Lightweight Internet Bandwidth Allocation and Isolation with *Fractional Fair Shares*

Marc Wyss, **Yih-Chun Hu, Vincent Lenders, Roland Meier, Adrian Perrig**

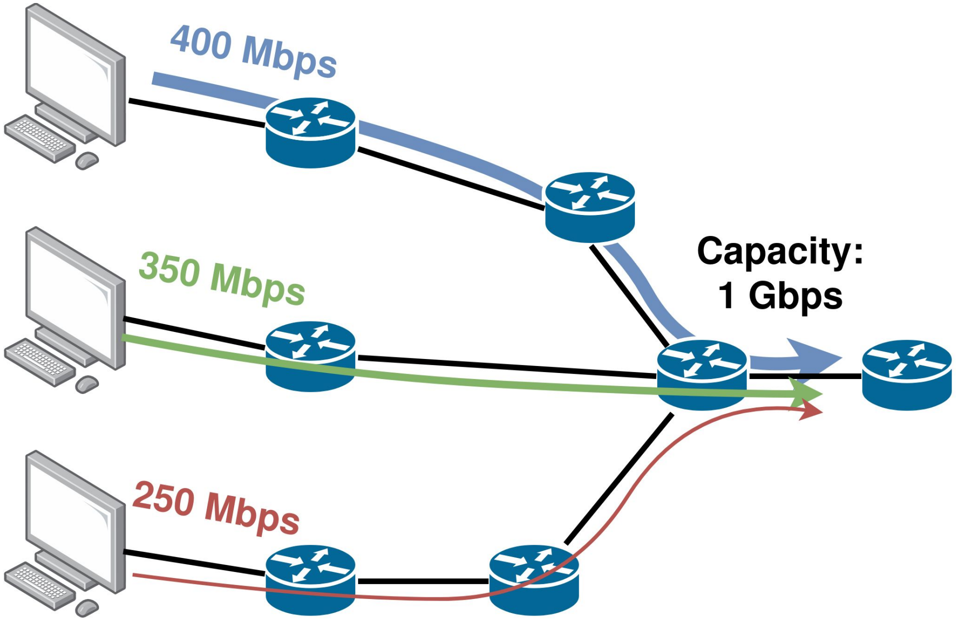
ETH zürich

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

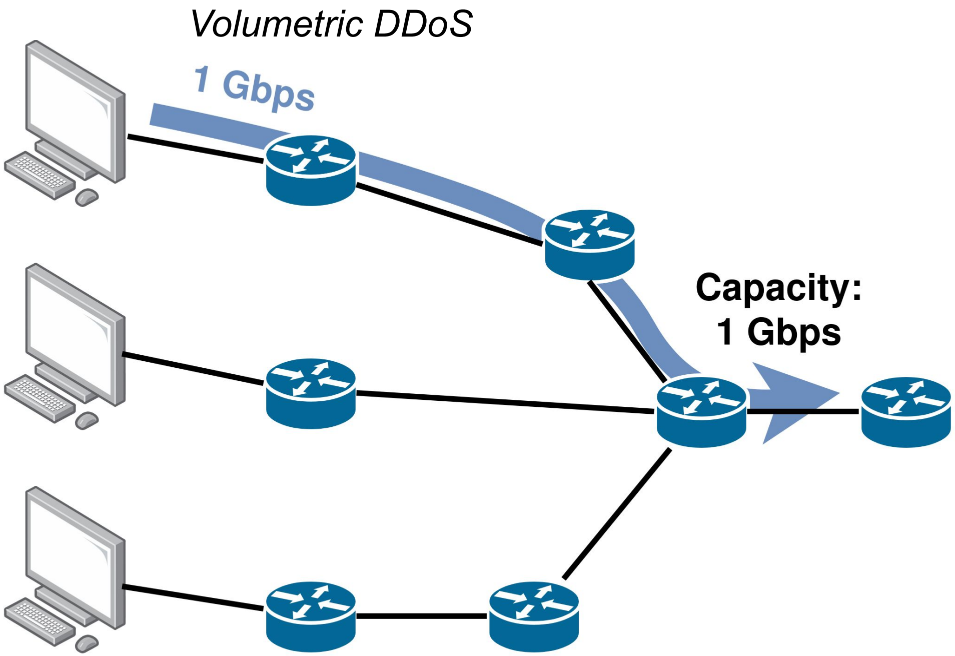
uni.lu UNIVERSITÉ DU LUXEMBOURG

armasuisse

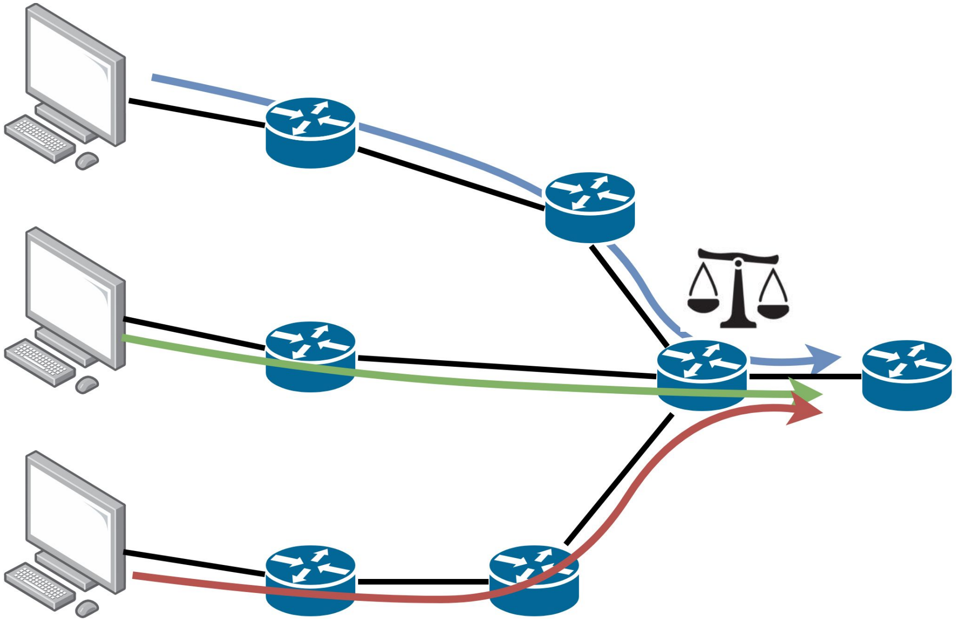
Not all flows are equal



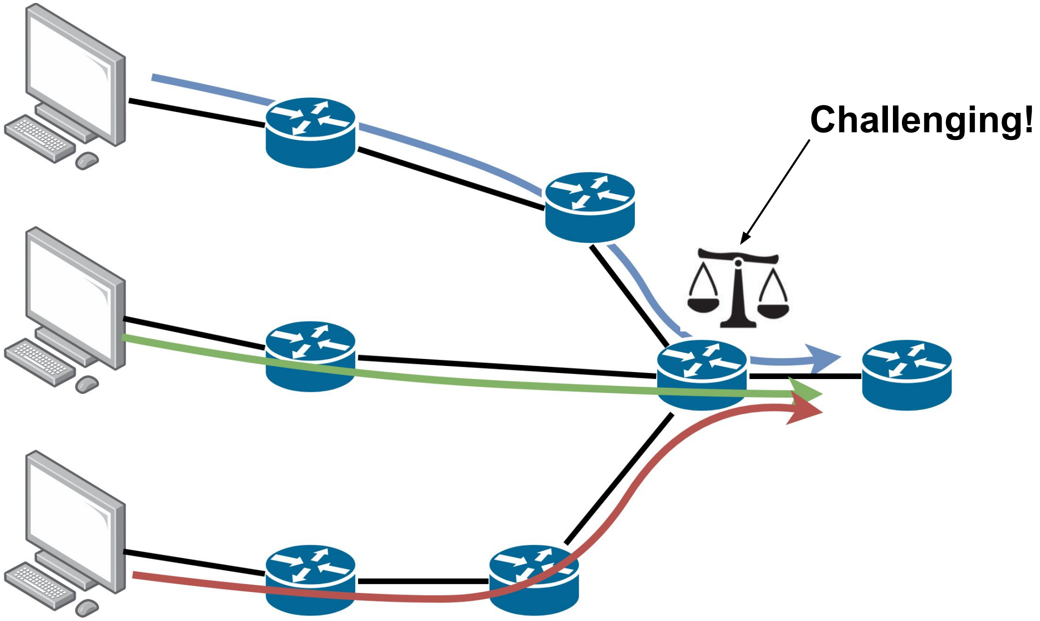
Not all flows are equal



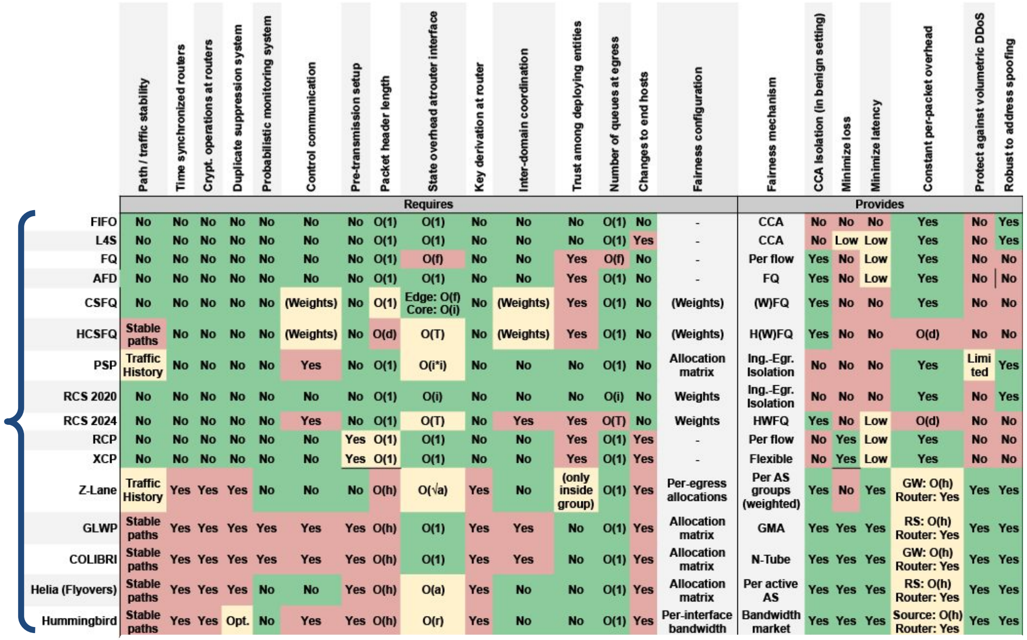
Vision: enforcing fairness in the network



Vision: enforcing fairness in the network



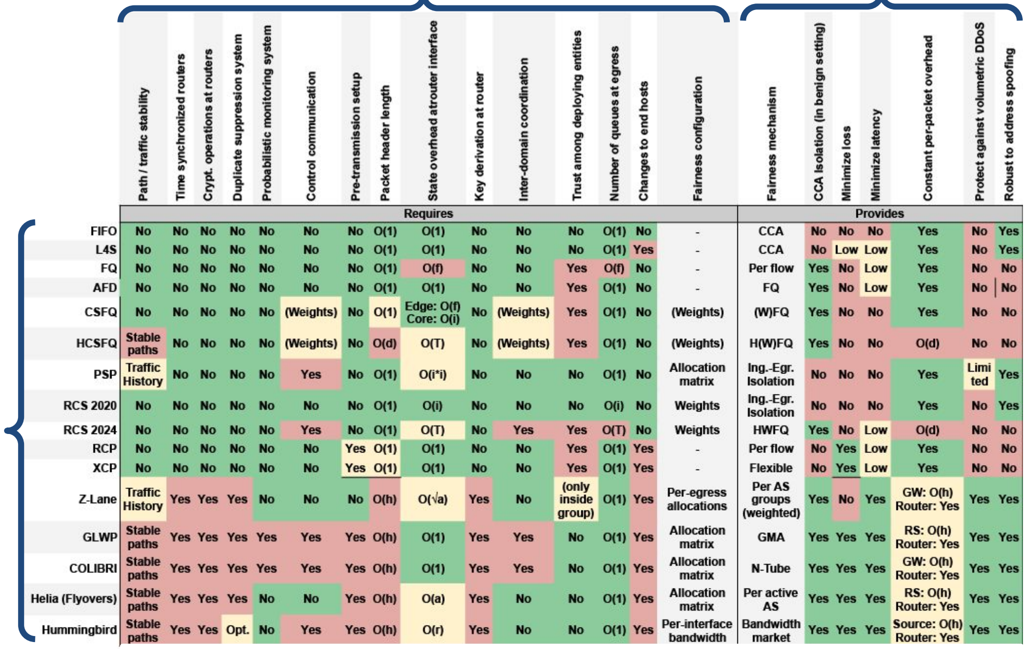
Deployability vs. security



Systems

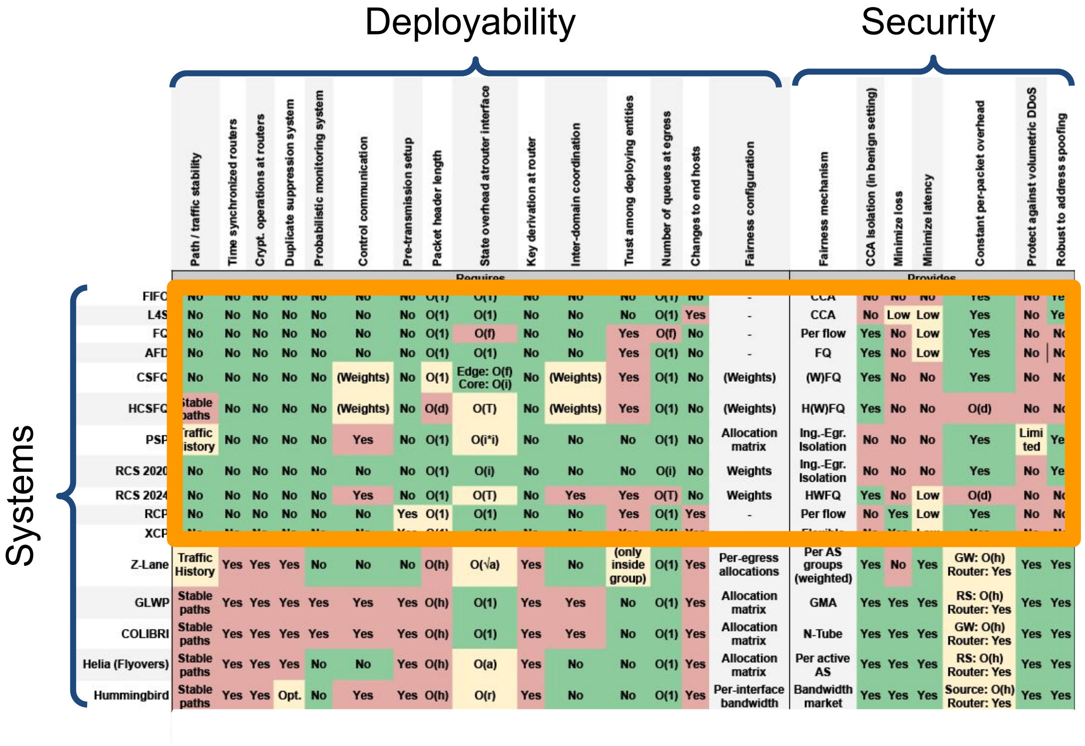
	Path / traffic stability	Time synchronized routers	Crypt. operations at routers	Duplicate suppression system	Probabilistic monitoring system	Control communication	Pre-transmission setup	Packet header length	State overhead atrouter interface	Key derivation at router	Inter-domain coordination	Trust among deploying entities	Number of queues at egress	Changes to end hosts	Fairness configuration	Fairness mechanism	CCA Isolation (in benign setting)	Minimize loss	Minimize latency	Constant per-packet overhead	Protect against volumetric DDoS	Robust to address spoofing
	Requires															Provides						
FIFO	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	No	-	CCA	No	No	No	Yes	No	Yes
L4S	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	Yes	-	CCA	No	Low	Low	Yes	No	Yes
FQ	No	No	No	No	No	No	No	O(1)	O(f)	No	No	Yes	O(f)	No	-	Per flow	Yes	No	Low	Yes	No	No
