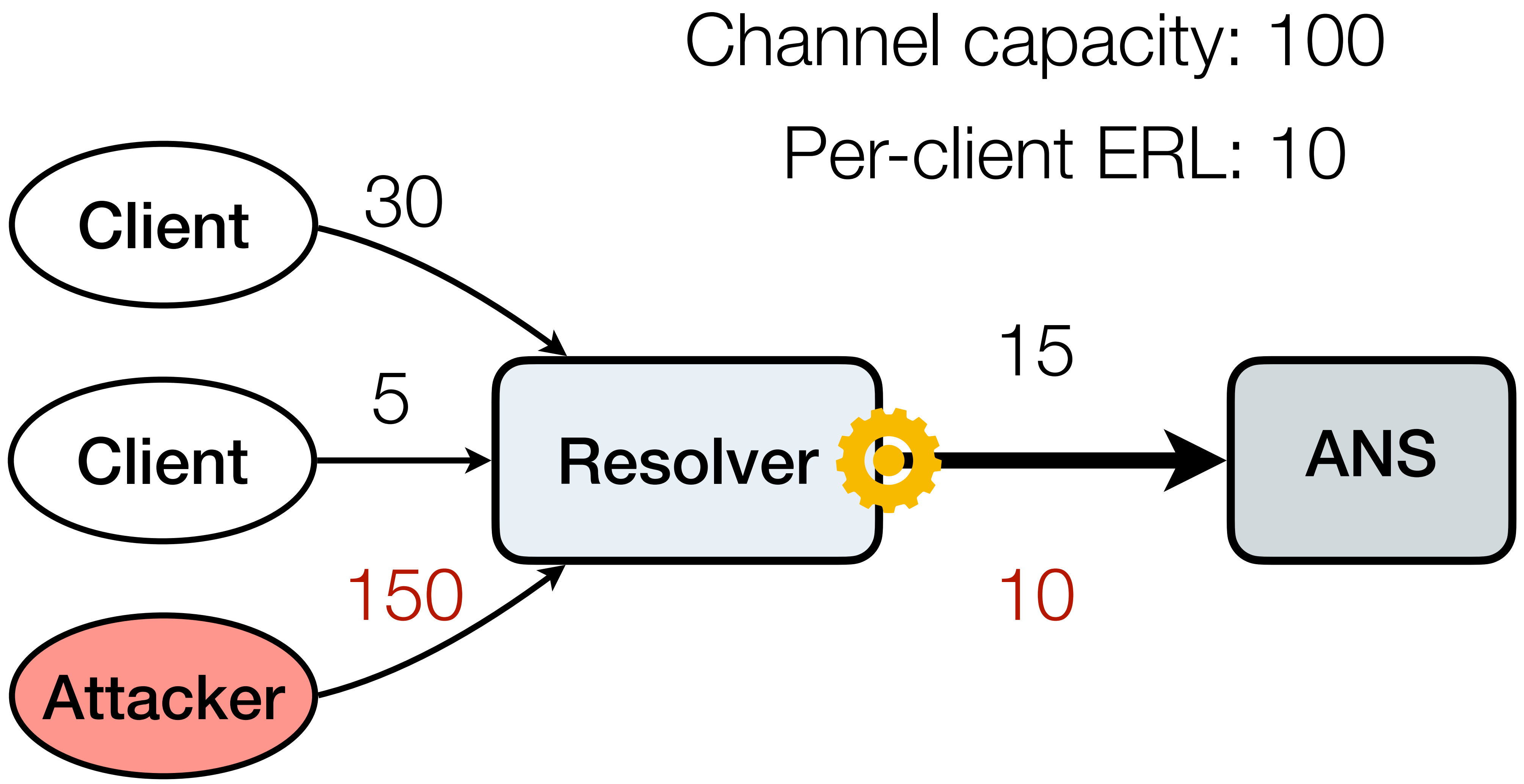


Design intuitions for mitigation

Congestion control at downstream

Per-client egress query RL?

- **Not work-conserving**

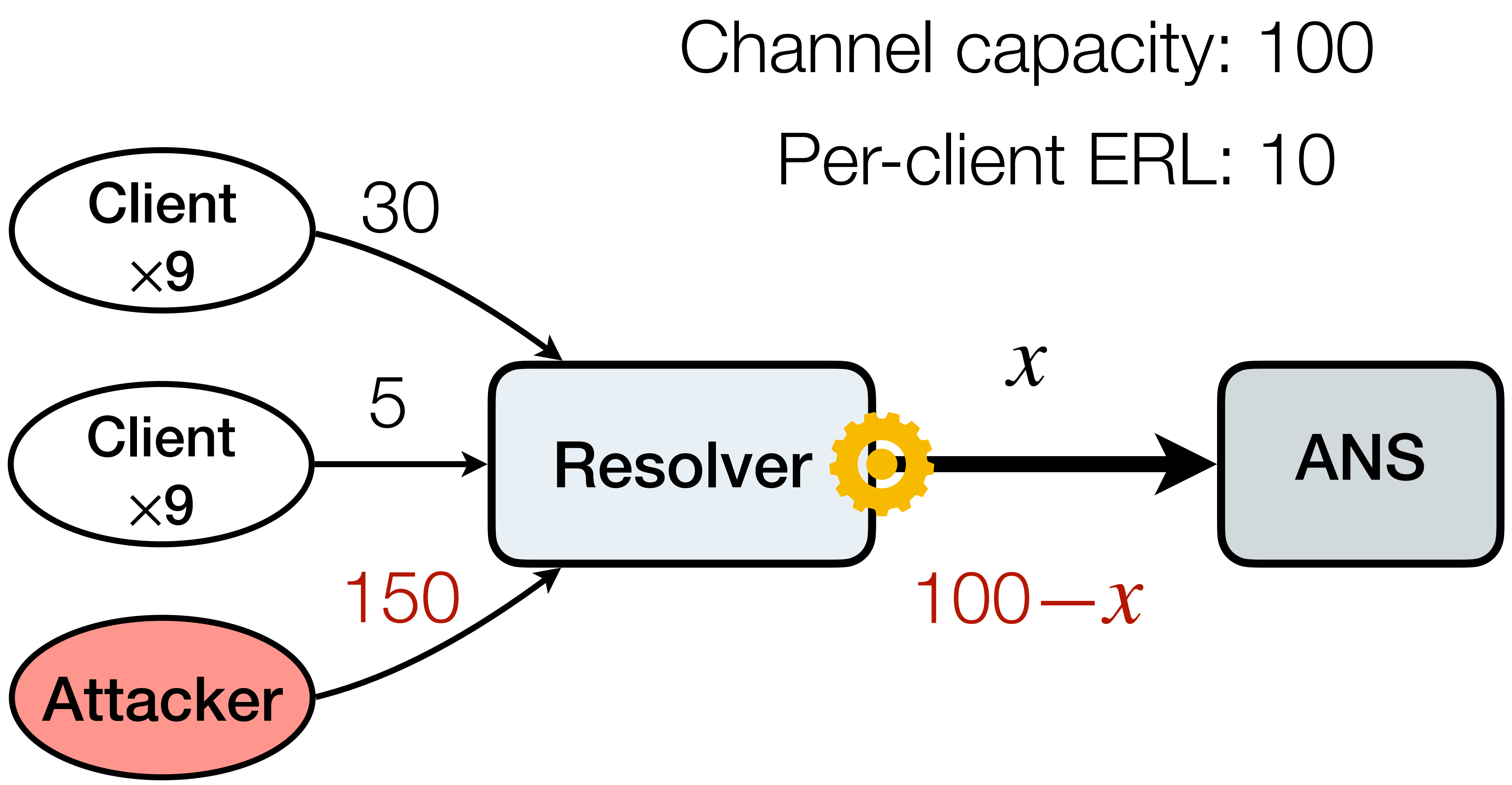


Design intuitions for mitigation

Congestion control at downstream

Per-client egress query RL?

- Not work-conserving
- **No guaranteed access**



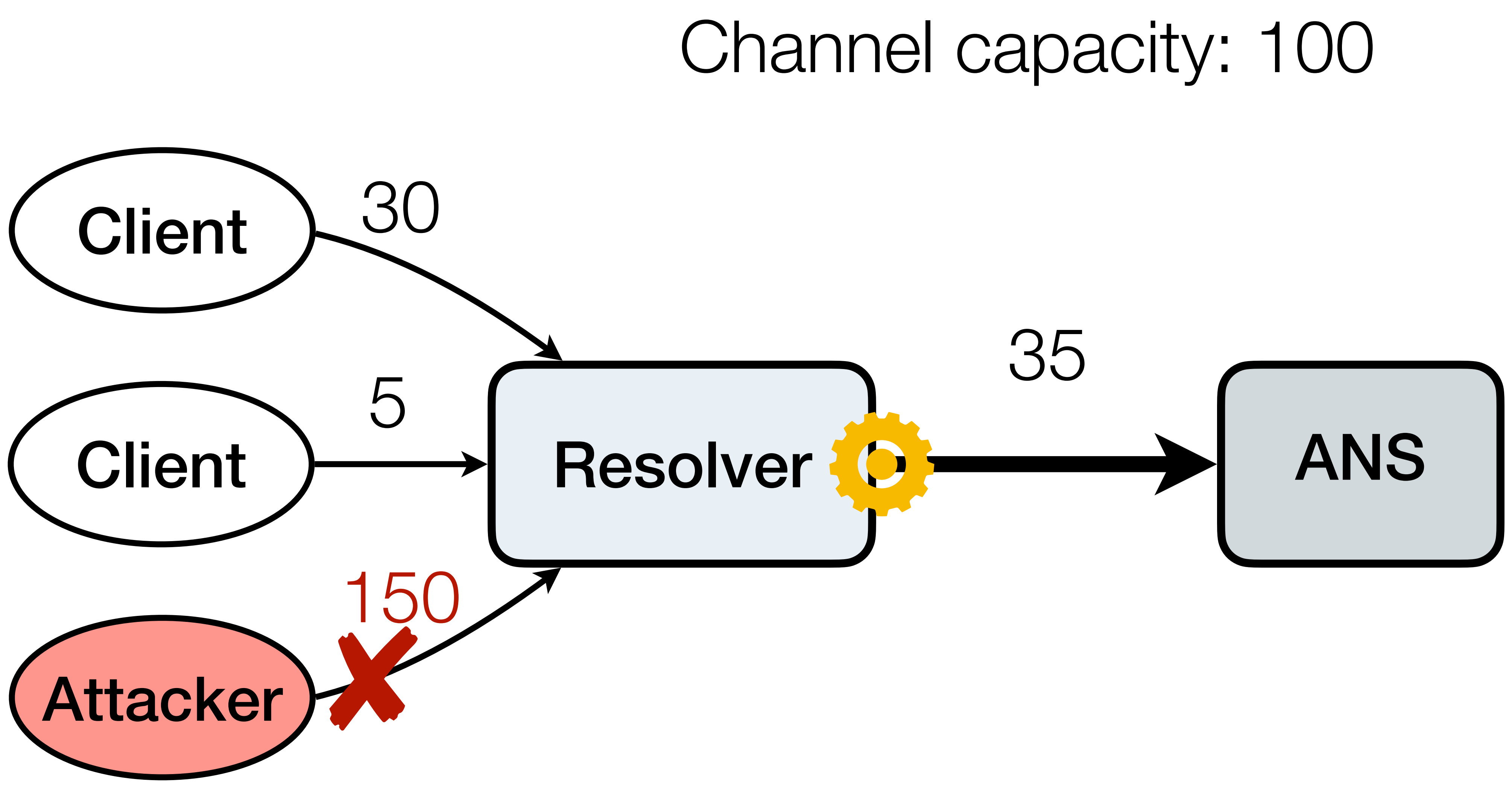
Design intuitions for mitigation

Congestion control at downstream

Per-client egress query RL?

