iCAN Do Traffic Better

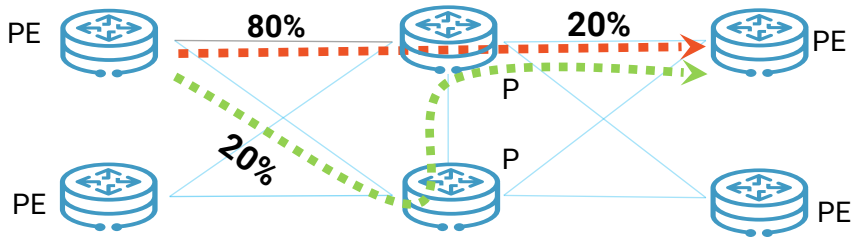
(instant Congestion Assessment Network)

draft-liu-ican-00

Bing Liu
@HotRFC, ietf105

What's the Problem?

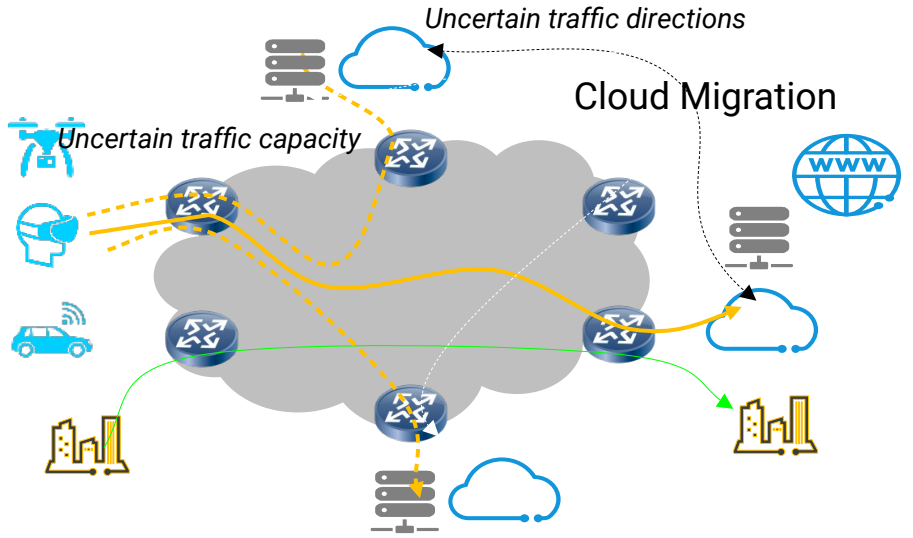
Network traffic is heavily unbalanced!



ECMP/UCMP don't really work well!

- ECMP doesn't care about the real situation of links/paths
- Devices don't know the flow size to enforce UCMP

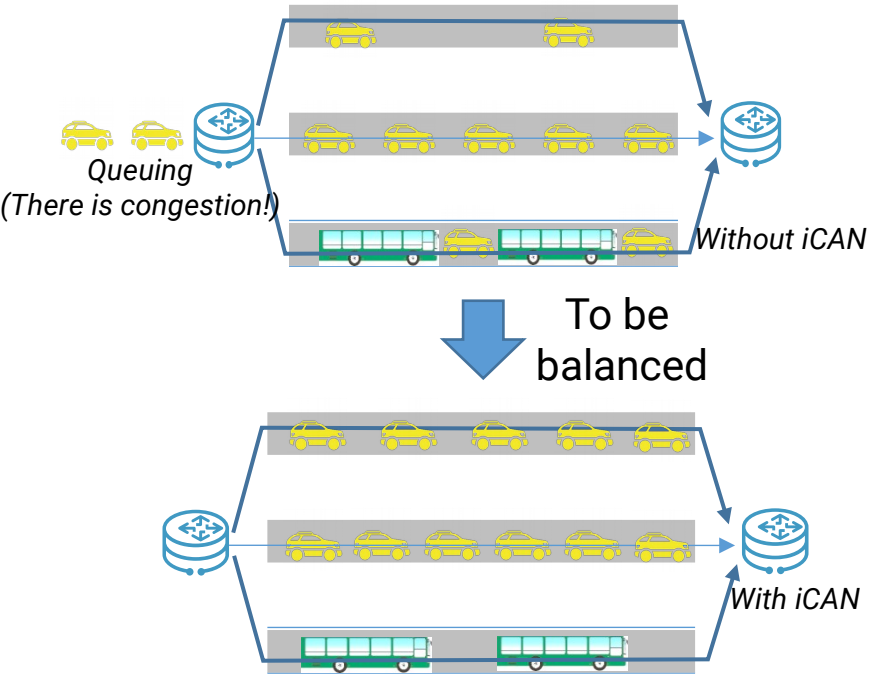
Traffic planning is more and more challenging



Traditional TE can hardly adapt to the highly dynamic traffic change.