AFD	No	No	No	No	No	No	No	O(1)	O(1)	No	No	Yes	O(1)	No	-	FQ	Yes	No	Low	Yes	No	No
CSFQ	No	No	No	No	No	(Weights)	No	O(1)	Edge: O(f) Core: O(i)	No	(Weights)	Yes	O(1)	No	(Weights)	(W)FQ	Yes	No	No	Yes	No	No
HCSFQ	Stable paths	No	No	No	No	(Weights)	No	O(d)	O(T)	No	(Weights)	Yes	O(1)	No	(Weights)	H(W)FQ	Yes	No	No	O(d)	No	No
PSP	Traffic History	No	No	No	No	Yes	No	O(1)	O(i*i)	No	No	No	O(1)	No	Allocation matrix	Ing.-Egr. Isolation	No	No	No	Yes	Limi ted	Yes
RCS 2020	No	No	No	No	No	No	No	O(1)	O(i)	No	No	No	O(i)	No	Weights	Ing.-Egr. Isolation	No	No	No	Yes	No	Yes
RCS 2024	No	No	No	No	No	Yes	No	O(1)	O(T)	No	Yes	Yes	O(T)	No	Weights	HWFQ	Yes	No	Low	O(d)	No	No
RCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Per flow	No	Yes	Low	Yes	No	No
XCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Flexible	No	Yes	Low	Yes	No	No
Z-Lane	Traffic History	Yes	Yes	Yes	No	No	No	O(h)	O(√a)	Yes	No	(only inside group)	O(1)	Yes	Per-egress allocations	Per AS groups (weighted)	Yes	No	Yes	GW: O(h) Router: Yes	Yes	Yes
GLWP	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	GMA	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
COLIBRI	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	N-Tube	Yes	Yes	Yes	GW: O(h) Router: Yes	Yes	Yes
Helia (Flyovers)	Stable paths	Yes	Yes	Yes	No	No	Yes	O(h)	O(a)	Yes	No	No	O(1)	Yes	Allocation matrix	Per active AS	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
Hummingbird	Stable paths	Yes	Yes	Opt.	No	Yes	Yes	O(h)	O(r)	Yes	No	No	O(1)	Yes	Per-interface bandwidth	Bandwidth market	Yes	Yes	Yes	Source: O(h) Router: Yes	Yes	Yes