- Not work-conserving
- No guaranteed access

Detect and police suspicious sender?



Design intuitions for mitigation

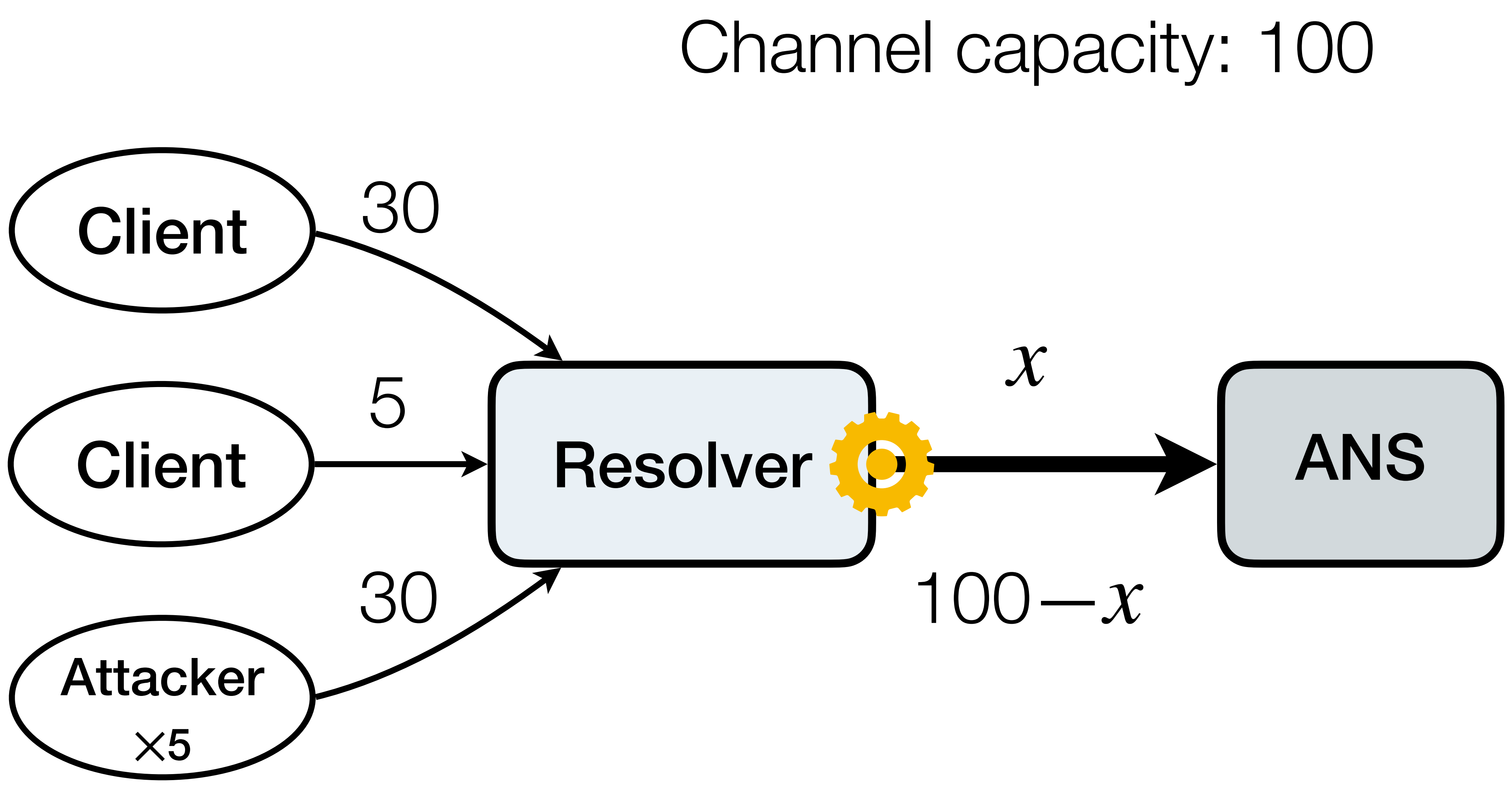
Congestion control at downstream

Per-client egress query RL?

- Not work-conserving
- No guaranteed access

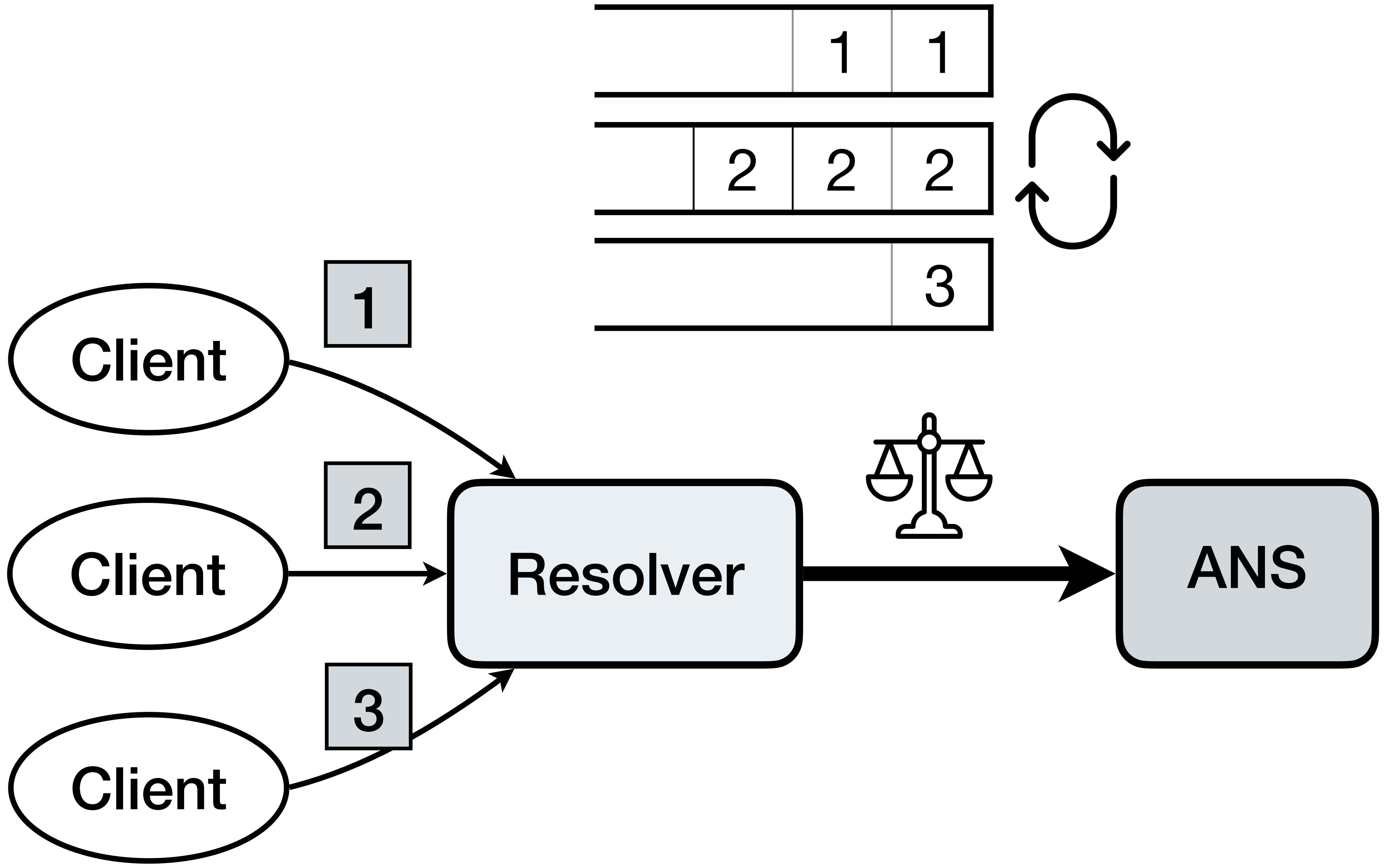
Detect and police suspicious sender?

- **Attacker can mimic benign clients**



Fair queuing (FQ) as a principled solution

Worst-case guarantees of fair access

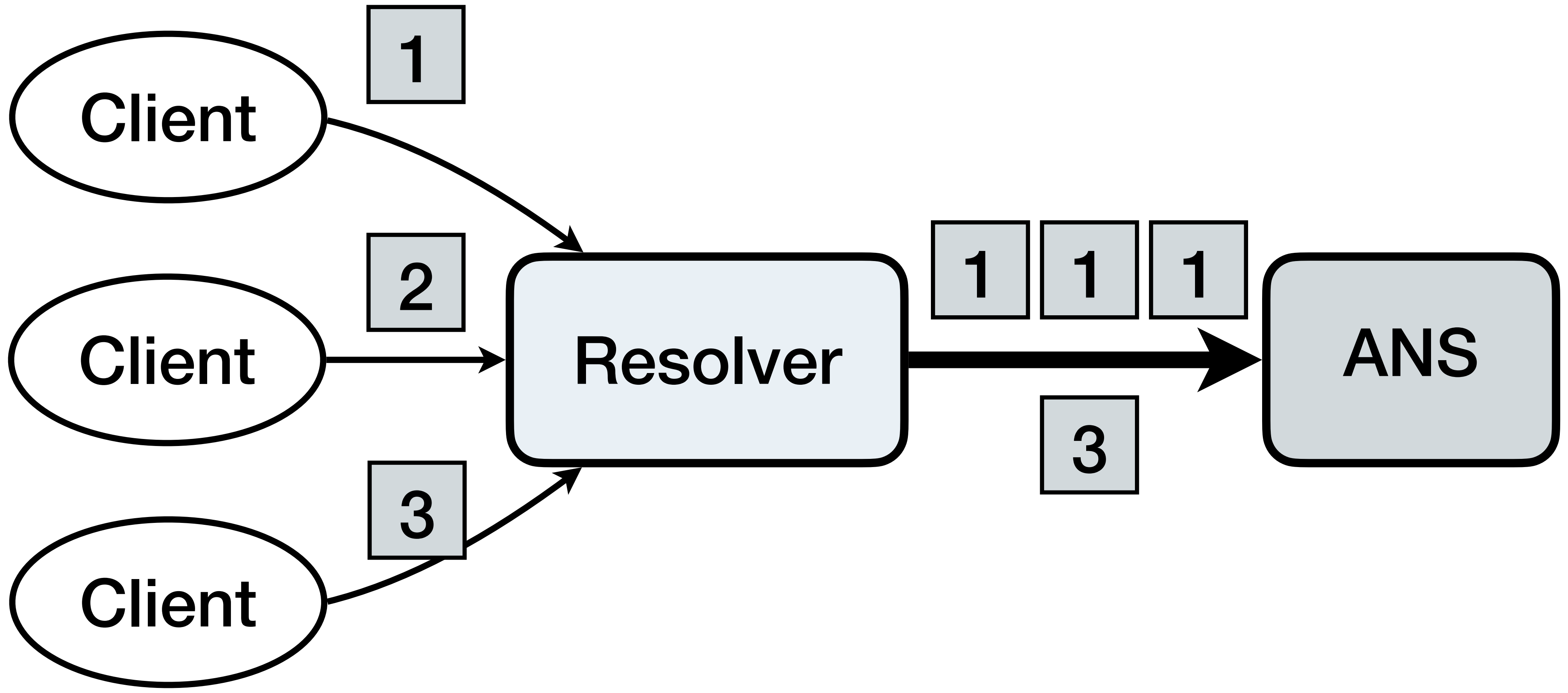


Fair queuing (FQ) as a principled solution

Worst-case guarantees of fair access

Why unique in DNS?

