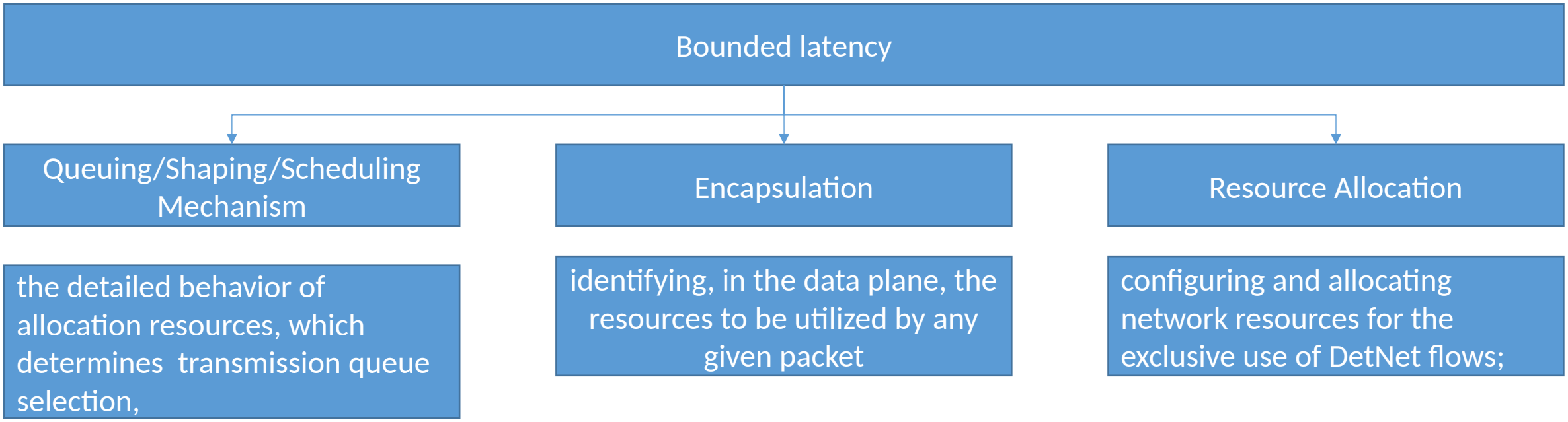
CSQF:
Cycle Specified Queuing and Forwarding

Xuesong Geng, Huawei

IETF DetNet WG Discussion
May 2023

DetNet Enhancement Data Plane Encapsulation

As defined in RFC 9320, Enhanced DetNet services of bounded latency and zero congestion loss depends upon the following 3 parts of mechanism:



Foundation of the bounded latency and determines how to calculate the latency bound

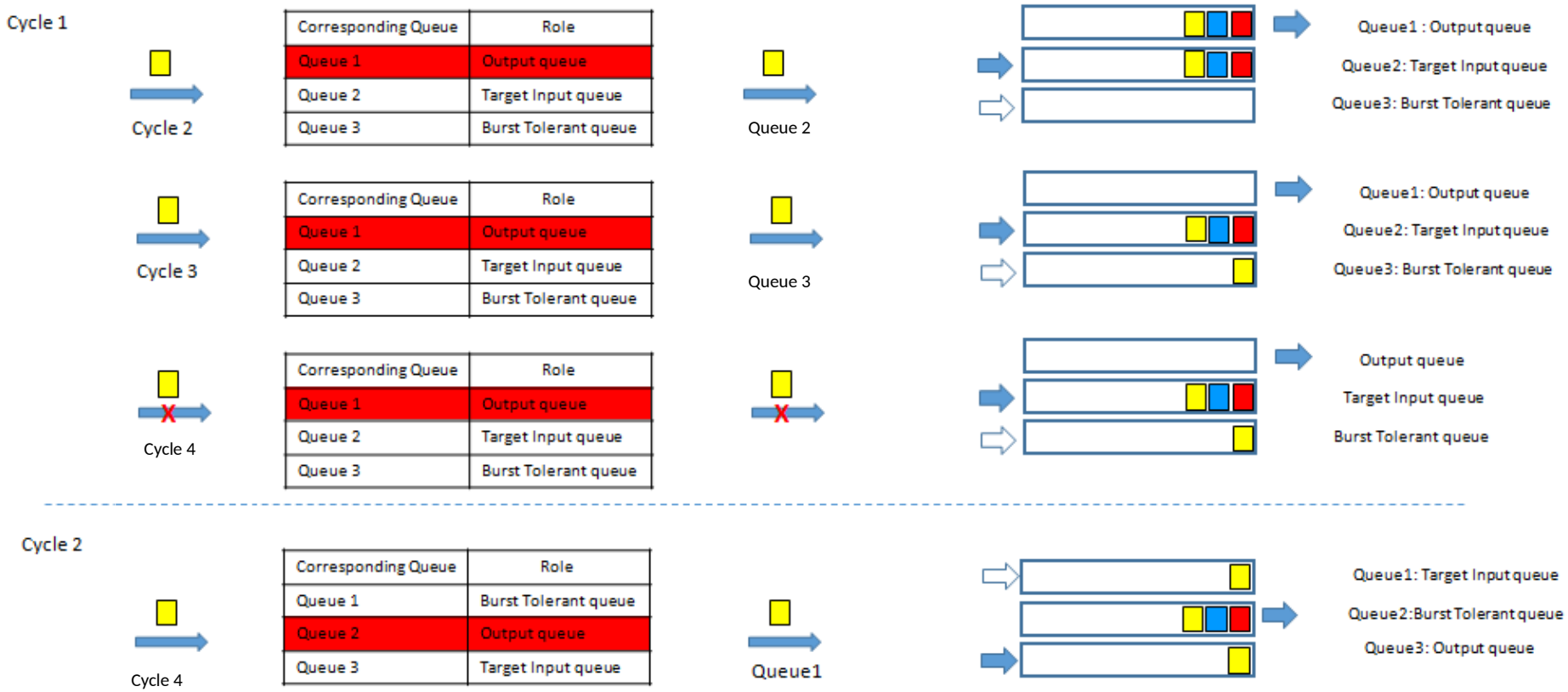
Data Plane encapsulation in the packet to support the underlying mechanism