Deployability vs. security



	Deployability															Security						
Systems	Requires															Provides						
	Path / traffic stability	Time synchronized routers	Crypt. operations at routers	Duplicate suppression system	Probabilistic monitoring system	Control communication	Pre-transmission setup	Packet header length	State overhead atrouter interface	Key derivation at router	Inter-domain coordination	Trust among deploying entities	Number of queues at egress	Changes to end hosts	Fairness configuration	Fairness mechanism	CCA Isolation (in benign setting)	Minimize loss	Minimize latency	Constant per-packet overhead	Protect against volumetric DDoS	Robust to address spoofing
FIFO	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	No	-	CCA	No	No	No	Yes	No	Yes
L4S	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	Yes	-	CCA	No	Low	Low	Yes	No	Yes
FQ	No	No	No	No	No	No	No	O(1)	O(f)	No	No	Yes	O(f)	No	-	Per flow	Yes	No	Low	Yes	No	No
AFD	No	No	No	No	No	No	No	O(1)	O(1)	No	No	Yes	O(1)	No	-	FQ	Yes	No	Low	Yes	No	No
CSFQ	No	No	No	No	No	(Weights)	No	O(1)	Edge: O(f) Core: O(i)	No	(Weights)	Yes	O(1)	No	(Weights)	(W)FQ	Yes	No	No	Yes	No	No
HCSFQ	Stable paths	No	No	No	No	(Weights)	No	O(d)	O(T)	No	(Weights)	Yes	O(1)	No	(Weights)	H(W)FQ	Yes	No	No	O(d)	No	No
PSP	Traffic History	No	No	No	No	Yes	No	O(1)	O(i*i)	No	No	No	O(1)	No	Allocation matrix	Ing.-Egr. Isolation	No	No	No	Yes	Limi ted	Yes
RCS 2020	No	No	No	No	No	No	No	O(1)	O(i)	No	No	No	O(i)	No	Weights	Ing.-Egr. Isolation	No	No	No	Yes	No	Yes
RCS 2024	No	No	No	No	No	Yes	No	O(1)	O(T)	No	Yes	Yes	O(T)	No	Weights	HWFQ	Yes	No	Low	O(d)	No	No
RCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Per flow	No	Yes	Low	Yes	No	No
XCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Flexible	No	Yes	Low	Yes	No	No
Z-Lane	Traffic History	Yes	Yes	Yes	No	No	No	O(h)	O(√a)	Yes	No	(only inside group)	O(1)	Yes	Per-egress allocations	Per AS groups (weighted)	Yes	No	Yes	GW: O(h) Router: Yes	Yes	Yes
GLWP	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	GMA	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
COLIBRI	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	N-Tube	Yes	Yes	Yes	GW: O(h) Router: Yes	Yes	Yes
Helia (Flyovers)	Stable paths	Yes	Yes	Yes	No	No	Yes	O(h)	O(a)	Yes	No	No	O(1)	Yes	Allocation matrix	Per active AS	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
Hummingbird	Stable paths	Yes	Yes	Opt.	No	Yes	Yes	O(h)	O(r)	Yes	No	No	O(1)	Yes	Per-interface bandwidth	Bandwidth market	Yes	Yes	Yes	Source: O(h) Router: Yes	Yes	Yes

Deployability vs. security



Deployability

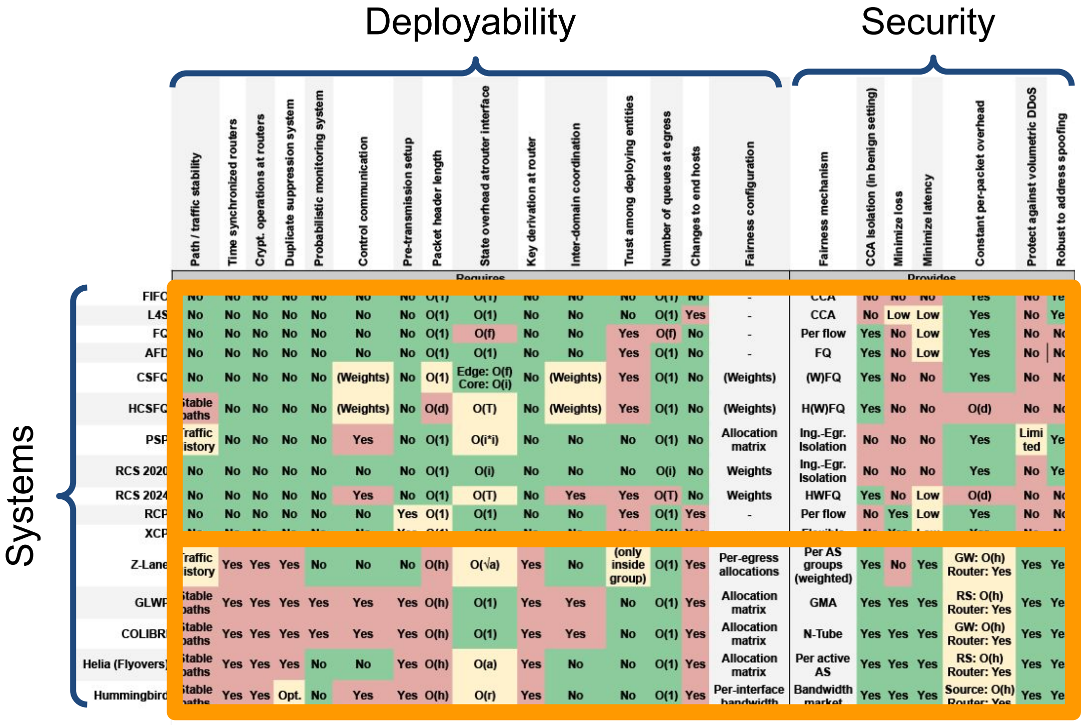
Security

Systems

	Requires															Provides						
	Path / traffic stability	Time synchronized routers	Crypt. operations at routers	Duplicate suppression system	Probabilistic monitoring system	Control communication	Pre-transmission setup	Packet header length	State overhead at router interface	Key derivation at router	Inter-domain coordination	Trust among deploying entities	Number of queues at egress	Changes to end hosts	Fairness configuration	Fairness mechanism	CCA Isolation (in benign setting)	Minimize loss	Minimize latency	Constant per-packet overhead	Protect against volumetric DDoS	Robust to address spoofing
FIFO	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	No	-	CCA	No	No	No	Yes	No	Yes
L4S	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	Yes	-	CCA	No	Low	Low	Yes	No	Yes
FQ	No	No	No	No	No	No	No	O(1)	O(f)	No	No	Yes	O(f)	No	-	Per flow	Yes	No	Low	Yes	No	No
AFD	No	No	No	No	No	No	No	O(1)	O(1)	No	No	Yes	O(1)	No	-	FQ	Yes	No	Low	Yes	No	No
CSFQ	No	No	No	No	No	(Weights)	No	O(1)	Edge: O(f) Core: O(i)	No	(Weights)	Yes	O(1)	No	(Weights)	(W)FQ	Yes	No	No	Yes	No	No
HCSFQ	Stable paths	No	No	No	No	(Weights)	No	O(d)	O(T)	No	(Weights)	Yes	O(1)	No	(Weights)	H(W)FQ	Yes	No	No	O(d)	No	No
PSP	Traffic history	No	No	No	No	Yes	No	O(1)	O(i*i)	No	No	No	O(1)	No	Allocation matrix	Ing.-Egr. Isolation	No	No	No	Yes	Limited	Yes
RCS 2020	No	No	No	No	No	No	No	O(1)	O(i)	No	No	No	O(i)	No	Weights	Ing.-Egr. Isolation	No	No	No	Yes	No	Yes
RCS 2024	No	No	No	No	No	Yes	No	O(1)	O(T)	No	Yes	Yes	O(T)	No	Weights	HWFQ	Yes	No	Low	O(d)	No	No
RCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Per flow	No	Yes	Low	Yes	No	No
XCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Flexible	No	Yes	Low	Yes	No	No
Z-Lane	Traffic History	Yes	Yes	Yes	No	No	No	O(h)	O(√a)	Yes	No	(only inside group)	O(1)	Yes	Per-egress allocations	Per AS groups (weighted)	Yes	No	Yes	GW: O(h) Router: Yes	Yes	Yes
GLWP	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	GMA	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
COLIBRI	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	N-Tube	Yes	Yes	Yes	GW: O(h) Router: Yes	Yes	Yes
Helia (Flyovers)	Stable paths	Yes	Yes	Yes	No	No	Yes	O(h)	O(a)	Yes	No	No	O(1)	Yes	Allocation matrix	Per active AS	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
Hummingbird	Stable paths	Yes	Yes	Opt.	No	Yes	Yes	O(h)	O(r)	Yes	No	No	O(1)	Yes	Per-interface bandwidth	Bandwidth market	Yes	Yes	Yes	Source: O(h) Router: Yes	Yes	Yes