- **No 1:1 relation between in & out msg**

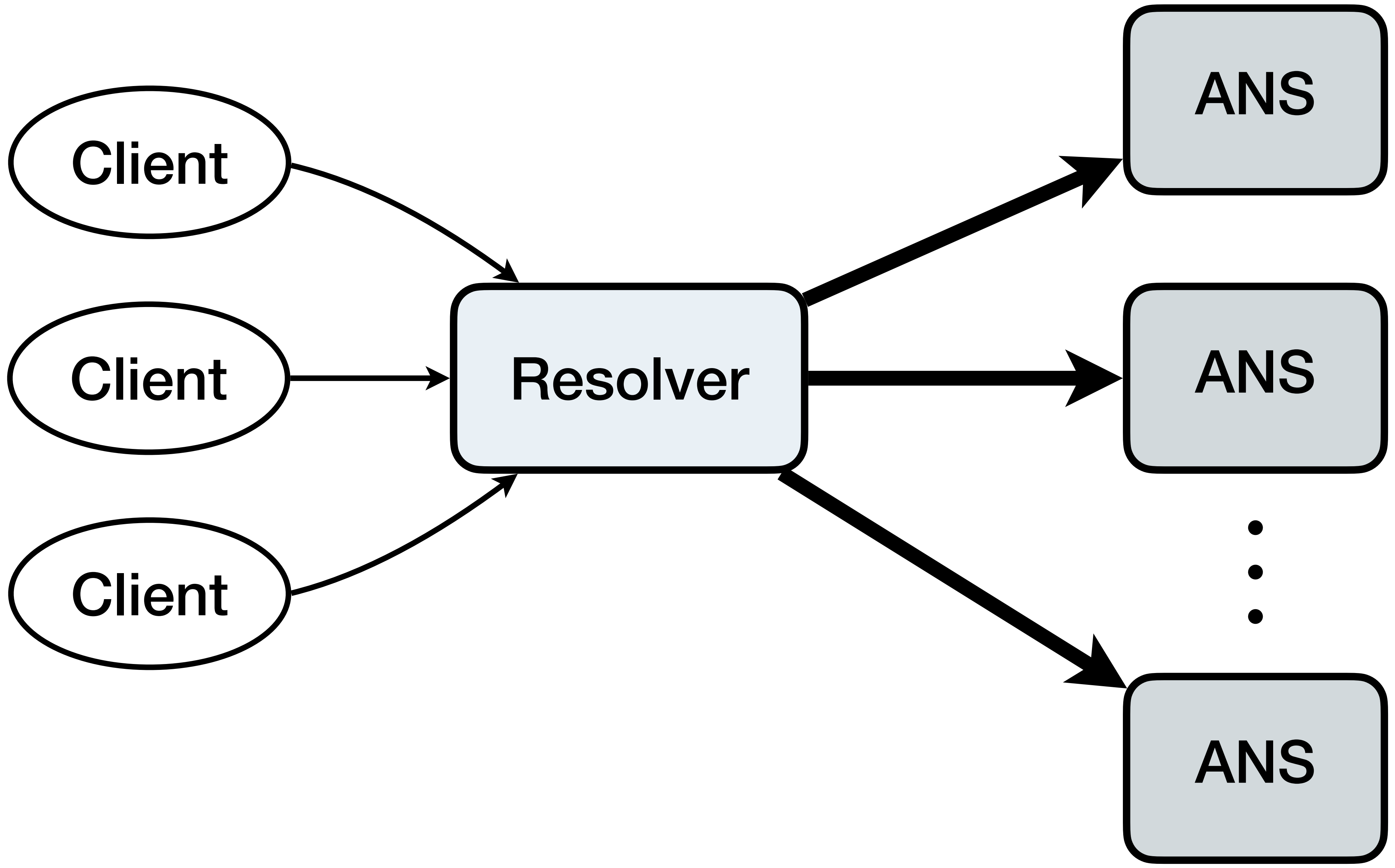


Fair queuing (FQ) as a principled solution

Worst-case guarantees of fair access

Why unique in DNS?

- No 1:1 relation between in & out msg
- **Many distinct output channels**

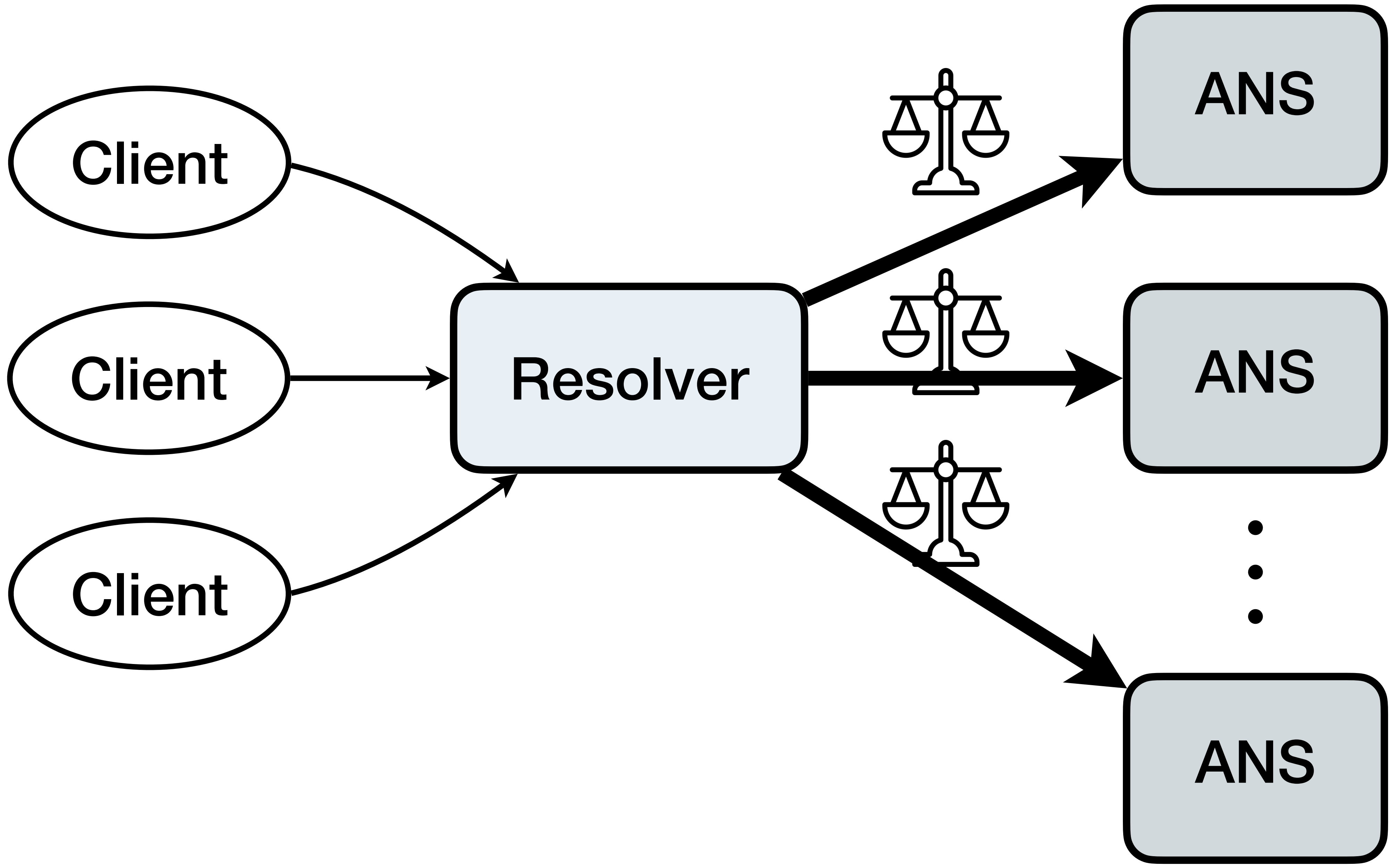


Fair queuing (FQ) as a principled solution

Worst-case guarantees of fair access

Why unique in DNS?

- No 1:1 relation between in & out msg
- Many distinct output channels
- **Fairness for individual channels**