Control plane extension to avoid resource confliction

Queuing Mechanism– CSQF

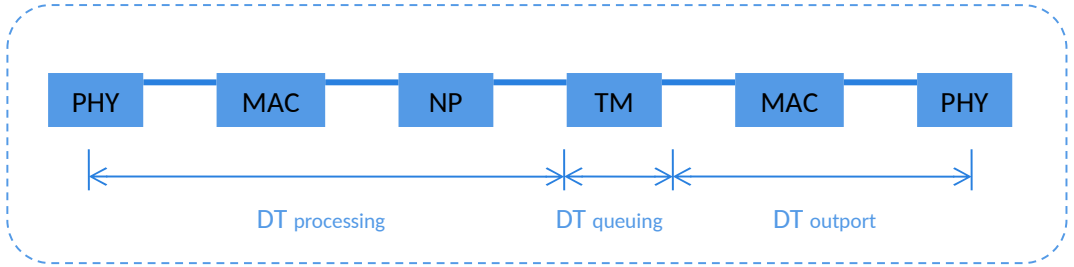
Output queue: Cycle 1

Corresponding Queue	Role	Target cycle number	Next Role
Queue 1	Burst Tolerant queue	Cycle 3	Target Input queue
Queue 2	Output queue	Cycle 1	Burst Tolerant queue
Queue 3	Target Input queue	Cycle 2	Output queue