Simple deployment, modest security

Deployability vs. security



Deployability
Security
Systems

	Path / traffic stability	Time synchronized routers	Crypt. operations at routers	Duplicate suppression system	Probabilistic monitoring system	Control communication	Pre-transmission setup	Packet header length	State overhead at router interface	Key derivation at router	Inter-domain coordination	Trust among deploying entities	Number of queues at egress	Changes to end hosts	Fairness configuration	Fairness mechanism	CCA Isolation (in benign setting)	Minimize loss	Minimize latency	Constant per-packet overhead	Protect against volumetric DDoS	Robust to address spoofing
	Requires															Provides						
FIFO	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	No	-	CCA	No	No	No	Yes	No	Yes
L4S	No	No	No	No	No	No	No	O(1)	O(1)	No	No	No	O(1)	Yes	-	CCA	No	Low	Low	Yes	No	Yes
FQ	No	No	No	No	No	No	No	O(1)	O(f)	No	No	Yes	O(f)	No	-	Per flow	Yes	No	Low	Yes	No	No
AFD	No	No	No	No	No	No	No	O(1)	O(1)	No	No	Yes	O(1)	No	-	FQ	Yes	No	Low	Yes	No	No
CSFQ	No	No	No	No	No	(Weights)	No	O(1)	Edge: O(f) Core: O(i)	No	(Weights)	Yes	O(1)	No	(Weights)	(W)FQ	Yes	No	No	Yes	No	No
HCSFQ	Stable paths	No	No	No	No	(Weights)	No	O(d)	O(T)	No	(Weights)	Yes	O(1)	No	(Weights)	H(W)FQ	Yes	No	No	O(d)	No	No
PSP	Traffic history	No	No	No	No	Yes	No	O(1)	O(i*i)	No	No	No	O(1)	No	Allocation matrix	Ing.-Egr. Isolation	No	No	No	Yes	Limited	Yes
RCS 2020	No	No	No	No	No	No	No	O(1)	O(i)	No	No	No	O(i)	No	Weights	Ing.-Egr. Isolation	No	No	No	Yes	No	Yes
RCS 2024	No	No	No	No	No	Yes	No	O(1)	O(T)	No	Yes	Yes	O(T)	No	Weights	HWFQ	Yes	No	Low	O(d)	No	No
RCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Per flow	No	Yes	Low	Yes	No	No
XCP	No	No	No	No	No	No	Yes	O(1)	O(1)	No	No	Yes	O(1)	Yes	-	Flexible	No	Yes	Low	Yes	No	No
Z-Lane	Traffic history	Yes	Yes	Yes	No	No	No	O(h)	O(√a)	Yes	No	(only inside group)	O(1)	Yes	Per-egress allocations	Per AS groups (weighted)	Yes	No	Yes	GW: O(h) Router: Yes	Yes	Yes
GLWP	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	GMA	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
COLIBRI	Stable paths	Yes	Yes	Yes	Yes	Yes	Yes	O(h)	O(1)	Yes	Yes	No	O(1)	Yes	Allocation matrix	N-Tube	Yes	Yes	Yes	GW: O(h) Router: Yes	Yes	Yes
Helia (Flyovers)	Stable paths	Yes	Yes	Yes	No	No	Yes	O(h)	O(a)	Yes	No	No	O(1)	Yes	Allocation matrix	Per active AS	Yes	Yes	Yes	RS: O(h) Router: Yes	Yes	Yes
Hummingbird	Stable paths	Yes	Yes	Opt.	No	Yes	Yes	O(h)	O(r)	Yes	No	No	O(1)	Yes	Per-interface bandwidth	Bandwidth market	Yes	Yes	Yes	Source: O(h) Router: Yes	Yes	Yes

Simple deployment, modest security

Complex deployment, strong security

Can we achieve both security and deployability?

Can we achieve both security and deployability?

Security

Traffic stream isolation

Resilience against address spoofing

Minimizing assumptions

Enforce fair allocations through end-to-end in-network bandwidth isolation.

Can we achieve both security and deployability?

Security

Traffic stream isolation

Resilience against address spoofing

Minimizing assumptions

Allocating bandwidth per IP address is vulnerable to spoofing, allowing attackers to consume bandwidth intended for legitimate senders.

Can we achieve both security and deployability?

Security

Traffic stream isolation

Resilience against address spoofing

Minimizing assumptions

Common assumptions in the literature:

- *Single bandwidth bottleneck along a path.*
- *Congestion only at the last mile.*
- *Trust among independent entities.*
- *It is enough to track the K largest flows.*

Can we achieve both security and deployability?

Security

- Traffic stream isolation
- Resilience against address spoofing
- Minimizing assumptions

Deployment

- Simple router operations only
- Performance
- Scalability
- No setup request needed
- No dependency on other systems
- Incremental deployment

Can we achieve both security and deployability?

Security

Traffic stream isolation

Resilience against address spoofing

Minimizing assumptions

Achieving all those properties is challenging!

Deployment

Simple router operations only

Performance

Scalability

No setup request needed

No dependency on other systems

Incremental deployment

Can we achieve both security and deployability?

Security

- Traffic stream isolation
- Resilience against address spoofing
- Minimizing assumptions

⚡

Deployment

- Simple router operations only
- Performance
- Scalability
- No setup request needed
- No dependency on other systems
- Incremental deployment

Can we achieve both security and deployability?

Security

- Traffic stream isolation
- Resilience against address spoofing
- Minimizing assumptions

Deployment

- Simple router operations only
- Performance
- Scalability
- No setup request needed
- No dependency on other systems
- Incremental deployment

Fractional Fair Shares (FFS)

Security

- ☑ Traffic stream isolation
- ☑ Resilience against address spoofing
- ☑ Minimizing assumptions

Deployment

- ☑ Simple router operations only
- ☑ Performance
- ☑ Scalability
- ☑ No setup request needed
- ☑ No dependency on other systems
- ☑ Incremental deployment

Fractional Fair Shares (FFS)

Security

- ☑ Traffic stream isolation
- ☑ Resilience against address spoofing
- ☑ Minimizing assumptions

Trade-offs

- ⚠ Introduce packet labels
- ⚠ Probabilistic forwarding guarantees

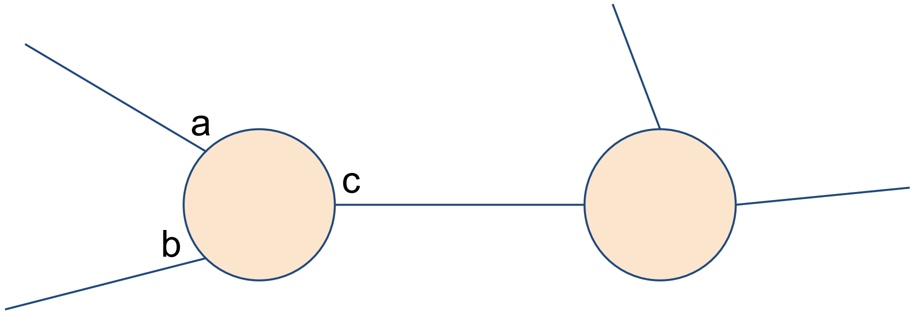
Deployment

- ☑ Simple router operations only
- ☑ Performance
- ☑ Scalability
- ☑ No setup request needed
- ☑ No dependency on other systems
- ☑ Incremental deployment

MAIN CONTRIBUTION

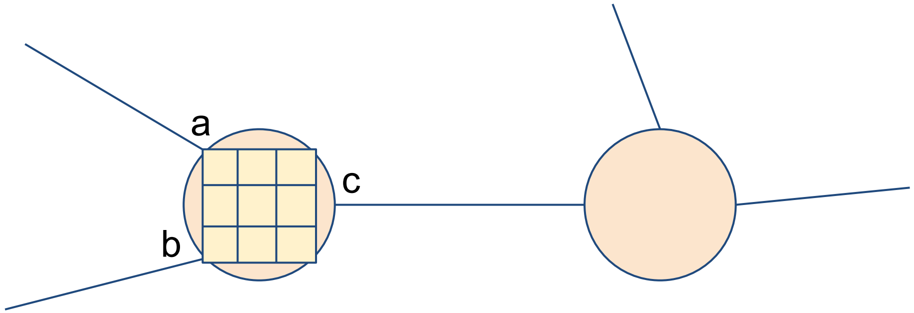
Simple cryptography-free algorithm ensuring **communication guarantees** under volumetric DDoS and address spoofing attacks.