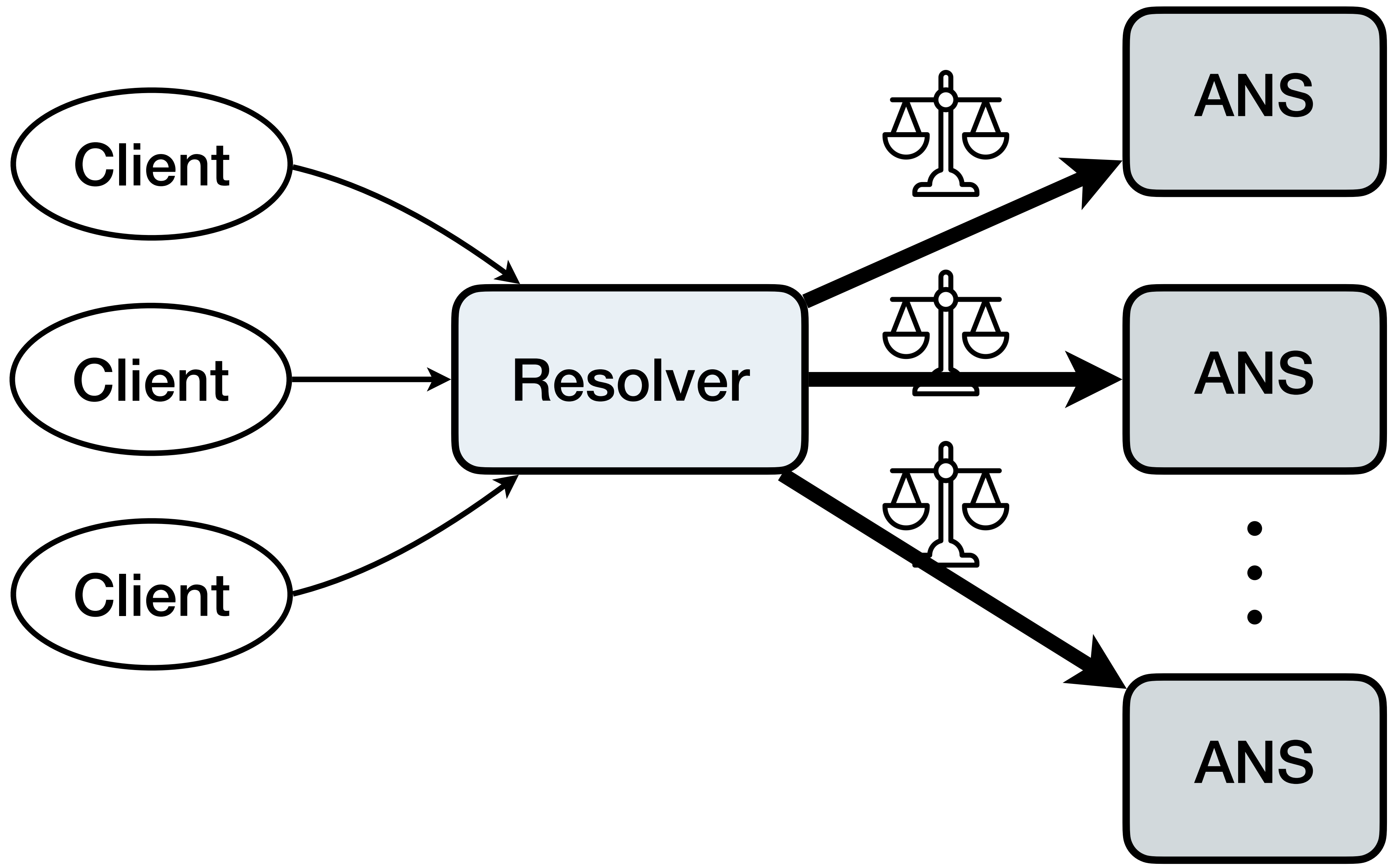


Fair queuing (FQ) as a principled solution

Worst-case guarantees of fair access

Why unique in DNS?

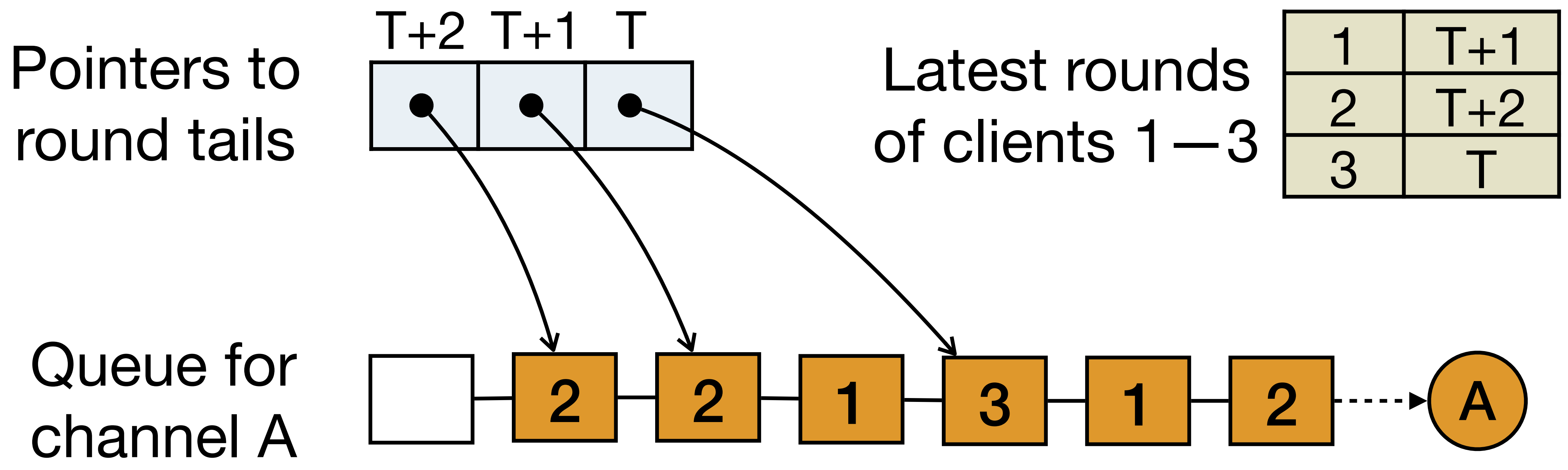
- No 1:1 relation between in & out msg
- Many distinct output channels
- Fairness for individual channels



Different from multi-server/queue/interface/resource FQ

MOPI-FQ (Multi-Output Pseudo-Isolated Fair Queuing)

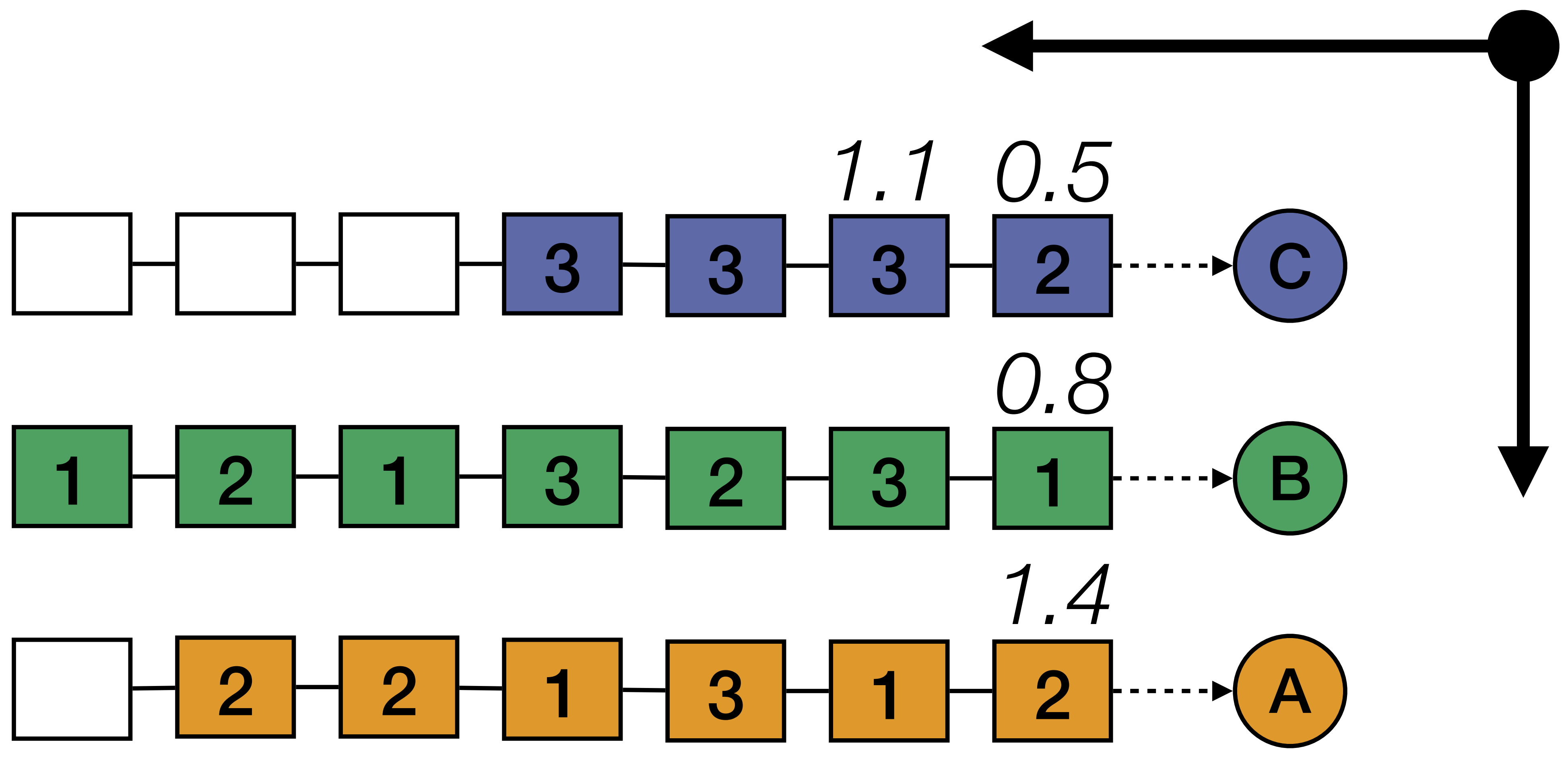
Simplified bit-by-bit round-robin per output channel —> max-min fairness



MOPI-FQ (Multi-Output Pseudo-Isolated Fair Queuing)

Simplified bit-by-bit round-robin per output channel —> max-min fairness

Order-preserving scheduling across channels —> confine queuing delay



Scheduling in two directions based on query arrival time

MOPI-FQ (Multi-Output Pseudo-Isolated Fair Queuing)

Simplified bit-by-bit round-robin per output channel —> max-min fairness

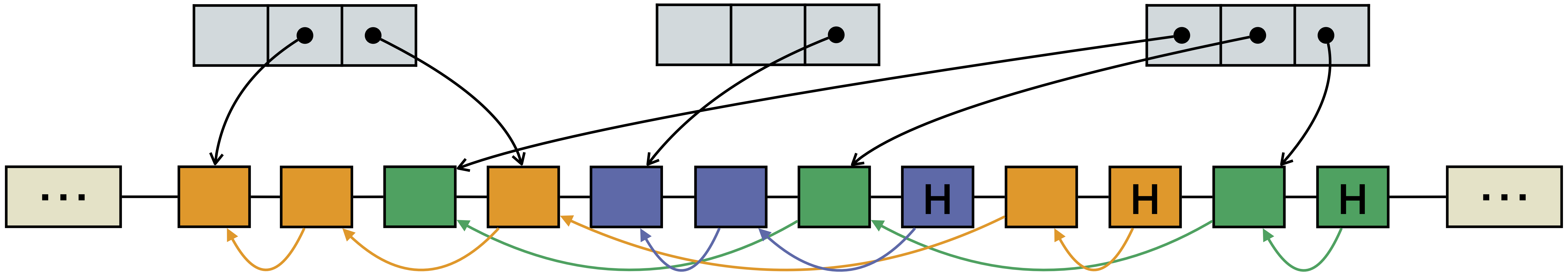
Order-preserving scheduling across channels —> confine queuing delay