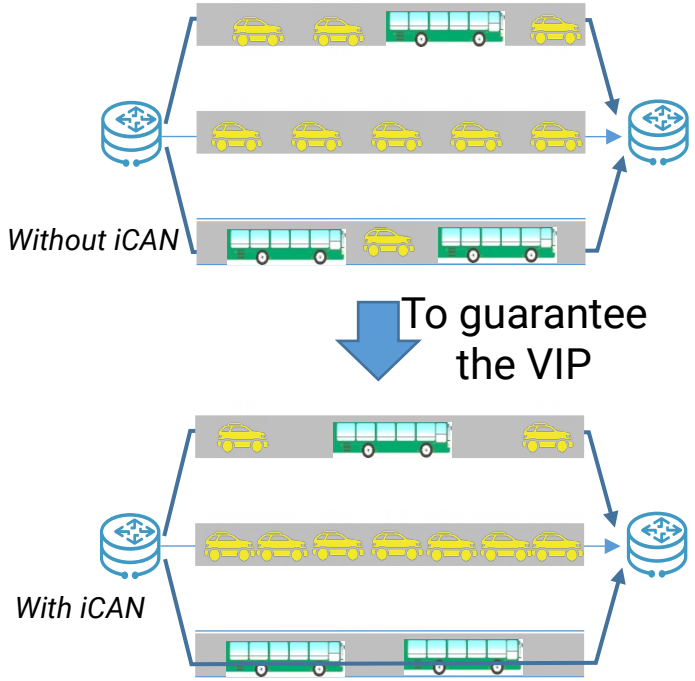
What iCAN do?

UC-1: Network Load Balancing (for real!)



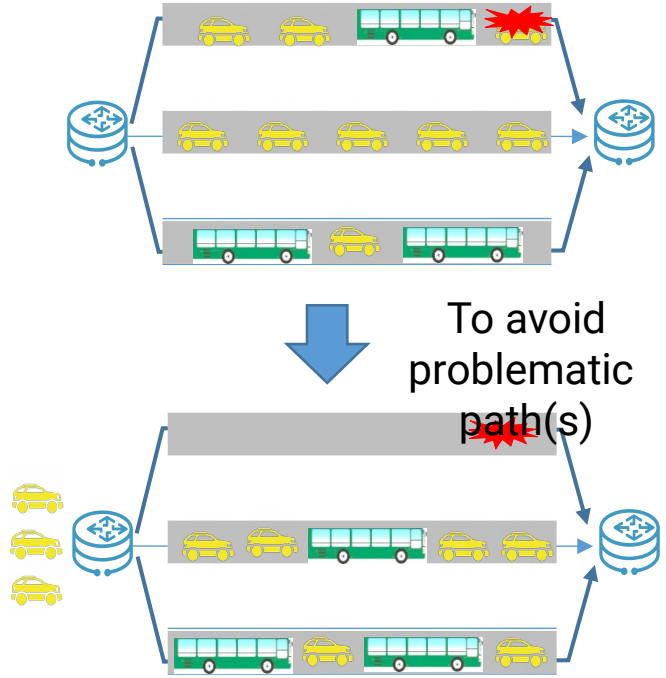
- No congestion (and probably no packet loss)
- Higher network throughput

UC-2: SLA Assassurance



- No potential SLA deterioration of high-priority services

UC-3: High Availability



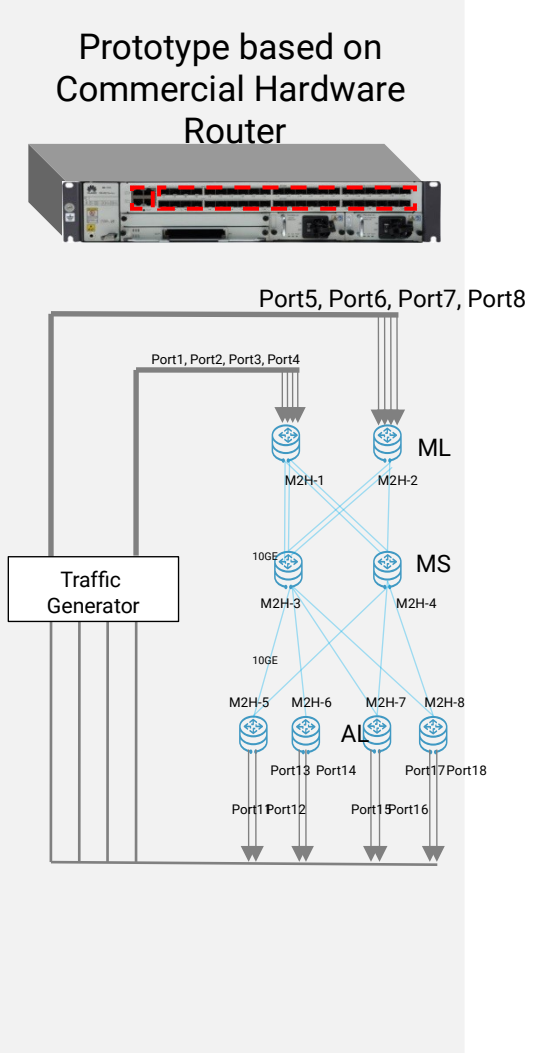
iCAN naturally supports BFD-alike functions, and can even do better:

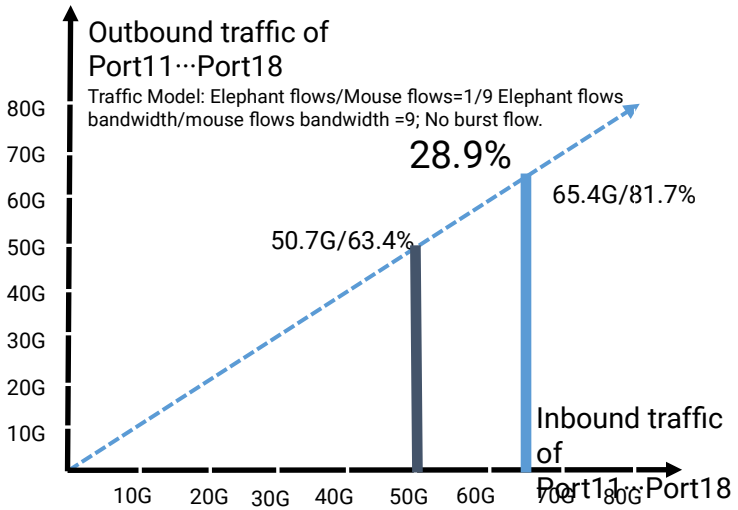
- No need for complex configurations
- Faster link failure detection
- Not only detecting path on/off, but also path quality deterioration
- Can distinguish individual paths in multi paths

✓ **For load balancing use case, we've developed a commercial hardware router based prototype, using SRv6 as the data plane.**

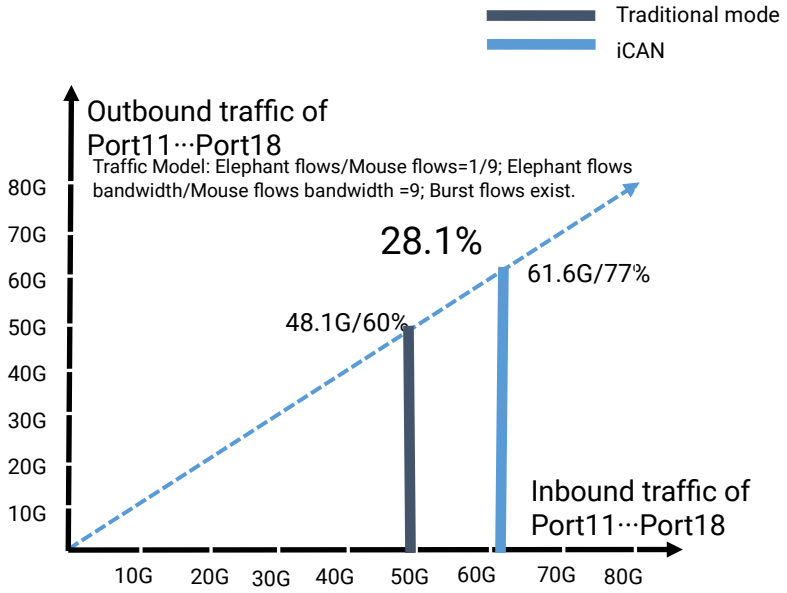
✓ **30% network throughput increment, according to the test in our lab.**