Fractional Fair Shares (FFS)



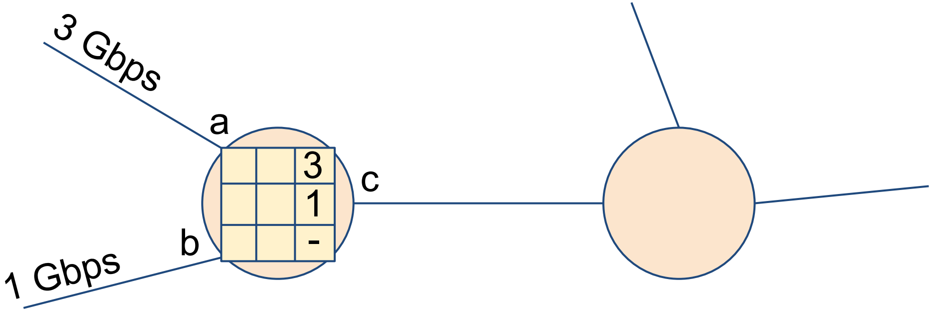
1. **Fairness matrix** → local fairness definition

Fractional Fair Shares (FFS)



1. **Fairness matrix** → local fairness definition

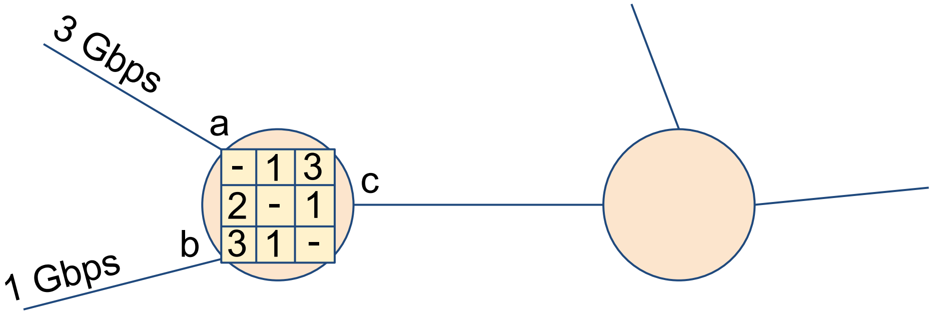
Fractional Fair Shares (FFS)



During congestion, how much of the **capacity of egress c** should be allocated to **ingresses a and b**?

1. **Fairness matrix** → local fairness definition

Fractional Fair Shares (FFS)

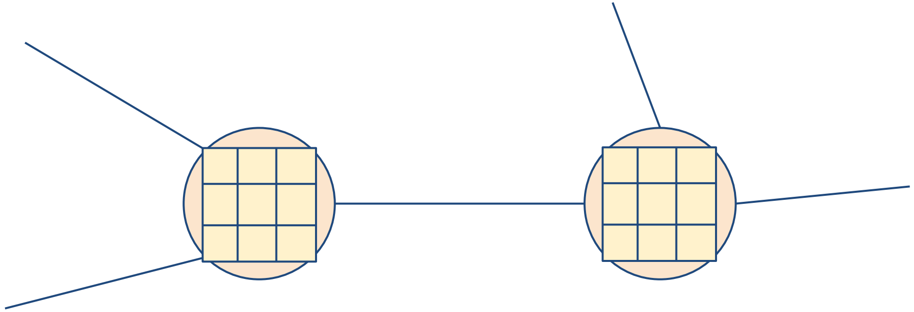


-	1	3
2	-	1
3	1	-

Repeat for all egresses.

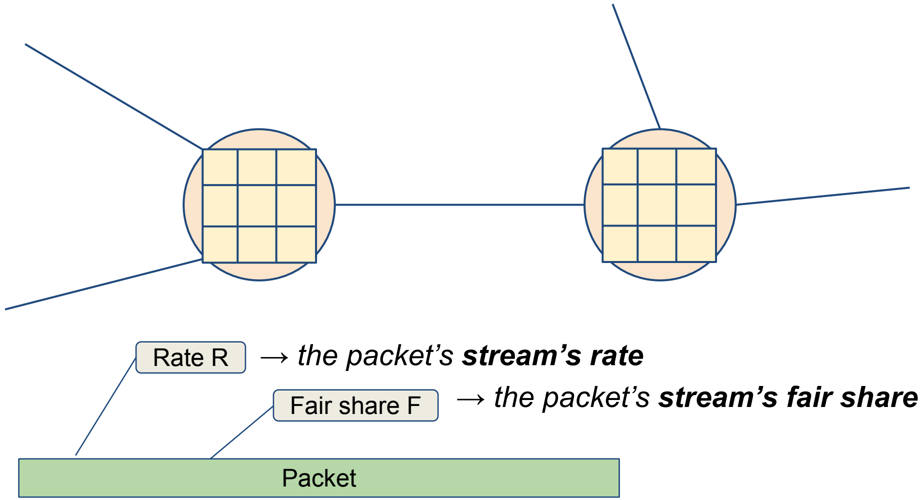
1. **Fairness matrix** → local fairness definition

Fractional Fair Shares (FFS)



1. **Fairness matrix** → local fairness definition

Fractional Fair Shares (FFS)



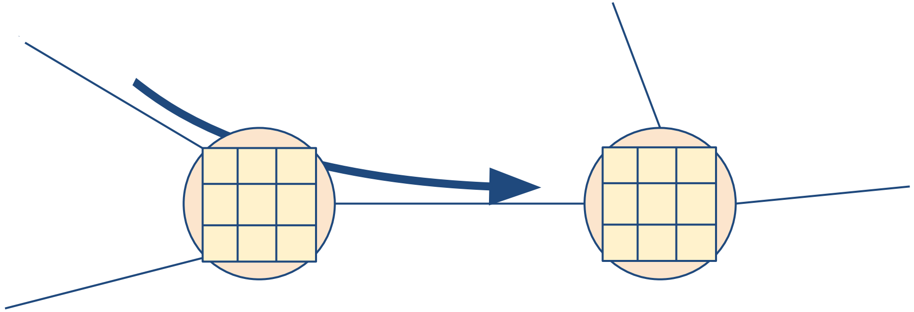
Rate R → *the packet's **stream's rate***

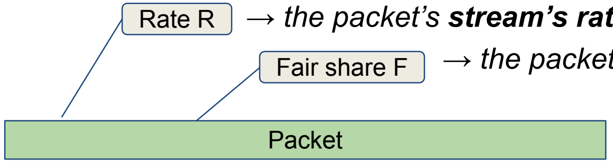
Fair share F → *the packet's **stream's fair share***

Packet

1. **Fairness matrix** → local fairness definition
2. **Packet labels** → propagate fairness globally

Fractional Fair Shares (FFS)



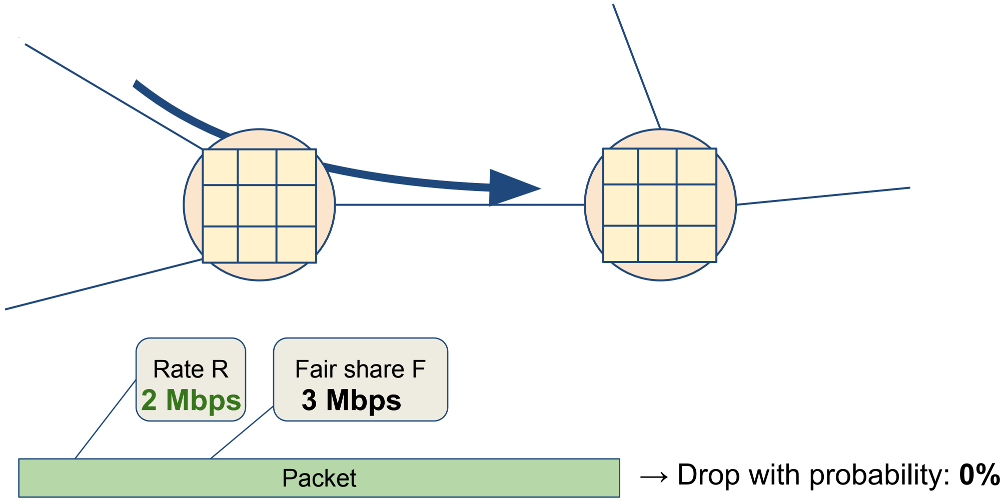


→ *the packet's **stream's rate***

→ *the packet's **stream's fair share***

1. **Fairness matrix** → local fairness definition
2. **Packet labels** → propagate fairness globally