Dynamic allocation of queues from shared pool —> minimise space overhead

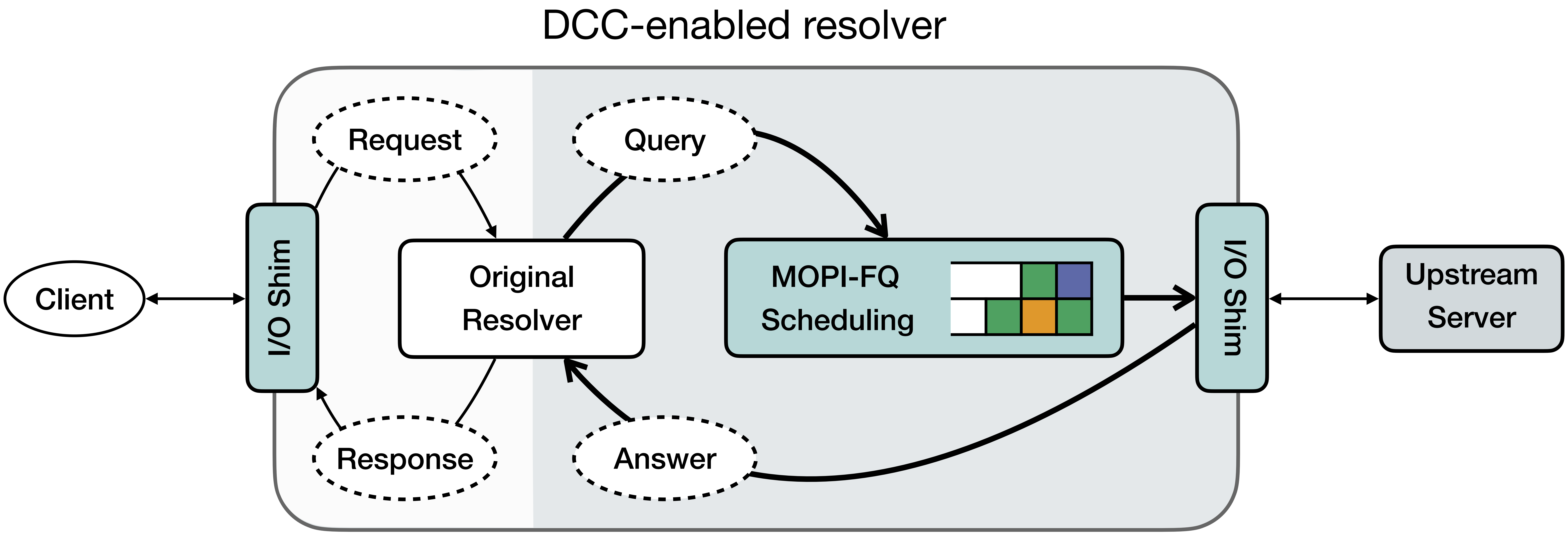
Space complexity: $\mathcal{O}(n+q)$ Time complexity: $\mathcal{O}(log(n))$

n: #output channels

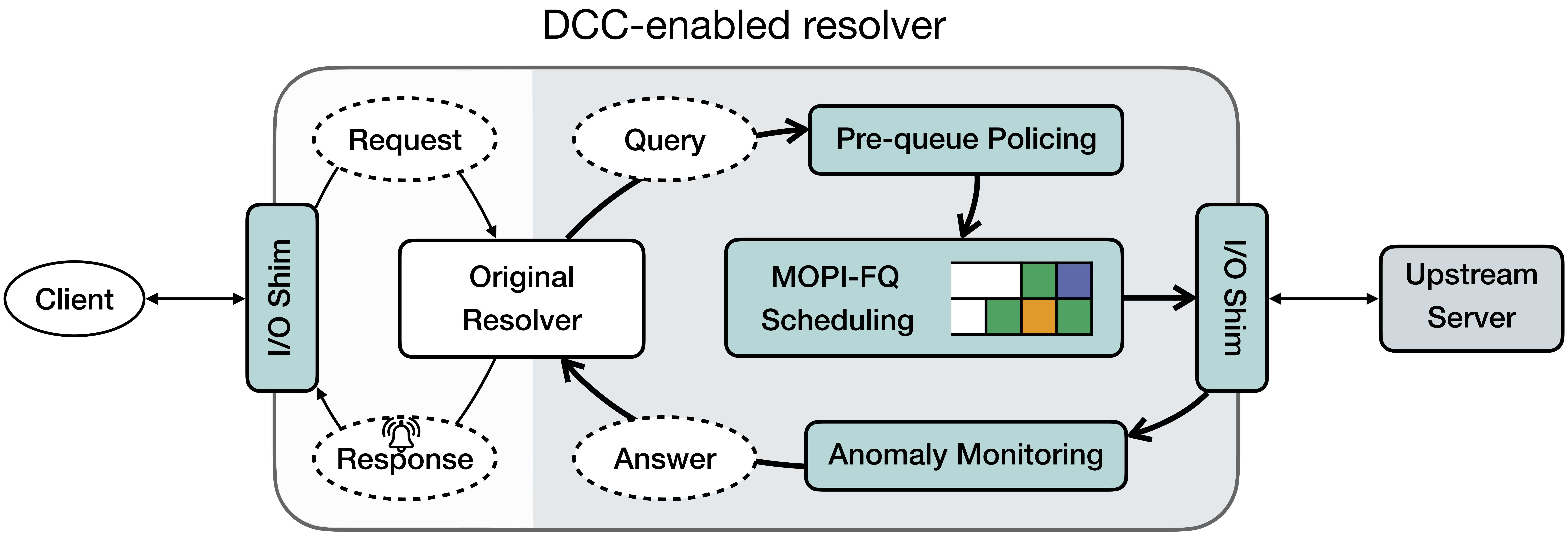
q: overall queue depth



DCC (DNS Congestion Control) overview



DCC (DNS Congestion Control) overview

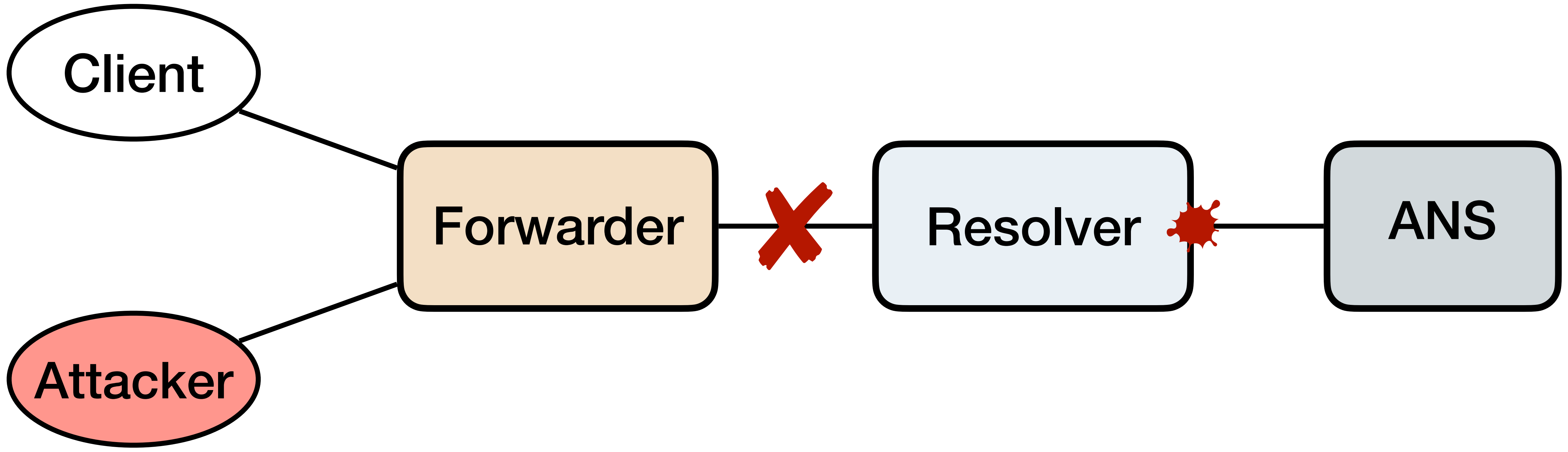


Signals generated on special events and encoded as EDNS option in response

DCC signalling

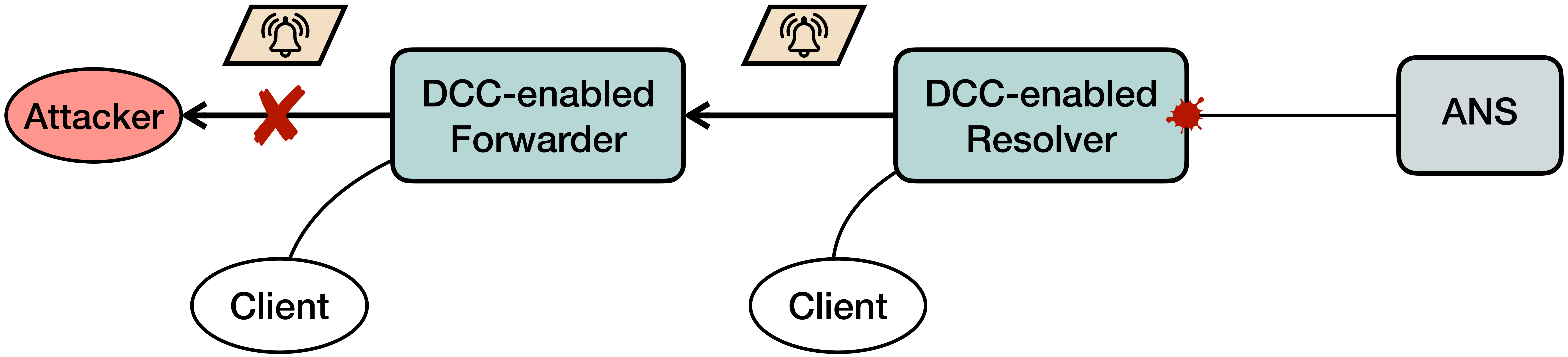
In general, blindly policing a client can cause **collateral damage**