Test Result: Network throughput is increased by around 30%





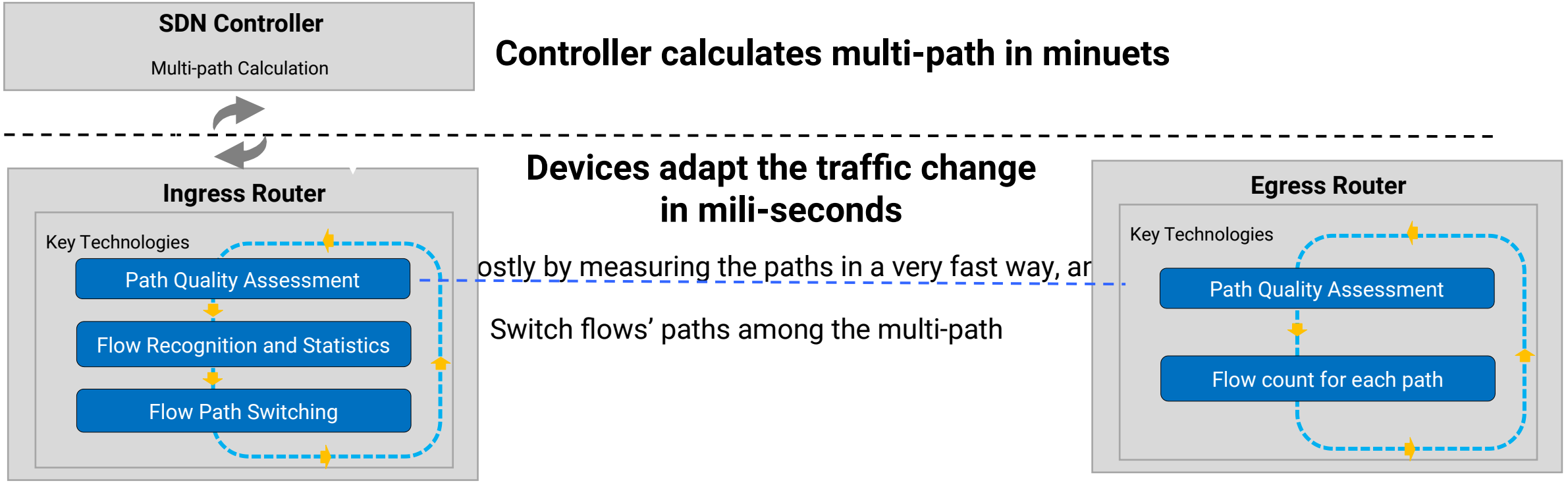
Network without sudden bursts



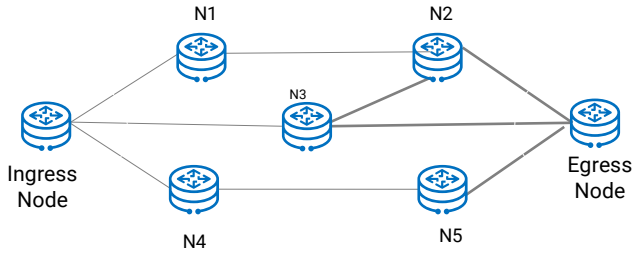
Network with bursts

- Physical capacity of the test bed is 80Gbps
- Without iCAN, it started to drop packets at 48-50Gbps
- With iCAN, it started to drop packets at 61-65Gbps, about **30% throughput increment**
- iCAN could work effectively under both burst/non-burst situations

How iCAN do it?



Intermediate Node do not need to support iCAN.



Welcome to join the discuss
in TEAS and RTGWG

or directly discuss with me (highly welcomed!)

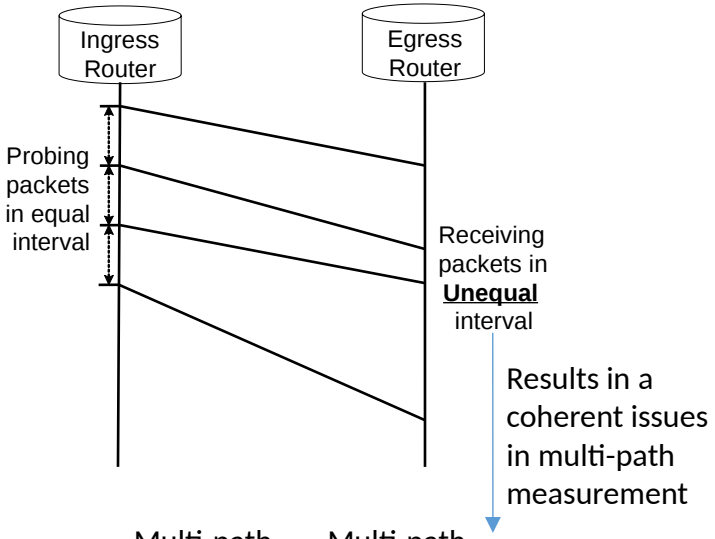
Thank you!

leo.liubing@Huawei.com
IETF105, Montreal

Backup Slides
for technical details of iCAN

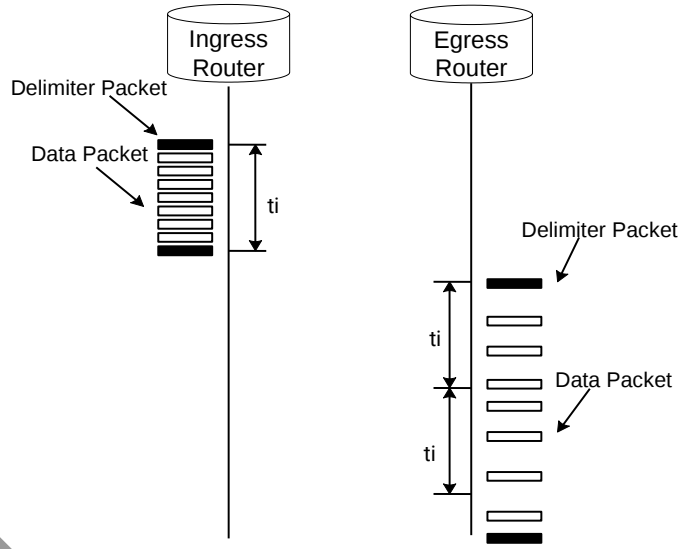
Path Quality Assessment 1/2: *coherent multi-path measurement*