Queuing Mechanism– CSQF

Queuing/Shaping/Scheduling Mechanism

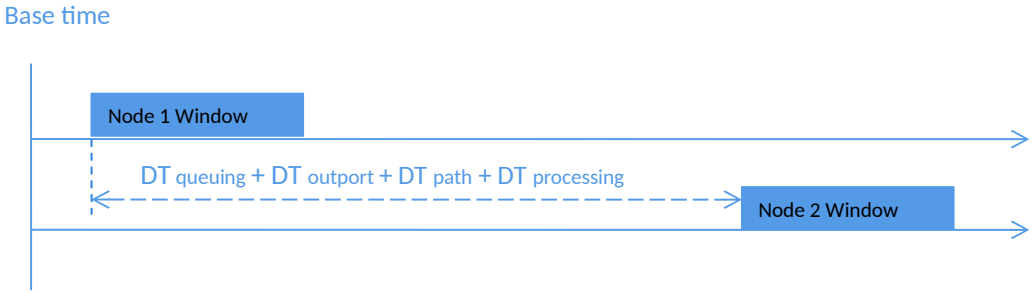


Delay Time per hop = DT queuing + DT outport + DT path + DT processing

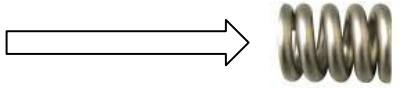
Delay Time per hop = DT queuing + Δ DT

Δ DT is changing all the time, which is the jitter of every hop

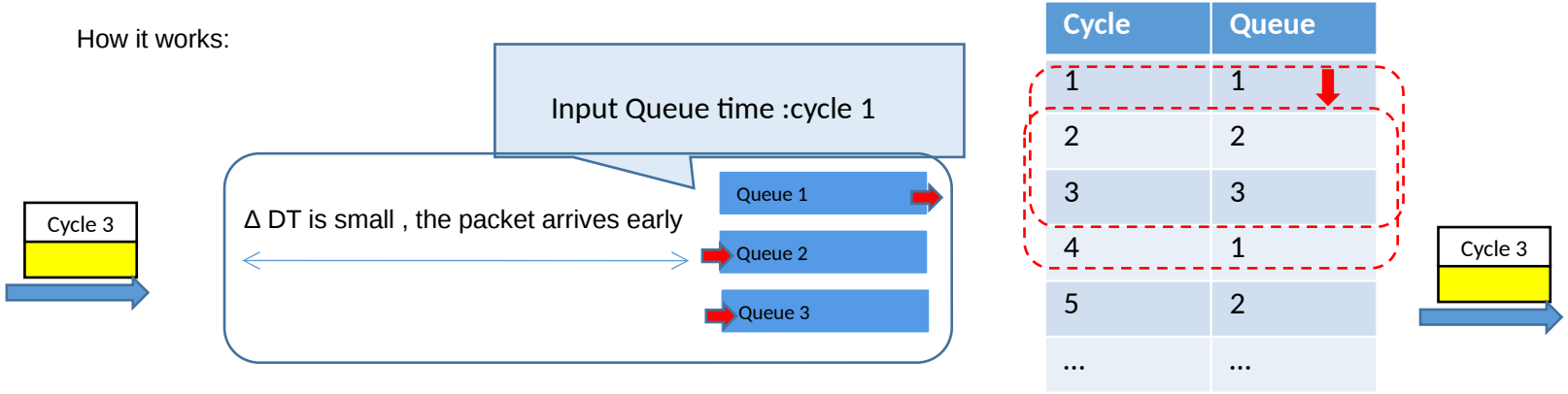
CSQF can guarantee that the **delay time per hop is constant** by changing the queuing delay time with the Δ DT:



When ΔDT is small ⇨  Queuing Delay is large

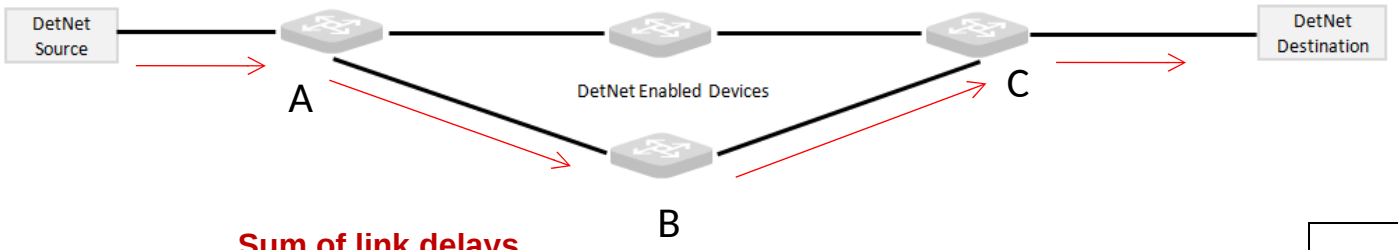
When ΔDT is large ⇨  Queuing Delay is small

How it works:



Considered scheduler for DetNet: CSQF

Path Planning



Time is divided in cycles

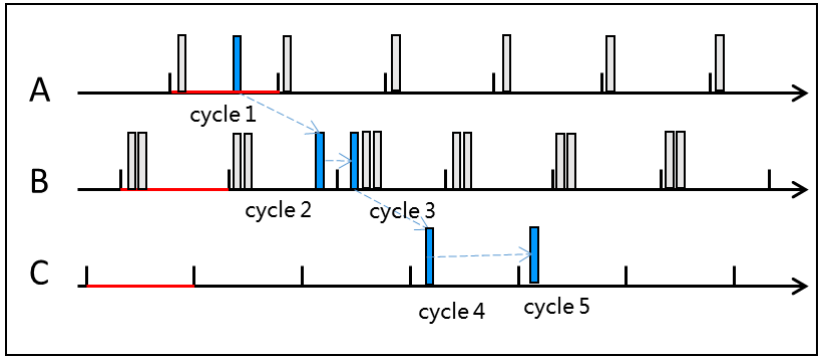
Cycle capacity = Link capacity * Cycle duration