—> **another architectural DoS vector**

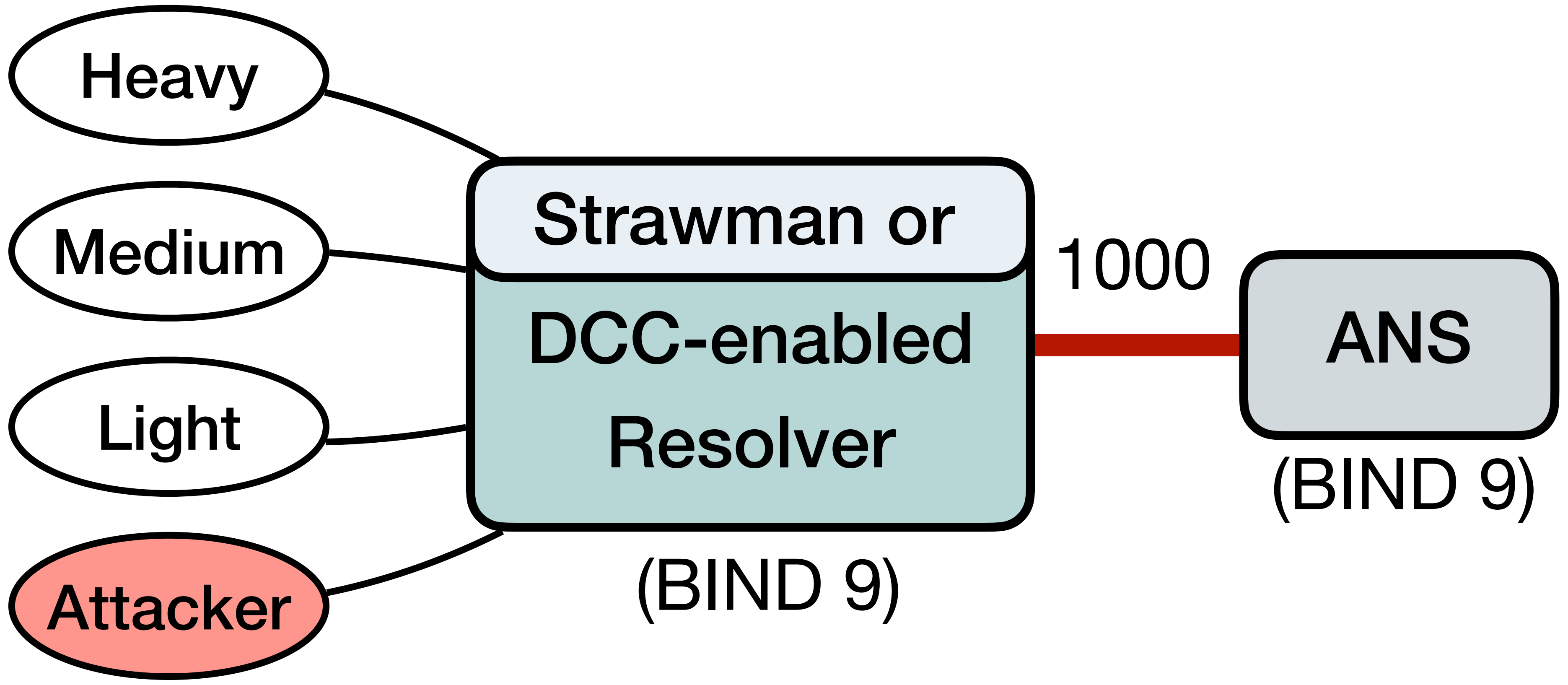


DCC signalling

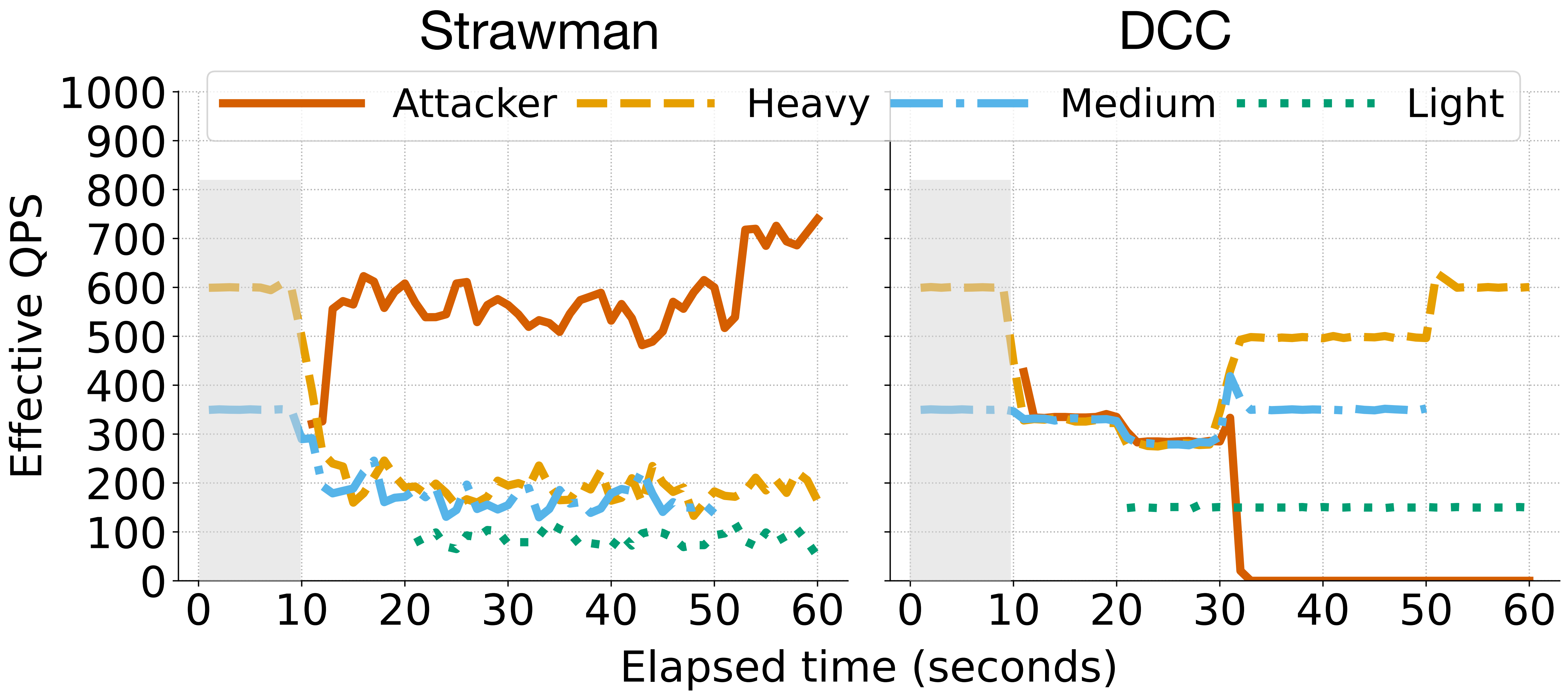
Signals propagated backwards to enable **fine-grained control**



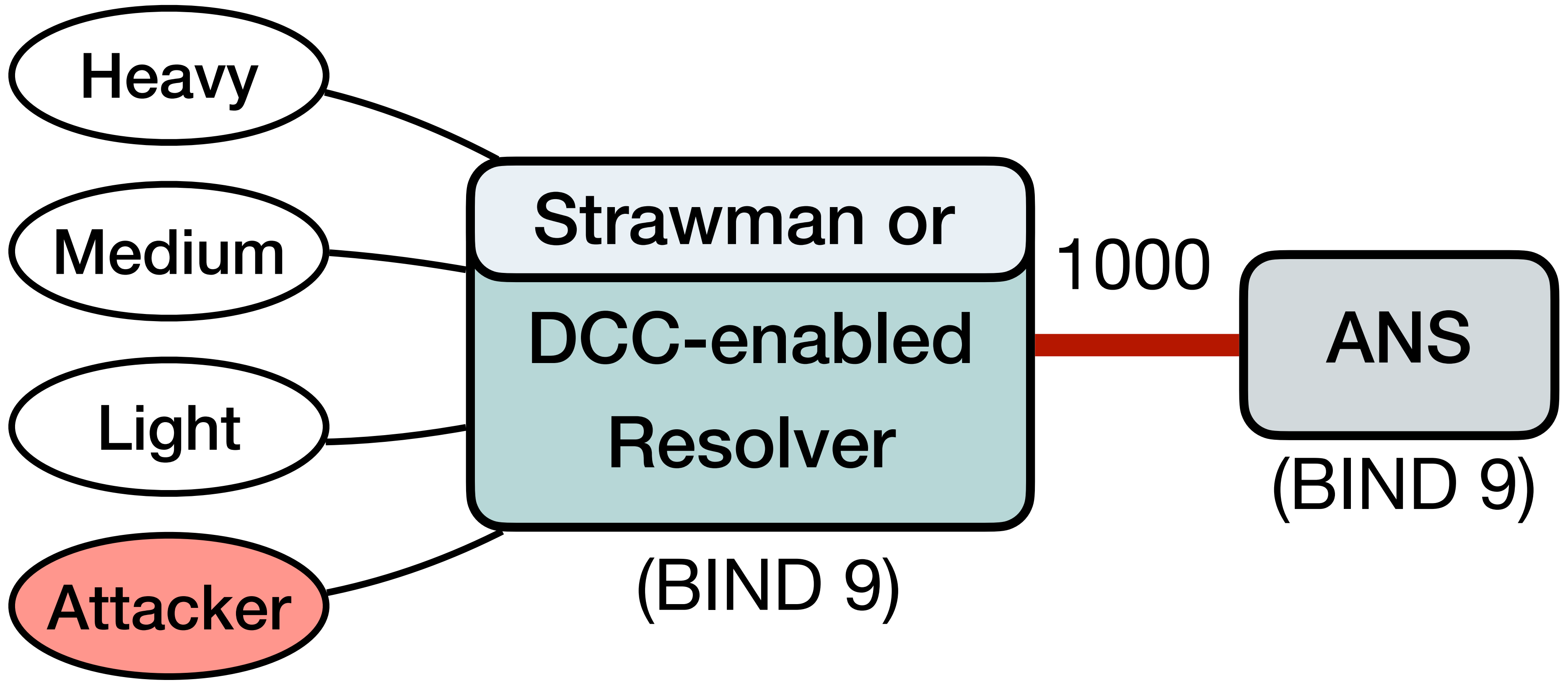
Evaluation of DCC prototype



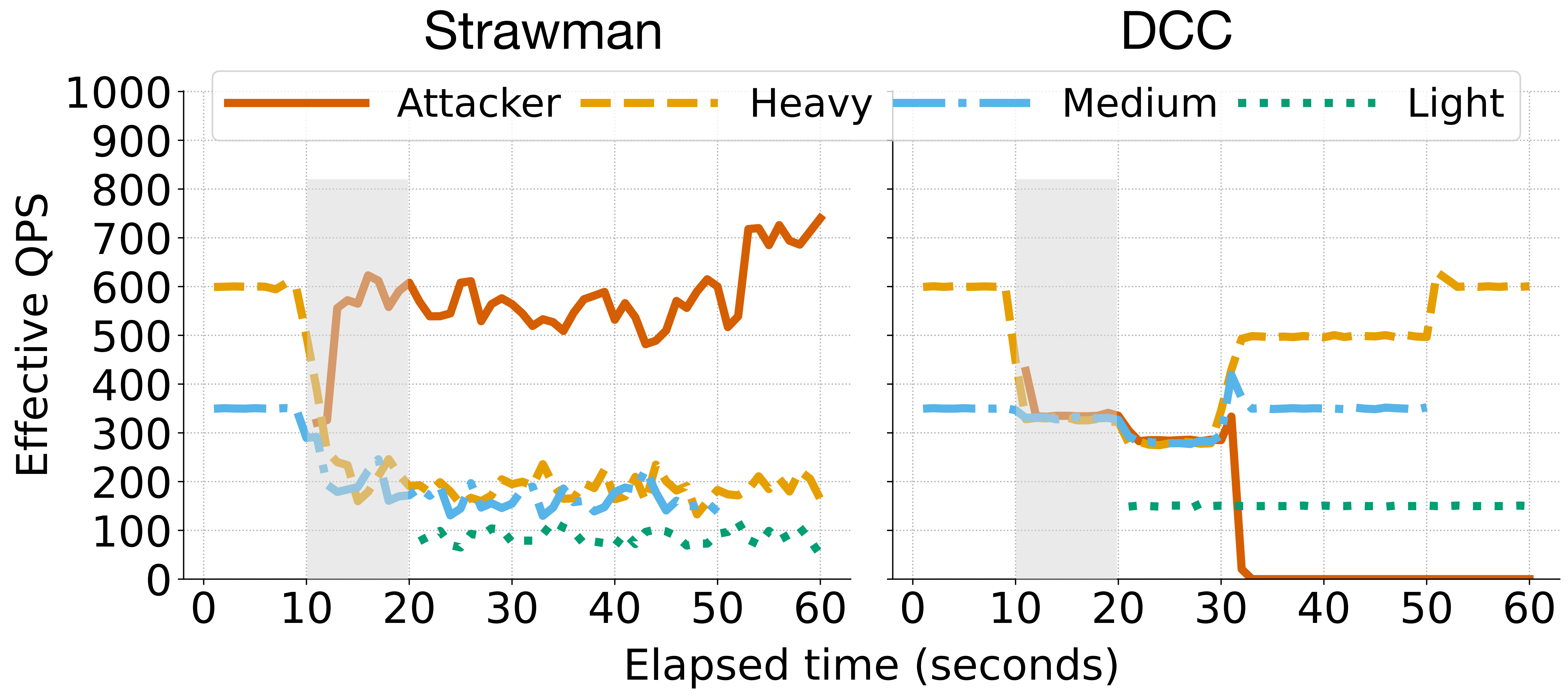
Client	Start	End	Req Rate	Query Pattern
Heavy	0	60	600	WC
Medium	0	50	350	WC
Light	20	60	150	WC
Attacker	10	60	50	FF