Problem

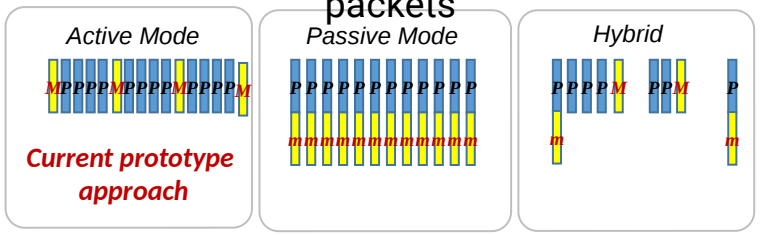


- On each path, the egress router would feedback the measurement results (m1, m2…) according to its own real interval.
- The ingress router would have to wait until the last m1/m2/m3 of the latest path come back.

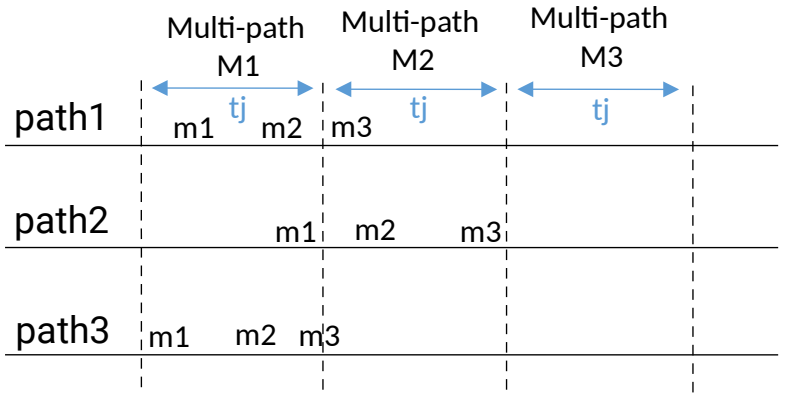
Solution



Methods of conveying delimiter packets

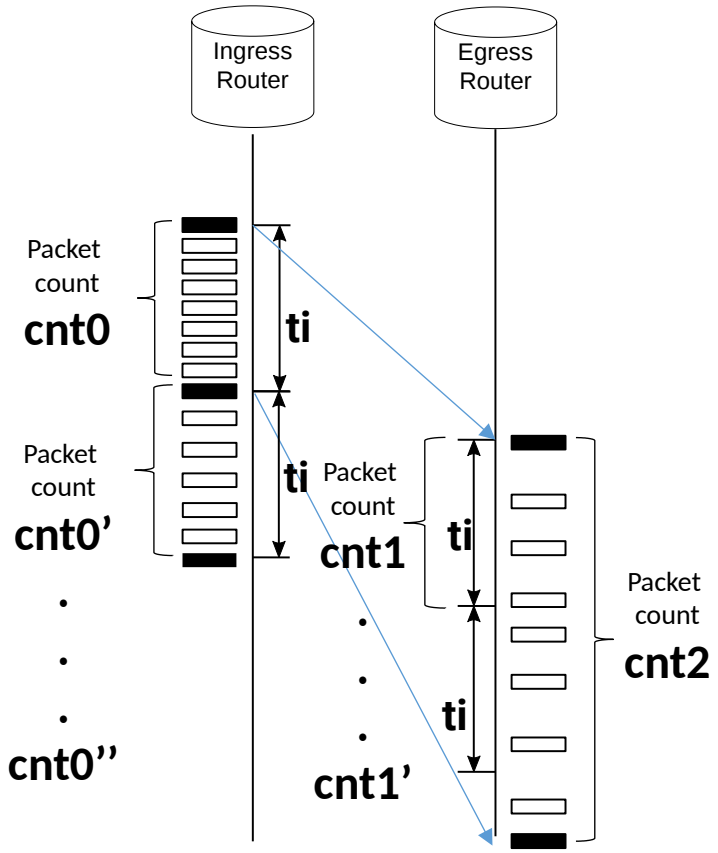


- The active probing packet acts as the delimiter packet among normal data packets. (In current prototype, the probing packet would be sent every 3.3ms, e.g. ti=3.3ms)
- Regardless of the shifting of the probing packets, the egress router would return the measurement result to the ingress router every ti internal.



- The ingress router would assess each path's congestion status every tj interval (In current prototype, tj=10ms)
- Tj should be larger enough than ti, so that every tj interval, the ingress would get at least one measurement result of each path.