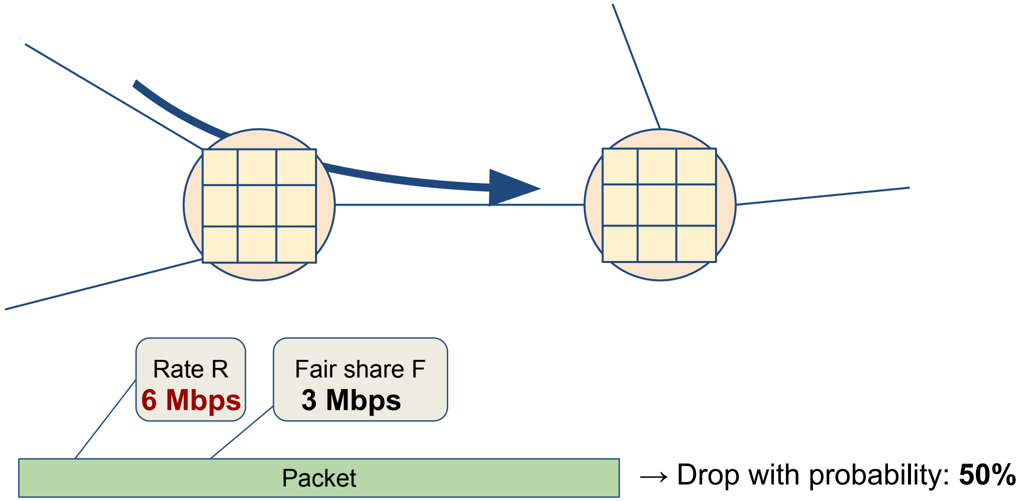
Fractional Fair Shares (FFS)



→ Drop with probability: **0%**

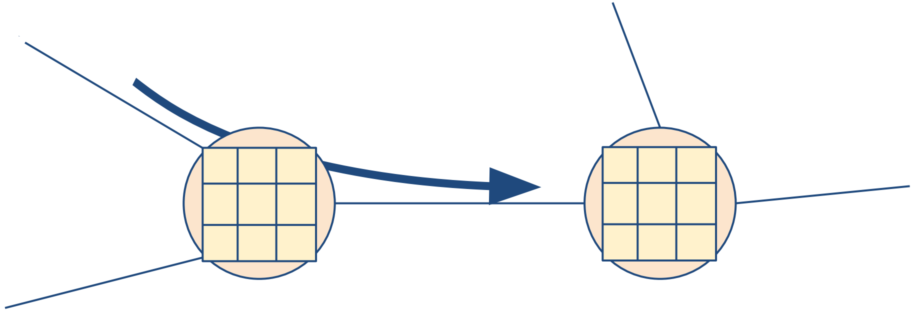
1. **Fairness matrix** → local fairness definition
2. **Packet labels** → propagate fairness globally
3. **Probabilistic forwarding** → scalability / mitigate spoofing attacks

Fractional Fair Shares (FFS)



1. **Fairness matrix** → local fairness definition
2. **Packet labels** → propagate fairness globally
3. **Probabilistic forwarding** → scalability / mitigate spoofing attacks

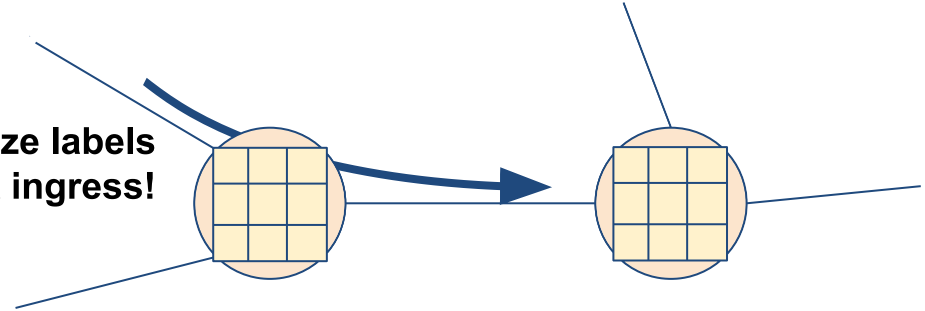
Fractional Fair Shares (FFS)



No cryptographic source authentication needed.

1. **Fairness matrix** → local fairness definition
2. **Packet labels** → propagate fairness globally
3. **Probabilistic forwarding** → scalability / mitigate spoofing attacks

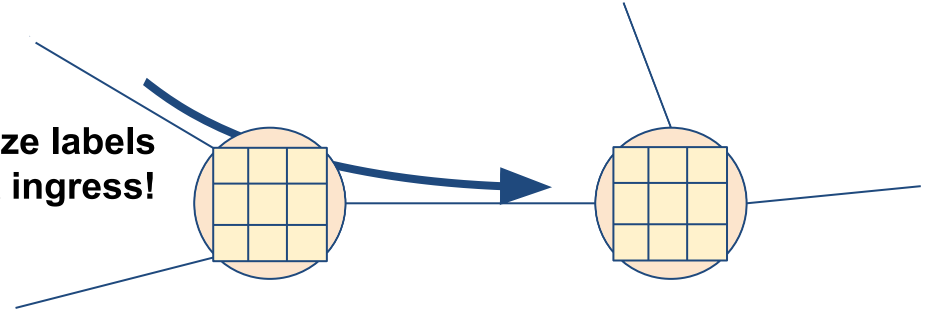
Fractional Fair Shares (FFS)



Normalize labels at ingress!

1. **Fairness matrix** → local fairness definition
2. **Packet labels** → propagate fairness globally
3. **Probabilistic forwarding** → scalability / mitigate spoofing attacks
4. **Label normalization** → prevent attacks exploiting labels

Fractional Fair Shares (FFS)

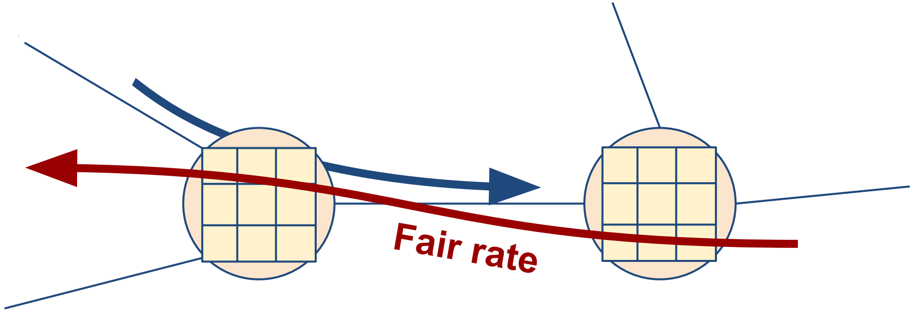


Normalize labels at ingress!

Each node treats <u>all other nodes as untrusted</u> entities.

1.	**Fairness matrix**	→	local fairness definition
2.	**Packet labels**	→	propagate fairness globally
3.	**Probabilistic forwarding**	→	scalability / mitigate spoofing attacks
4.	**Label normalization**	→	prevent attacks exploiting labels

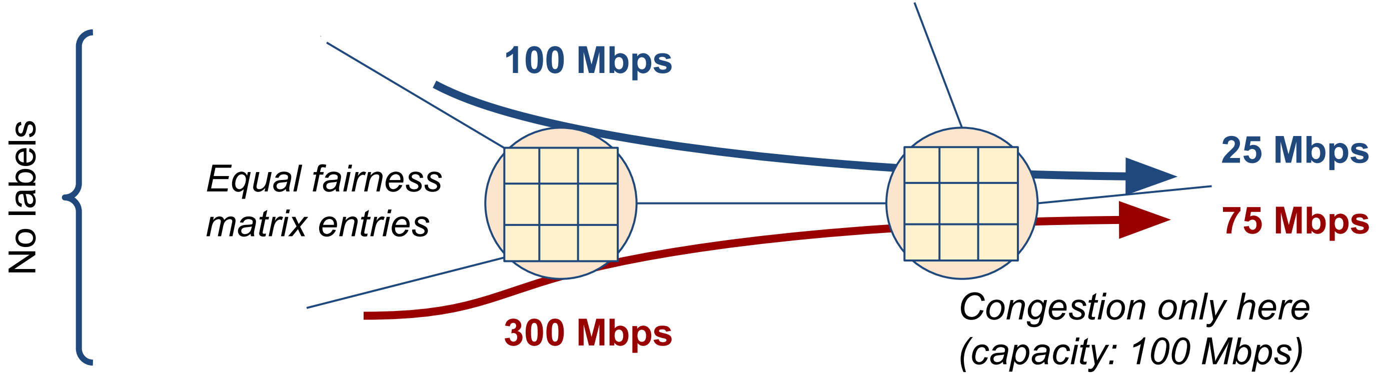
Fractional Fair Shares (FFS)