Evaluation of DCC prototype

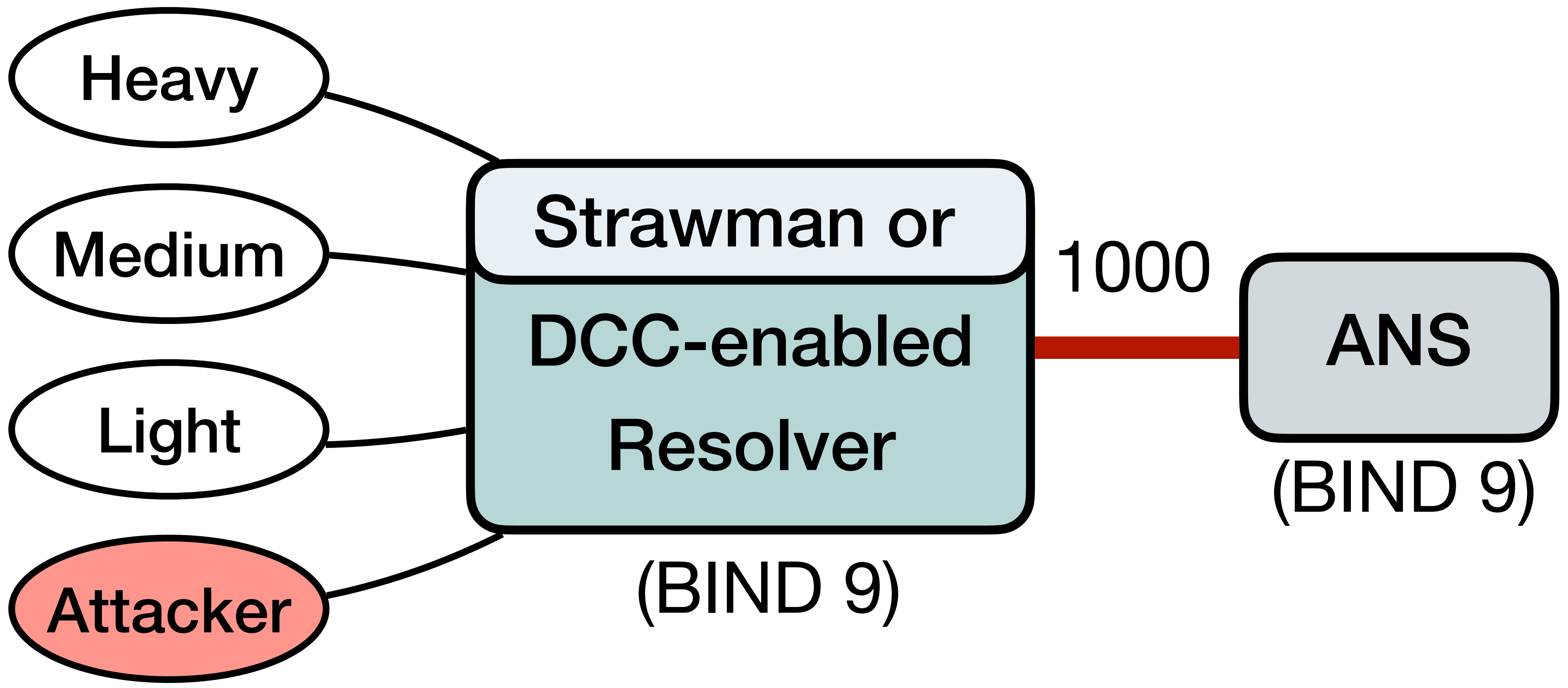


Client	Start	End	Req Rate	Query Pattern
Heavy	0	60	600	WC
Medium	0	50	350	WC
Light	20	60	150	WC
Attacker	10	60	50	FF

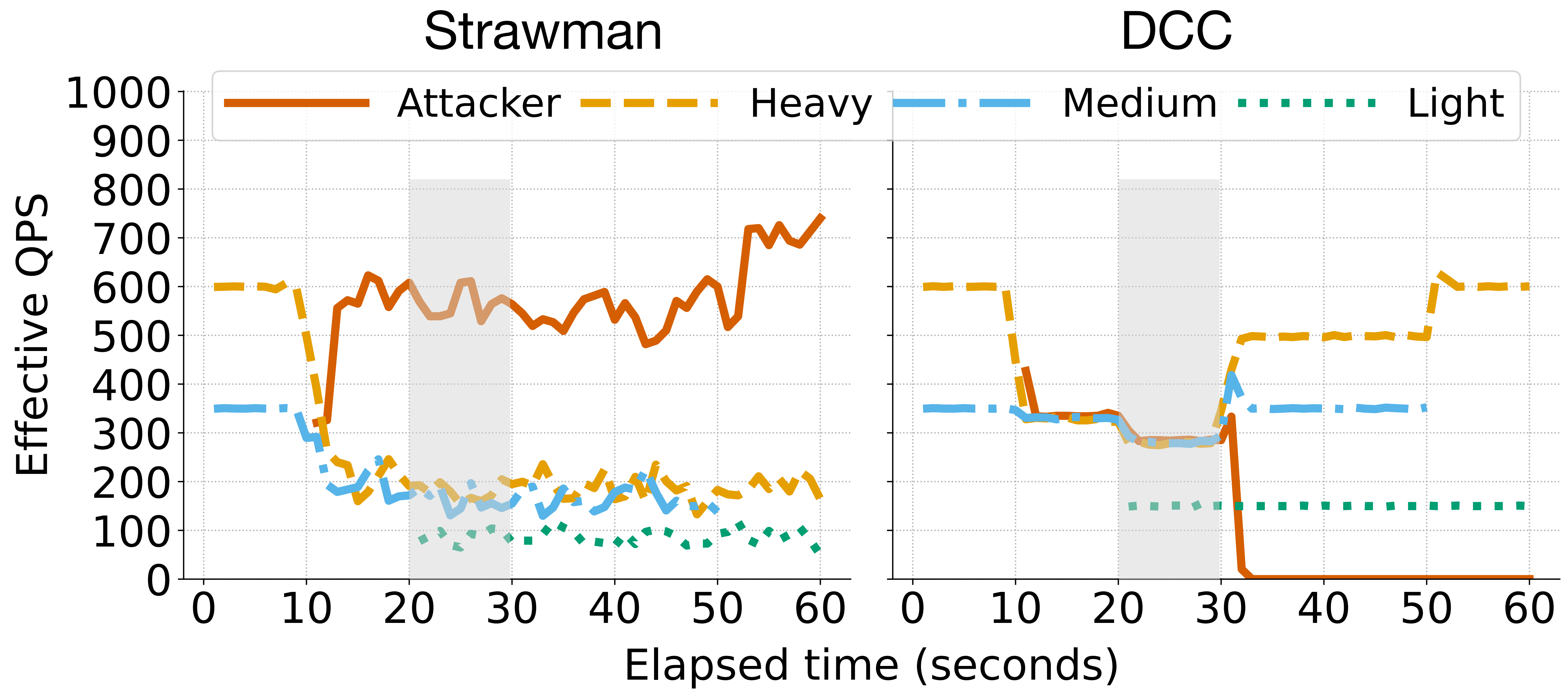


Attacker joins

Evaluation of DCC prototype

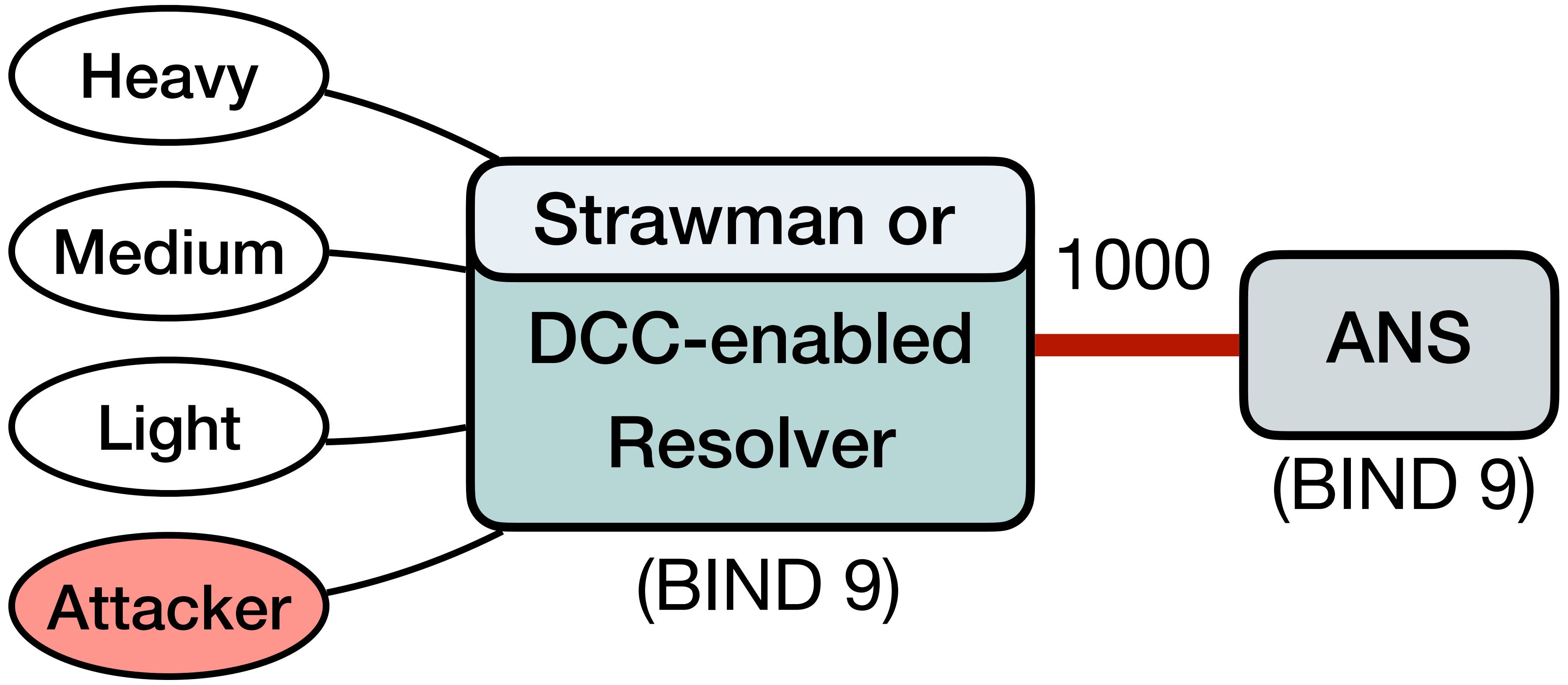


Client	Start	End	Req Rate	Query Pattern
Heavy	0	60	600	WC
Medium	0	50	350	WC
Light	20	60	150	WC
Attacker	10	60	50	FF