Path Quality Assessment 2/2: *Path congestion calculation*



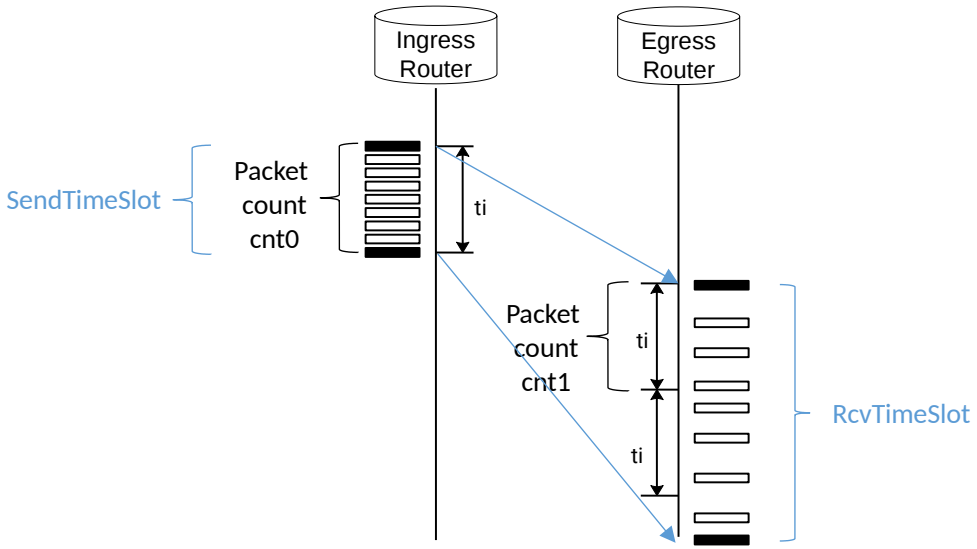
The Egress router read the cnt1 every ti interval, and send the result to the ingress; the Ingress gathers the results, and do calculation in everty ti*N interval. (e.g., ti=3.3ms, N=3)

- **TxRate** = (cnt0+cnt0'+cnt0''...) / ti*N
- **RxRate** = (cnt1+cnt1'+cnt1''...) / ti*N

PathCongestion = **RxRate** / **TxRate**

- The smallest one is the "worst" path; while the biggest one is the "best" path.
- If cnt<cnt0, it means there is packet loss happening, then the PathCongestion needs to be adjusted.

Flow path switching 1/2: basic method



$$AftrSwitch\_PathCon = \frac{(cnt1+cnt1'+cnt1''\ldots+ dRx+ dTx) / (ti*N + CurPathJitter)}{(cnt0+cnt0'+cnt0''\ldots + dRx + dTx\ ) / (ti*N)}$$

Other parameters:

- **CurPathJitter** = RcvTimeSlot-SendTimeSlot
- **dRx**: the count of flow(s) which is(are) planned to be switched into the current path
- **dTx**: the count of flow(s) which is(are) planned to be switched out of the current path

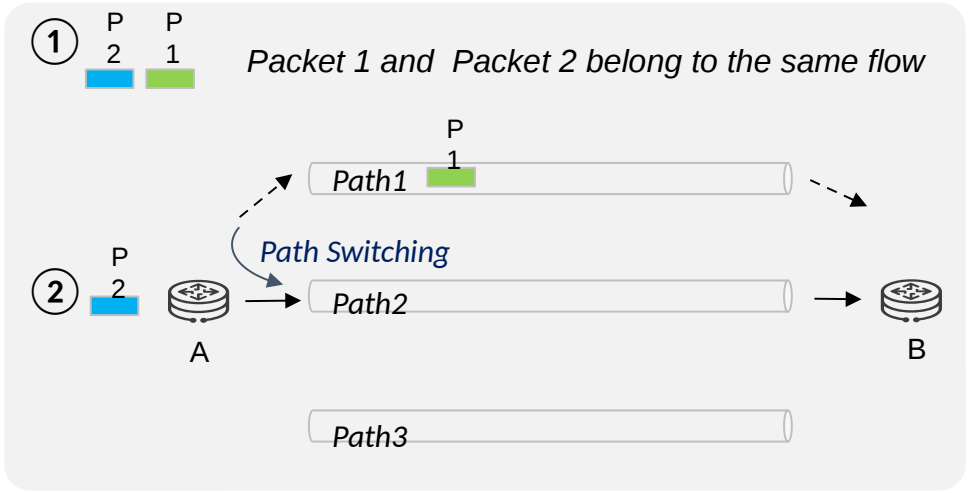
Basic Rules:

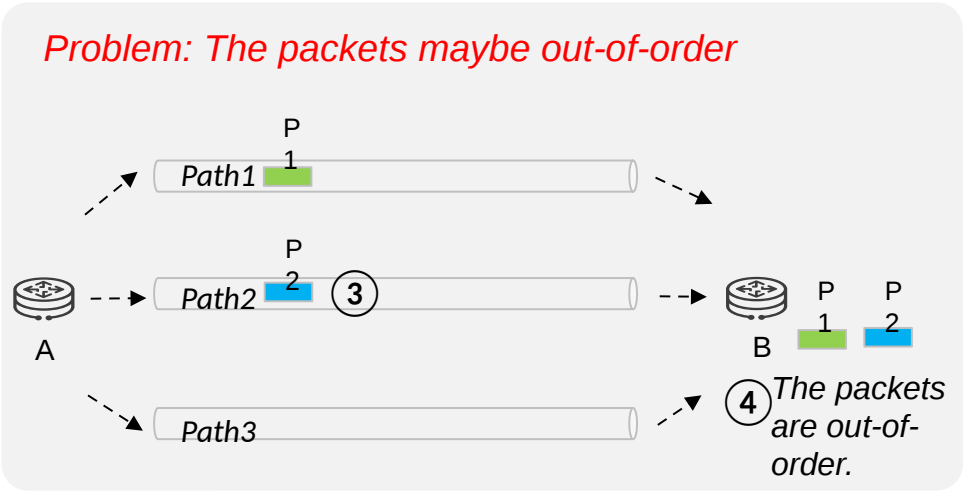
- Choose a flow in the “worst path”, and intend to switch it to the “best path”.
- Estimates the path congestion of each path, after the switching, according to the formula above. If the path congestion is more averaged than before, then the flow is considered a valid choice.
- Do the real path switch.
- Iterate above steps.

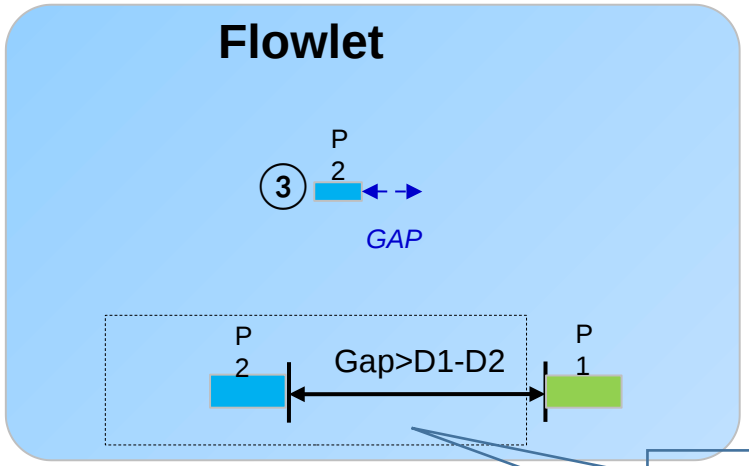
To avoid the flow switch oscillation, the flow that be switched would not be allowed to be switched again within a certain time slot (e.g. 5min).

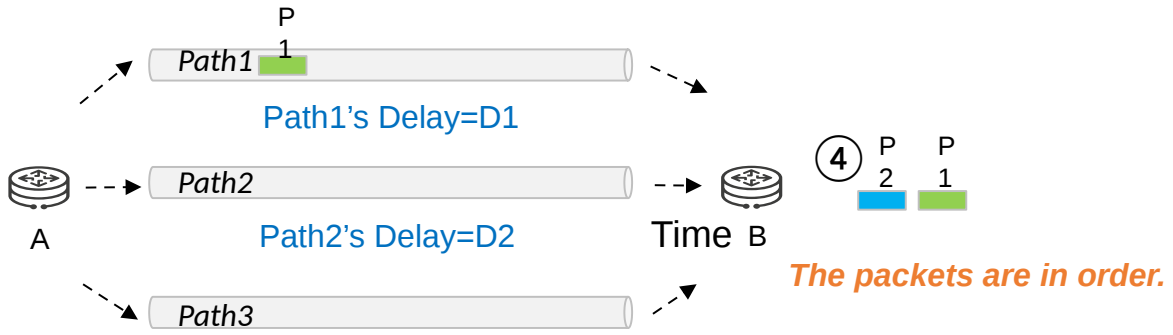
Flow path switching 2/2 : packet order assurance

Flowlet-based Scheduling ensure no packet ordering/loss issue during path switching