50% of the link capacity reserved for best effort traffic

Sum of link delays

E2E Latency= (8+6+7+9) + 3 * 2 * cycle = 90us <= 100us

Sum of node delays



Time Planning (Node Status Update)

Node ID	MaxPackets	Cycle Offset	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	...
A	20	8us	2	1	2	1	2	1	
B	20	4us	2	2	3	2	3	2	
C	20	1us	0	0	0	0	1	0	

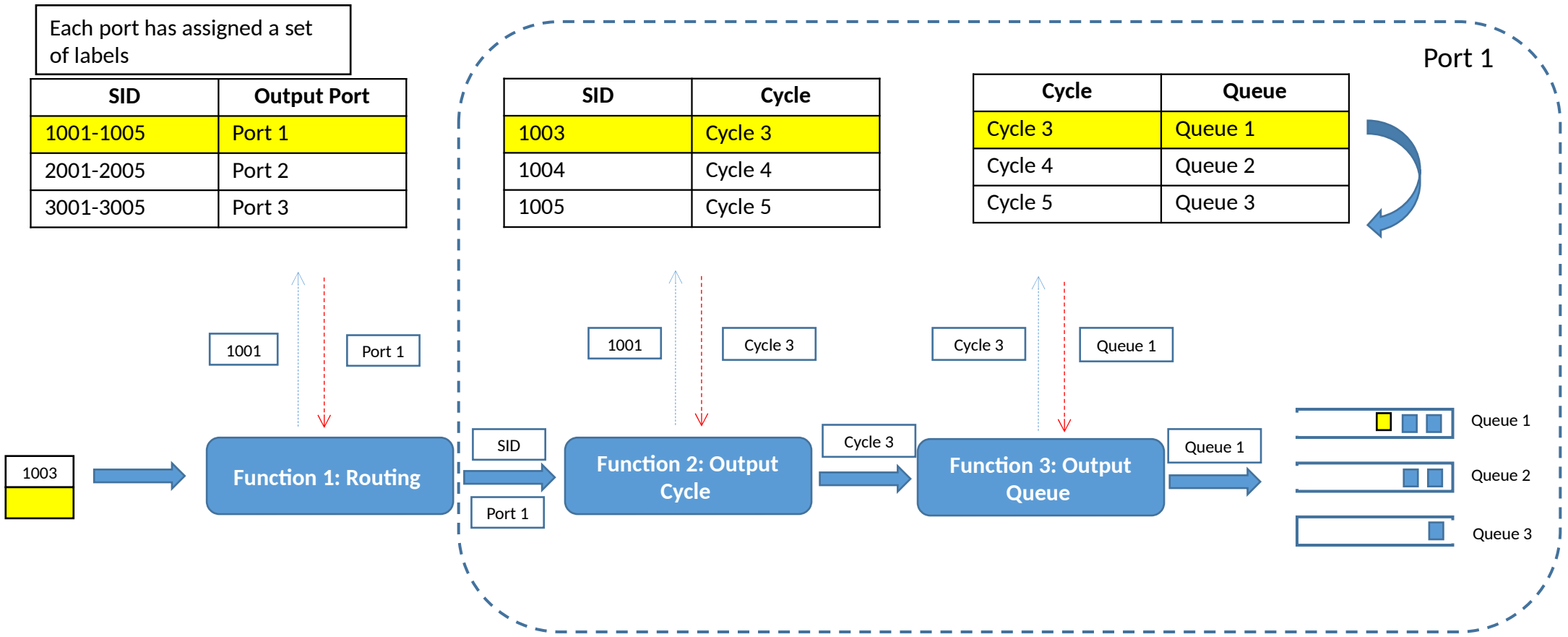
Adjacency	Max Latency without queuing latency
Source-A	8us
A-B	6us
B-C	7us
C-Destination	9us