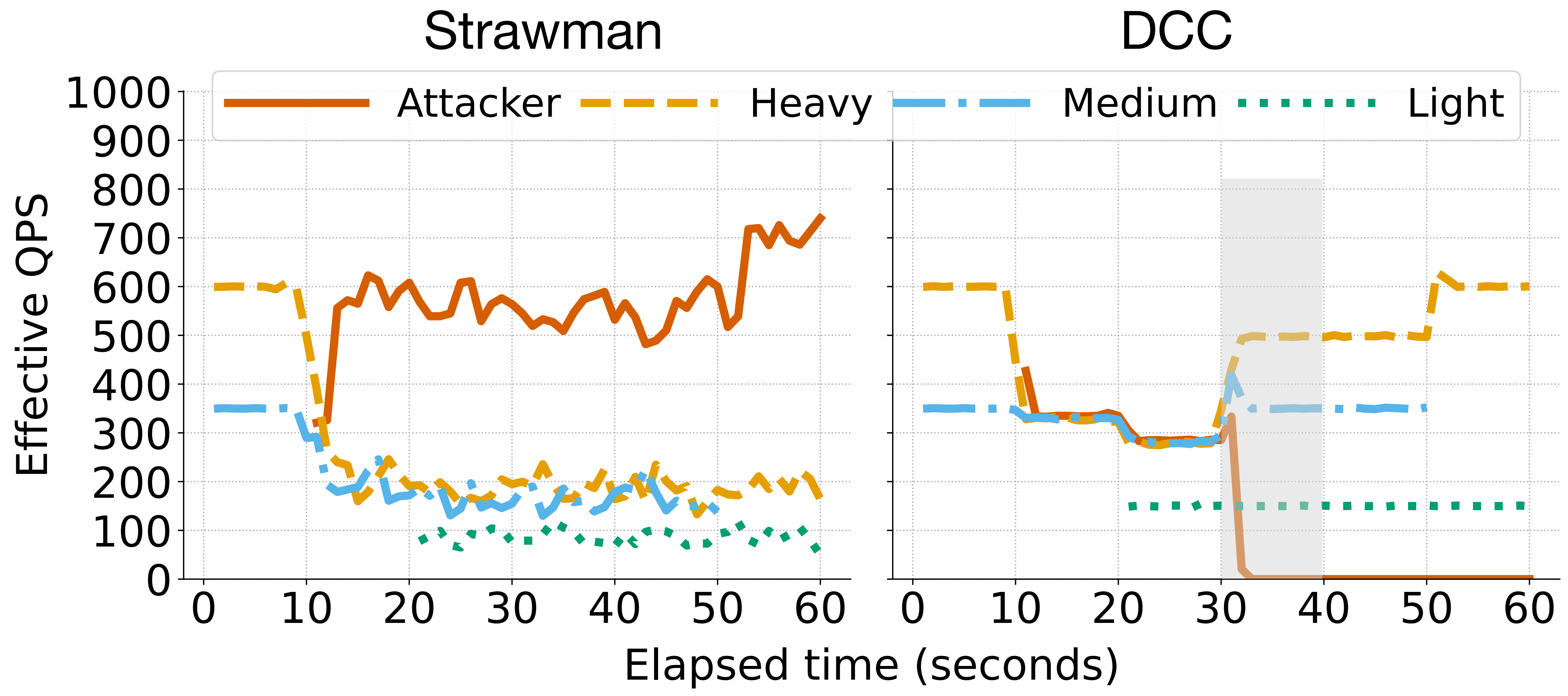


Light client joins

Evaluation of DCC prototype

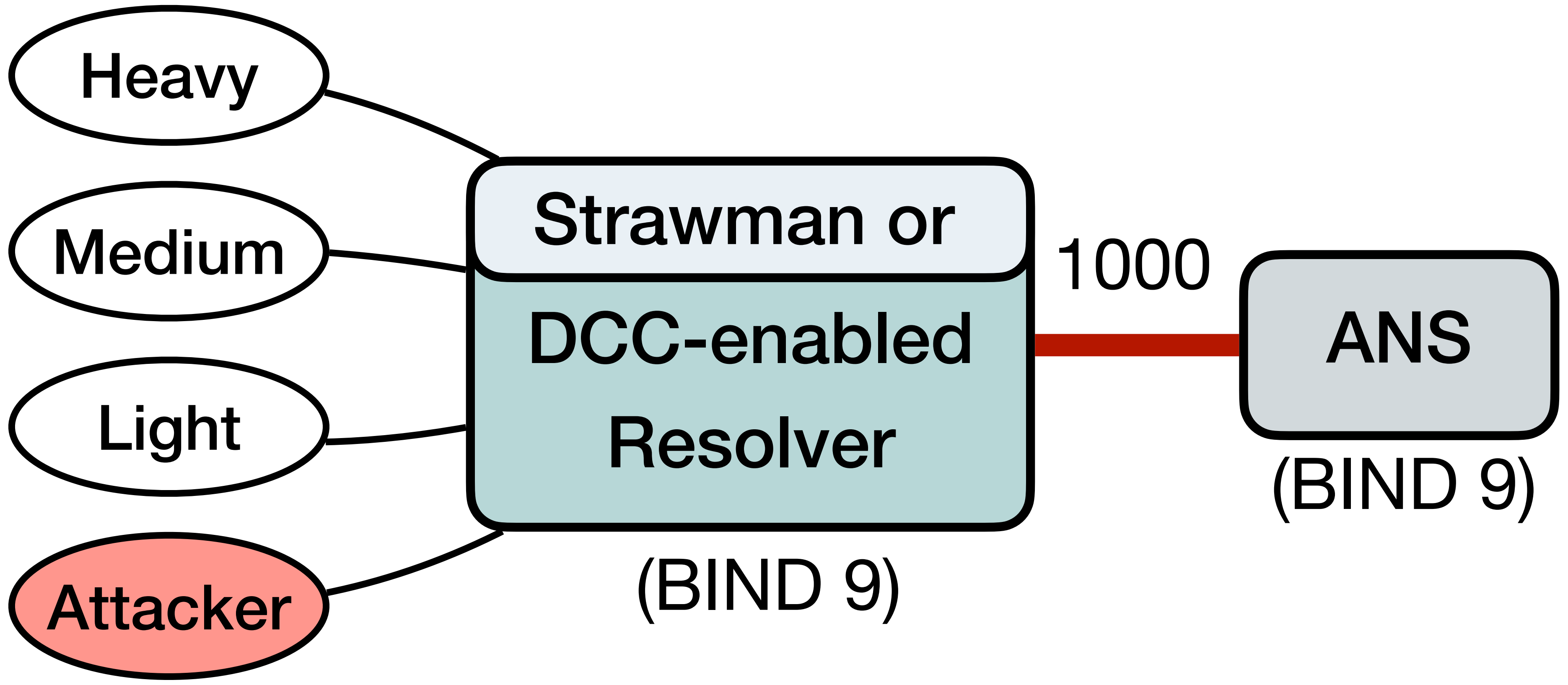


Client	Start	End	Req Rate	Query Pattern
Heavy	0	60	600	WC
Medium	0	50	350	WC
Light	20	60	150	WC
Attacker	10	60	50	FF

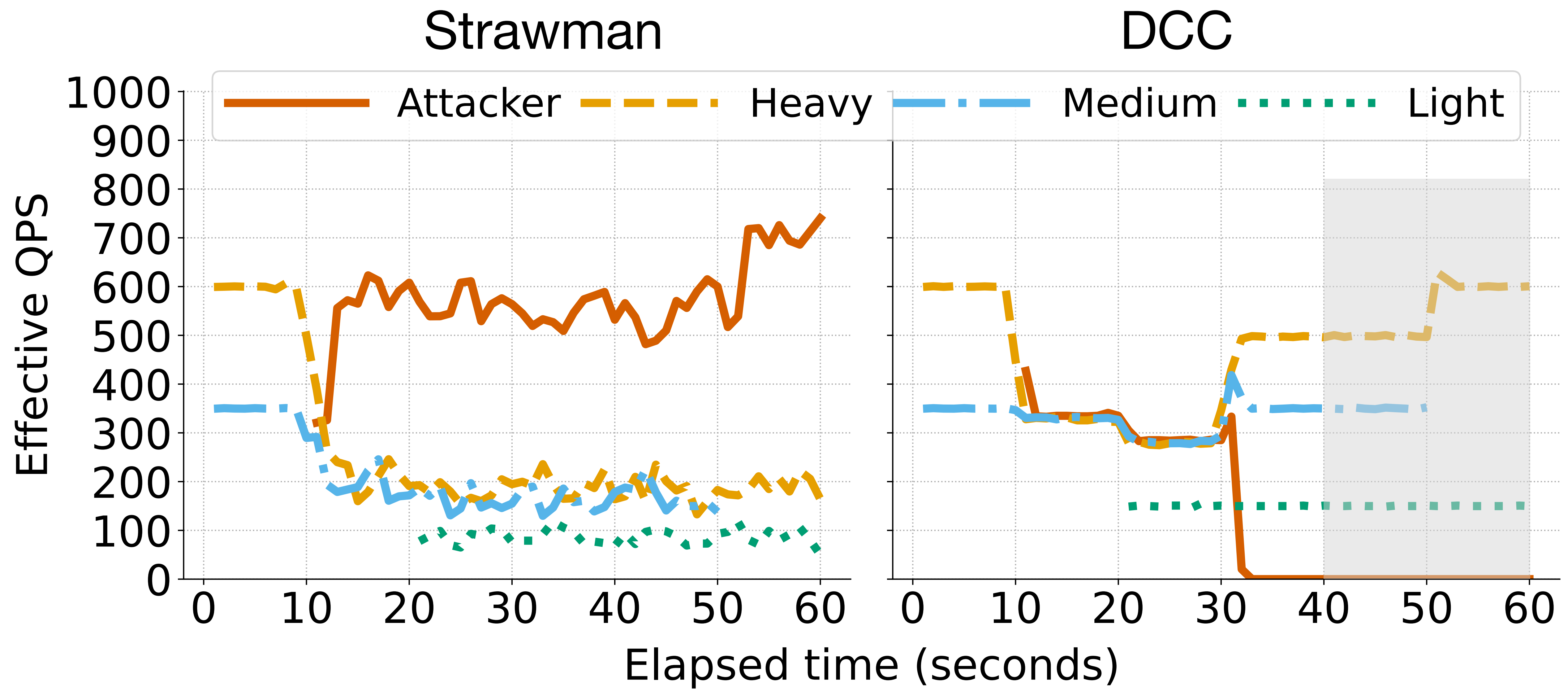


Attacker blocked

Evaluation of DCC prototype



Client	Start	End	Req Rate	Query Pattern
Heavy	0	60	600	WC
Medium	0	50	350	WC
Light	20	60	150	WC
Attacker	10	60	50	FF



Fairness maintained

Concluding remarks

DoS vulnerabilities are **pervasive** in DNS

Availability dilemma: rate limiting as countermeasure and enabler of DoS

DCC provides a **principled** and **generic** defense framework

Thank you!

Questions?

Contact: huayi.duan@inf.ethz.ch

Check paper for details