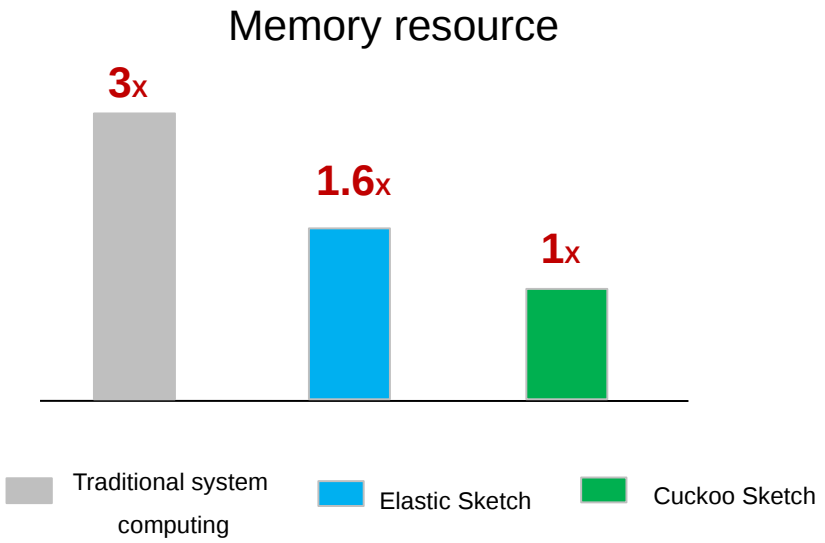






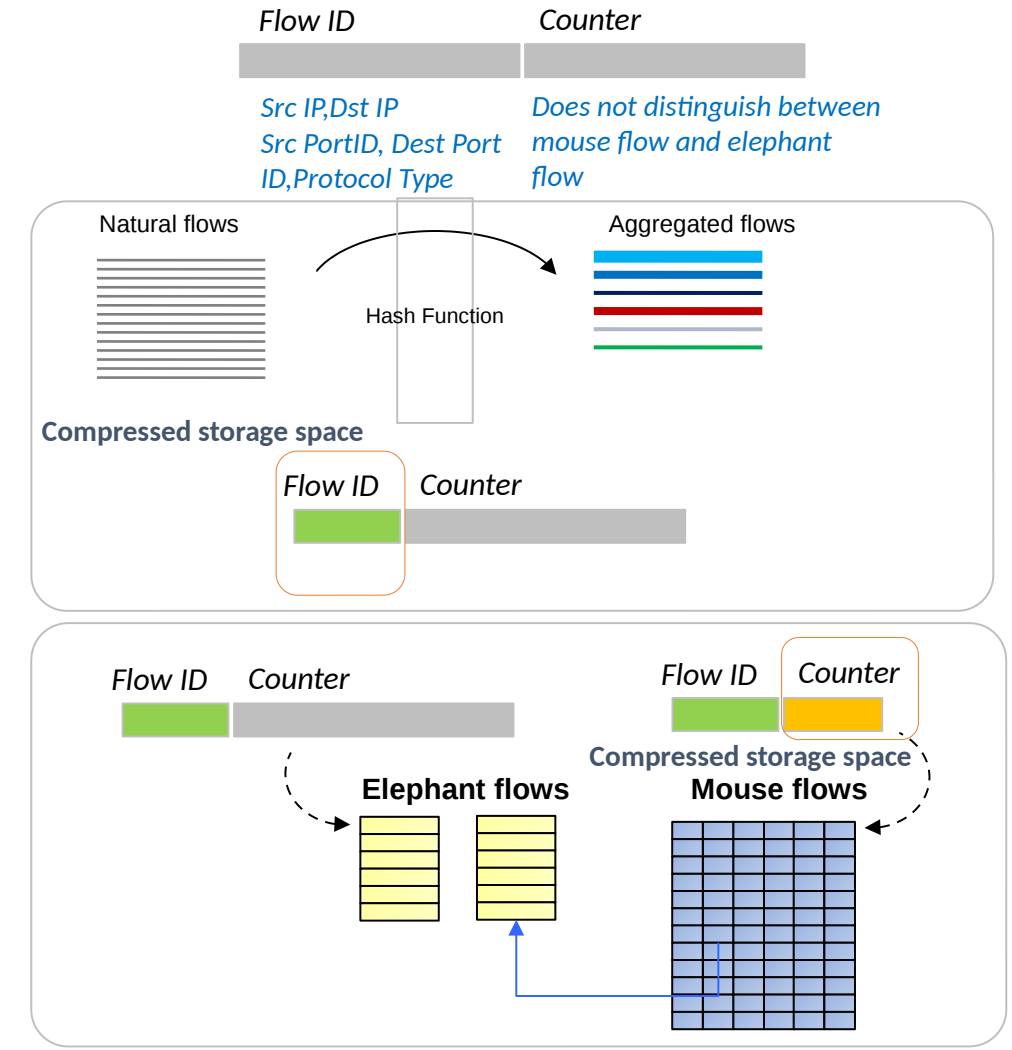
Allocating P2 a high-priority queue in the router, to avoid queuing time; and finding P2 a proper queue which has a queuing time larger than the gap time.

Flow statistics within router



The CAIDA Anonymized Internet Traces (177K streams, 2M packets, maximum stream 16K packets)		
Algorithm	**Accuracy**	**Memory resource**
Traditional system computing	100%	~1MB
Elastic Sketch (SIGCOMM 2018)	≥99%	600KB
Cuckoo Sketch	≥99%	385KB

Enhanced Cuckoo Sketch Algorithm



Deployment Scenarios: agnostic to underlay technologies/services