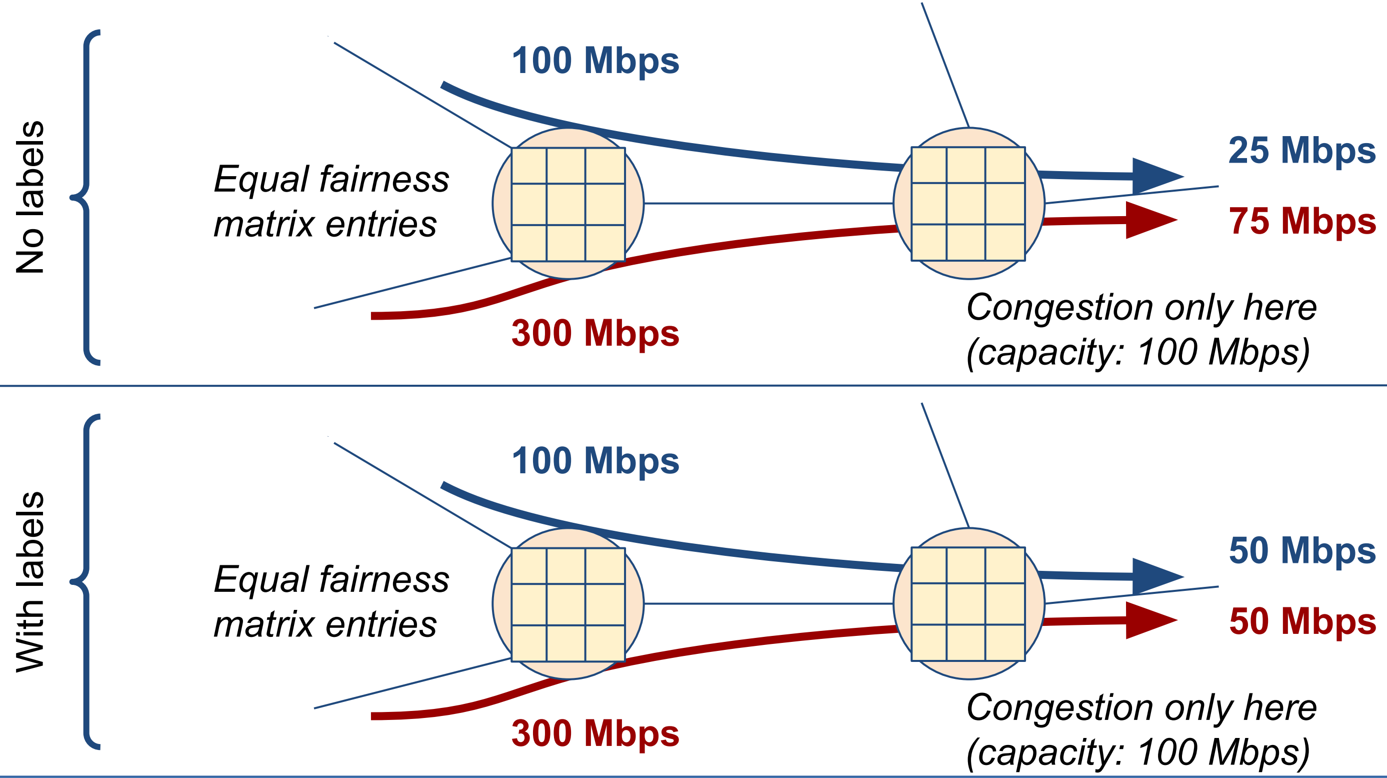
ⓘ End host support needed

1.	**Fairness matrix**	→	local fairness definition
2.	**Packet labels**	→	propagate fairness globally
3.	**Probabilistic forwarding**	→	scalability / mitigate spoofing attacks
4.	**Label normalization**	→	prevent attacks exploiting labels
5.	**Rate feedback** (optional)	→	end hosts learn fair share

Why do we need packet labels?


No labels
100 Mbps
Equal fairness matrix entries
300 Mbps
25 Mbps
75 Mbps
Congestion only here (capacity: 100 Mbps)

Why do we need packet labels?



Evaluation

Fairness / QoS

Network utilization

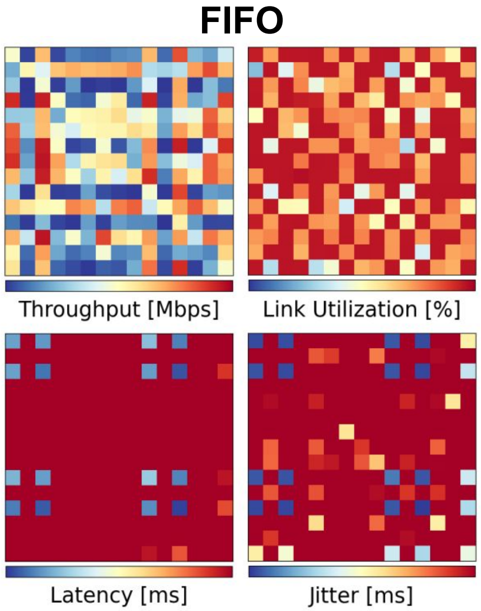
Security analysis

High-speed impl.

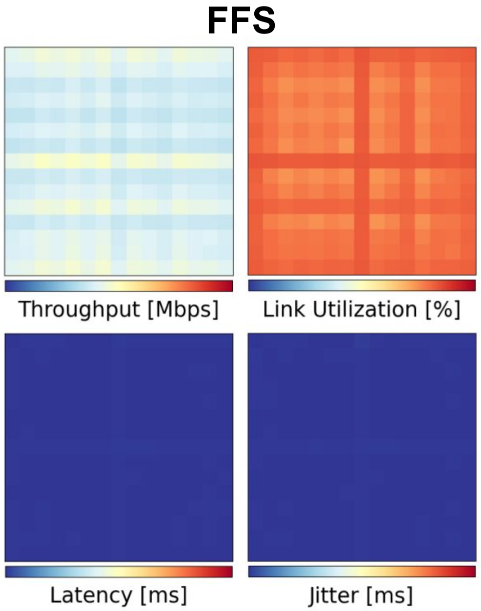
Evaluation

Fairness / QoS | Network utilization | Security analysis | High-speed impl.

uneven allocations



balanced allocations



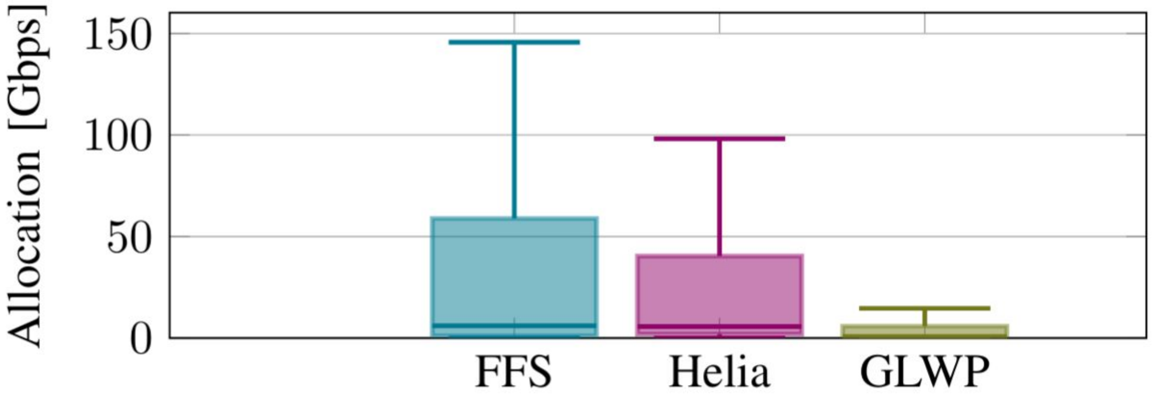
Evaluation

Fairness / QoS

Network utilization

Security analysis

High-speed impl.



Evaluation

Fairness / QoS | Network utilization | **Security analysis** | High-speed impl.

Corollary 1. **(simplified)**

For traffic from ingress i to egress j, at least up to **a rate of M(i,j) is guaranteed to be forwarded**, irrespective of the rate and FFS distributions of streams originating from other ingresses.

Evaluation

Fairness / QoS | Network utilization | **Security analysis** | High-speed impl.

Local

Corollary 1. **(simplified)**

For traffic from ingress i to egress j, at least up to **a rate of M(i,j) is guaranteed to be forwarded**, irrespective of the rate and FFS distributions of streams originating from other ingresses.

Evaluation

Fairness / QoS | Network utilization | **Security analysis** | High-speed impl.

Local

Corollary 1. **(simplified)**

For traffic from ingress i to egress j, at least up to **a rate of M(i,j) is guaranteed to be forwarded**, irrespective of the rate and FFS distributions of streams originating from other ingresses.