Queuing latency = 10 us

How it works in the device

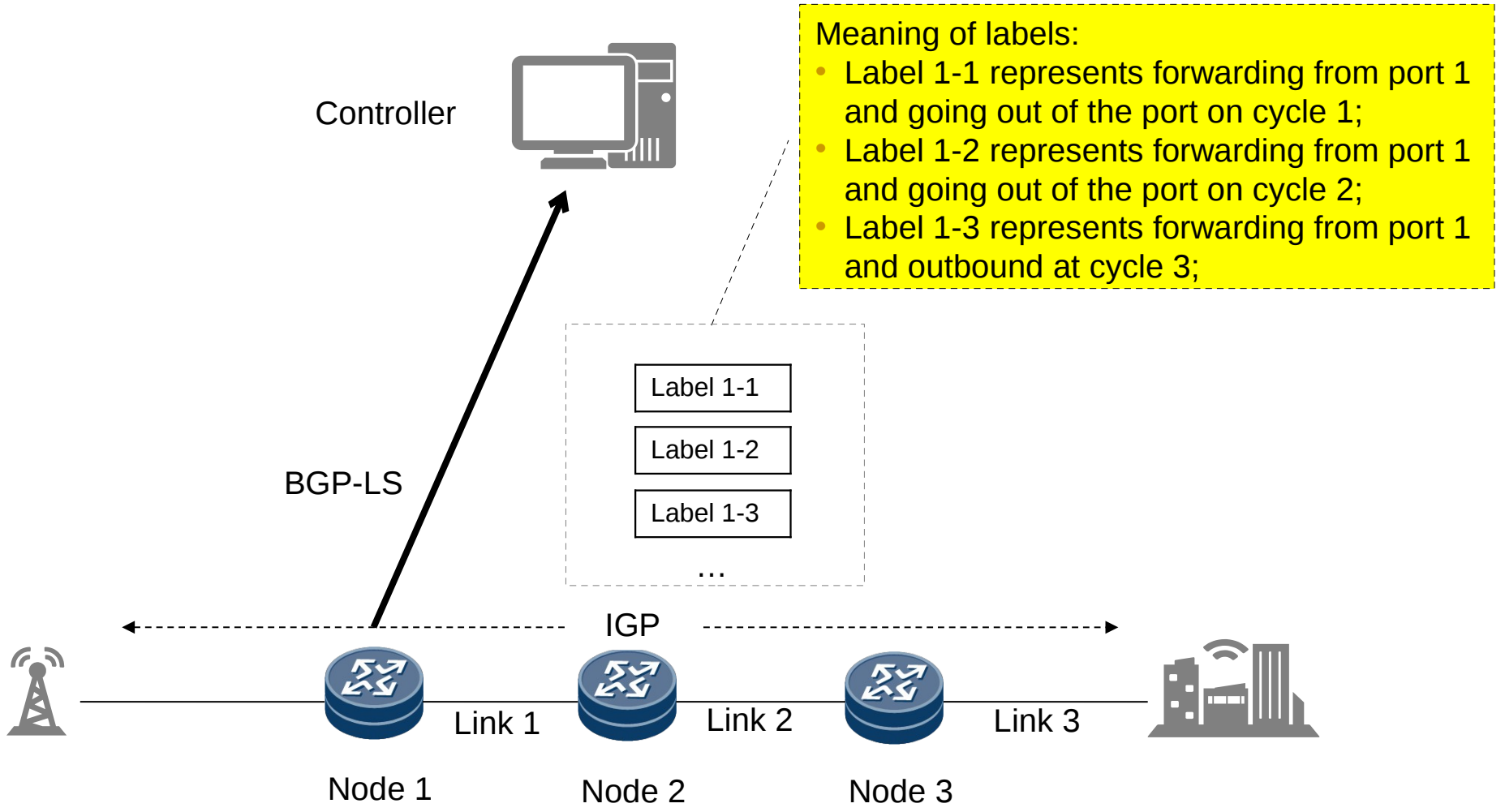
Each port has assigned a set of labels

SID	Output Port
1001-1005	Port 1
2001-2005	Port 2
3001-3005	Port 3

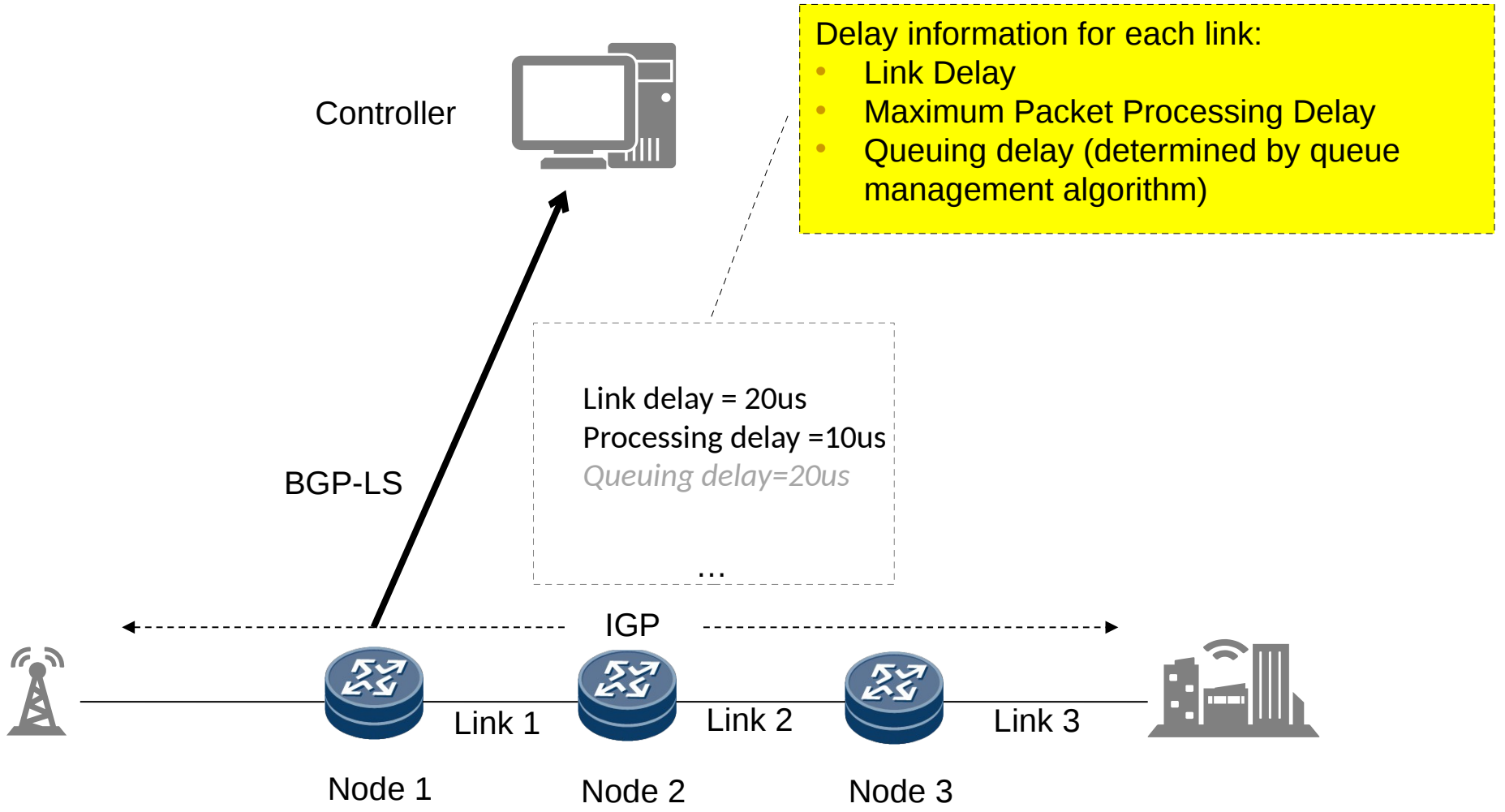


Given a label, the incoming packet can be routed over a given link in a given cycle