Global

Lemma 2. **(simplified)**

A stream s gets a **worst-case minimum guaranteed allocation** of $\min_{x=0}^{\ell}(\frac{\prod_{t=0}^{x}\mathrm{M}^{n^t}_{(i^t,j^t)}}{\prod_{t=0}^{x-1}\mathrm{C}^{n^t}_{j^t}})$

Evaluation

Fairness / QoS | Network utilization | Security analysis | **High-speed impl.**

FFS’ overhead is _constant_:

- memory (few dozens of bytes per interface)
- number of queues (only one)
- packet header size (few bytes)
- processing time (102 ns)

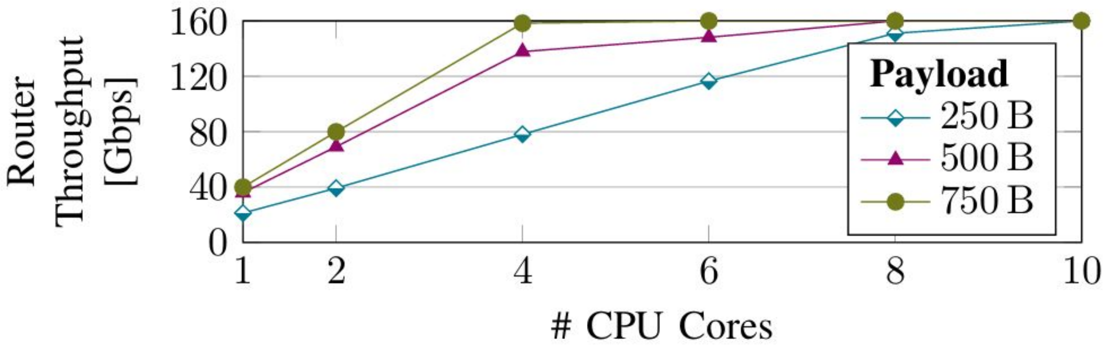
Evaluation

Fairness / QoS

Network utilization

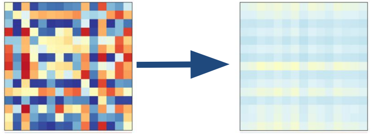
Security analysis

High-speed impl.



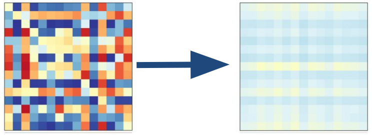
Conclusion

- FFS enforces fair bandwidth allocations directly in the network:



Conclusion

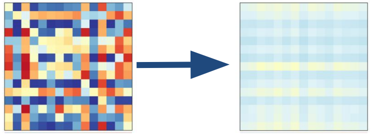
- FFS enforces fair bandwidth allocations directly in the network:



- Robust against:
  - Volumetric DDoS
  - Attacker creating many flows
  - Source address spoofing

Conclusion

- FFS enforces fair bandwidth allocations directly in the network:

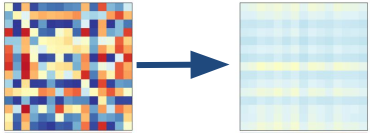


- Robust against:
  - Volumetric DDoS
  - Attacker creating many flows
  - Source address spoofing

- No particular (trust) assumptions needed.

Conclusion

- FFS enforces fair bandwidth allocations directly in the network:



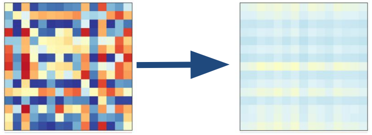
- Robust against:
    - Volumetric DDoS
    - Attacker creating many flows
    - Source address spoofing

- No particular (trust) assumptions needed.

- No dependencies on other systems.

Conclusion

- FFS enforces fair bandwidth allocations directly in the network:



- Robust against:
  - Volumetric DDoS
  - Attacker creating many flows
  - Source address spoofing

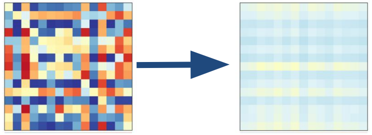
- No particular (trust) assumptions needed.

- No dependencies on other systems.

- **Promising point in the security↔deployability spectrum.**
  Significant step towards the vision of enforcing fairness directly in the network.

Conclusion

- FFS enforces fair bandwidth allocations directly in the network:



- Robust against:
  - Volumetric DDoS
  - Attacker creating many flows
  - Source address spoofing

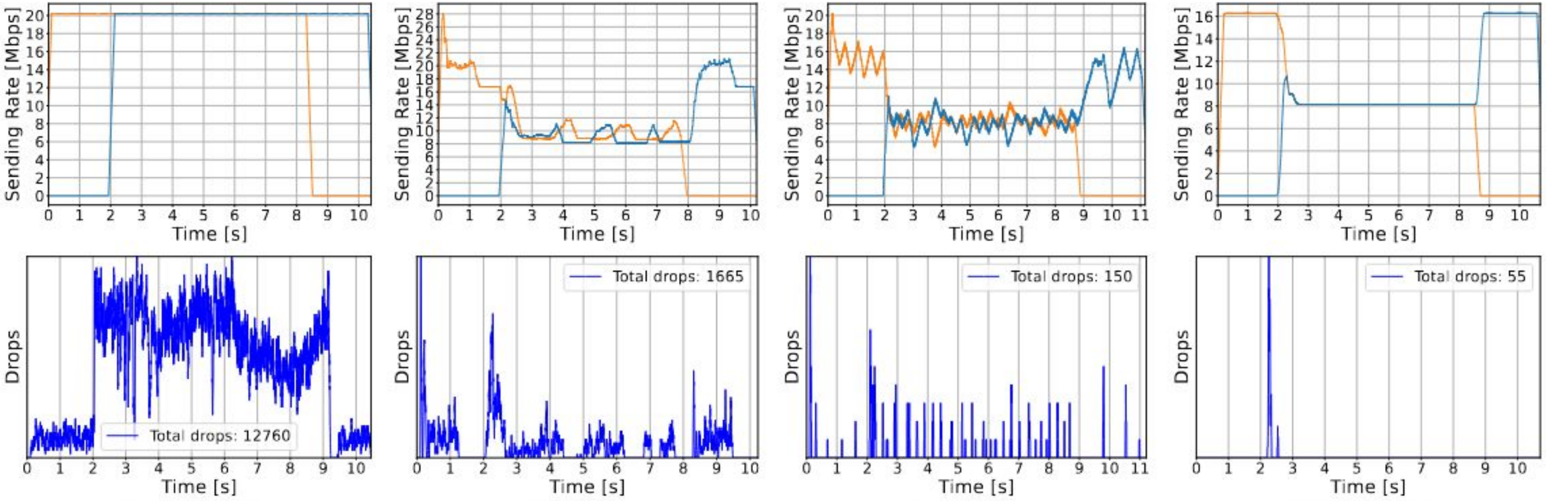
- No particular (trust) assumptions needed.

- No dependencies on other systems.

- **Promising point in the security↔deployability spectrum.**
  Significant step towards the vision of enforcing fairness directly in the network.

FFS Paper

Rate feedback evaluation



(a) MinTime (b) BBR (c) Cubic (d) MinLoss

Label normalization

Given (packet label “F”)

Normalized label

$$\mathrm{f}_s^n = \frac{\mathrm{f}_s^{\mathrm{pre}(n,s)}}{\mathrm{F}^{\mathrm{pre}(n,s)}} \times \mathbf{M}_{(i,j)}^n$$

Given (configuration)

?

How to estimate sum of fair shares without having to keep per-stream state?

Label normalization

Normalized label

Given (packet label “F”)

$$\mathrm{f}_s^n = \frac{\mathrm{f}_s^{\mathrm{pre}(n,s)}}{\mathrm{F}^{\mathrm{pre}(n,s)}} \times \mathrm{M}_{(i,j)}^n$$

Given (configuration)

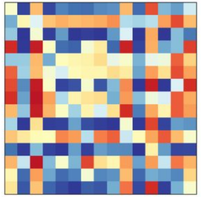
?

How to estimate sum of fair shares without having to keep per-stream state?

$$\mathrm{F}_{(i,j)}^{\mathrm{pre}(n,s)} = \sum\nolimits_{s\in \mathrm{S}_{(i,j)}^n} \mathrm{f}_s^{\mathrm{pre}(n,s)} = \underbrace{\boxed{\sum\nolimits_{s\in \mathrm{S}_{(i,j)}^n} \mathrm{r}_s^{\mathrm{pre}(n,s)}} \cdot \boxed{\frac{\mathrm{f}_s^{\mathrm{pre}(n,s)}}{\mathrm{r}_s^{\mathrm{pre}(n,s)}}}}$$

Compute the total traffic rate from packet lengths, applying a label-based weighting factor to each packet.

Evaluation: fairness definition



t1: Throughput of CCA 1

t2: Throughput of CCA 2

→

f = min(**t1**, **t2**) / max(**t1**, **t2**)

How should fairness matrices be defined?

- **Basic FFS configuration:** assign each ingress an equal share of egress capacity.

- But a node may use its fairness matrix to **prioritize certain neighbors** by assigning larger matrix entries to those with greater importance (e.g., higher-bandwidth agreements between ASes).

- **Example:** egress with a capacity of 10Gbps → can allocate matrix entries such as 1 Gbps each for 6 default-contract neighbors, and 4 Gbps for a premium-contract neighbor.

- An FFS node can **adjust its matrix anytime** without risking over-allocation, unlike bandwidth reservation systems.