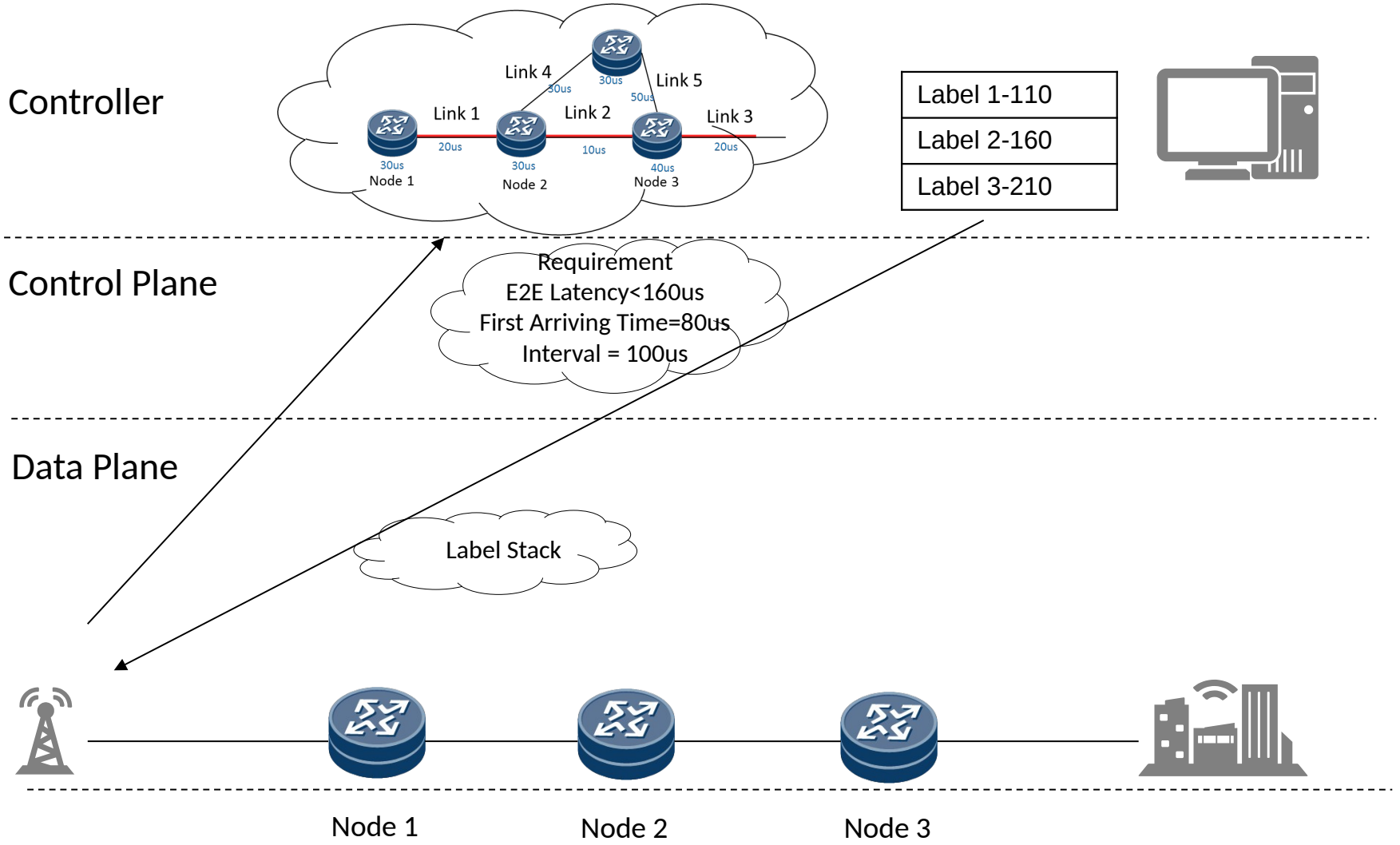
Step 1: Controller Topology Discovery and Information Collection



Step 1: Controller Topology Discovery and Information Collection

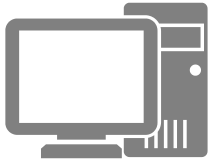


Step 2: Controller Path Calculation and Allocation



Step 3: Packet Forwarding

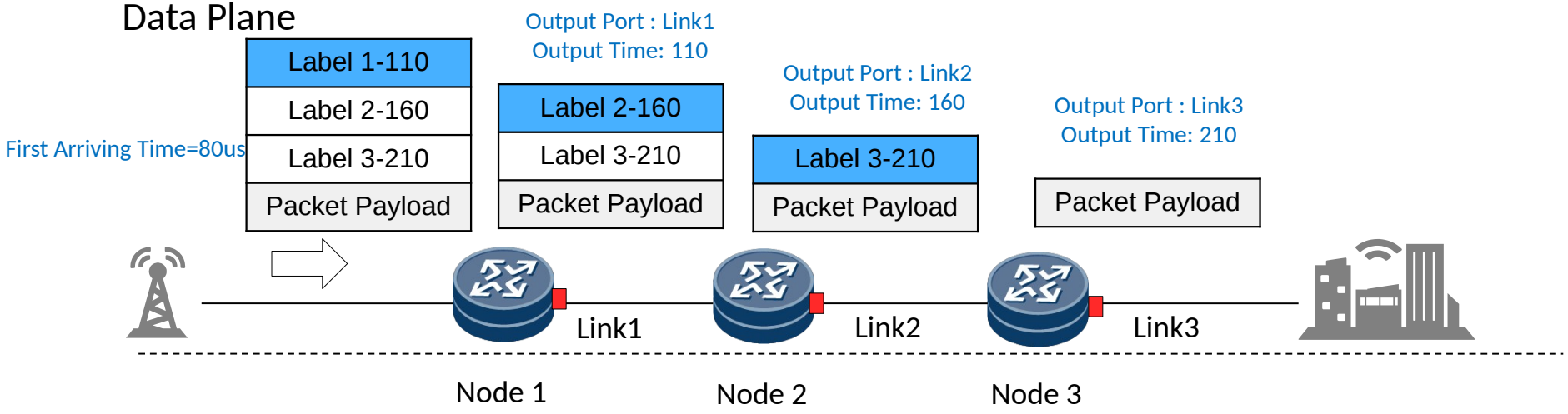
Controller



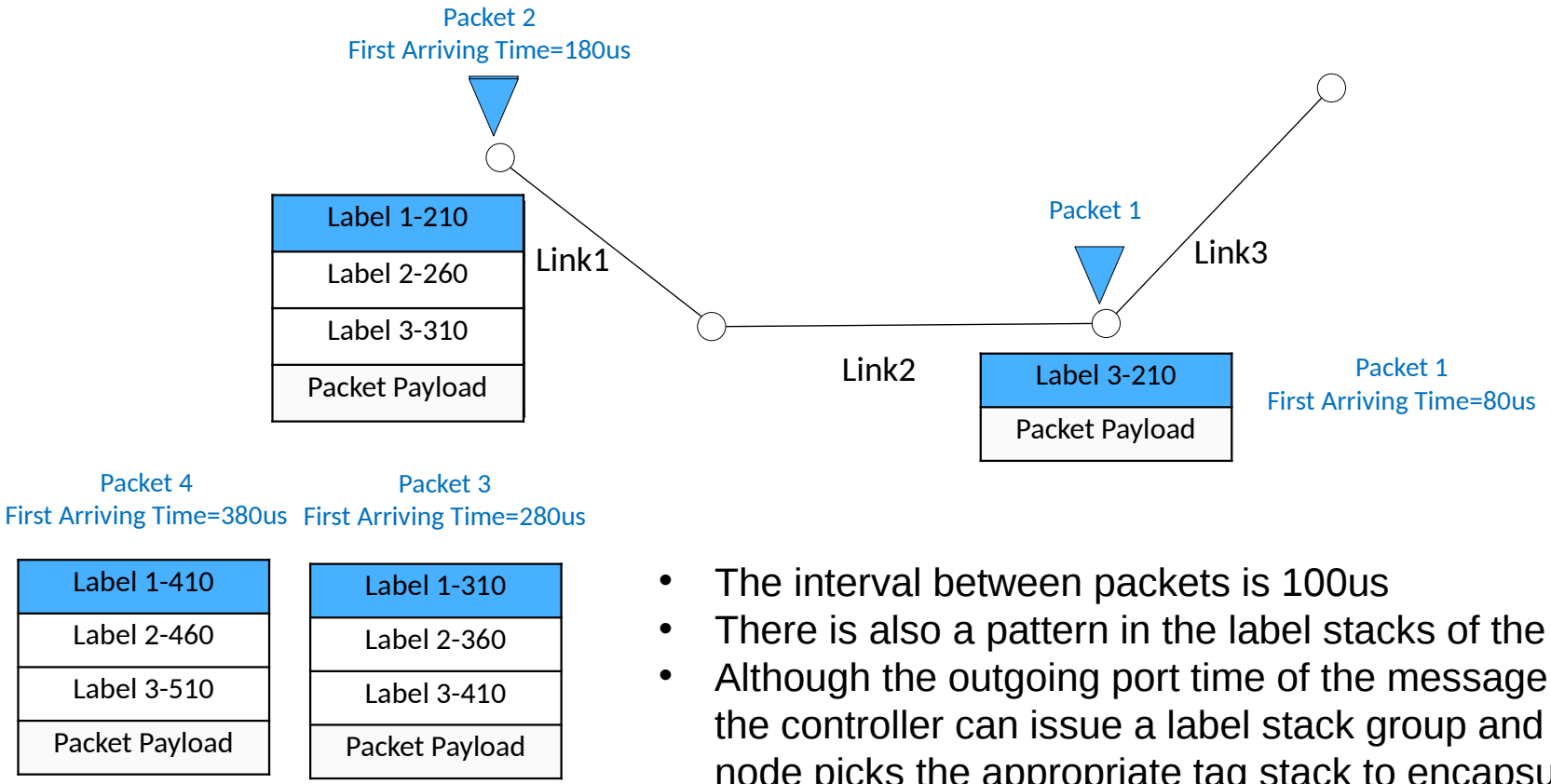
Control Plane

Maximum E2E Latency=150us
Maximum E2E jitter = 2*cycle=20us

Data Plane



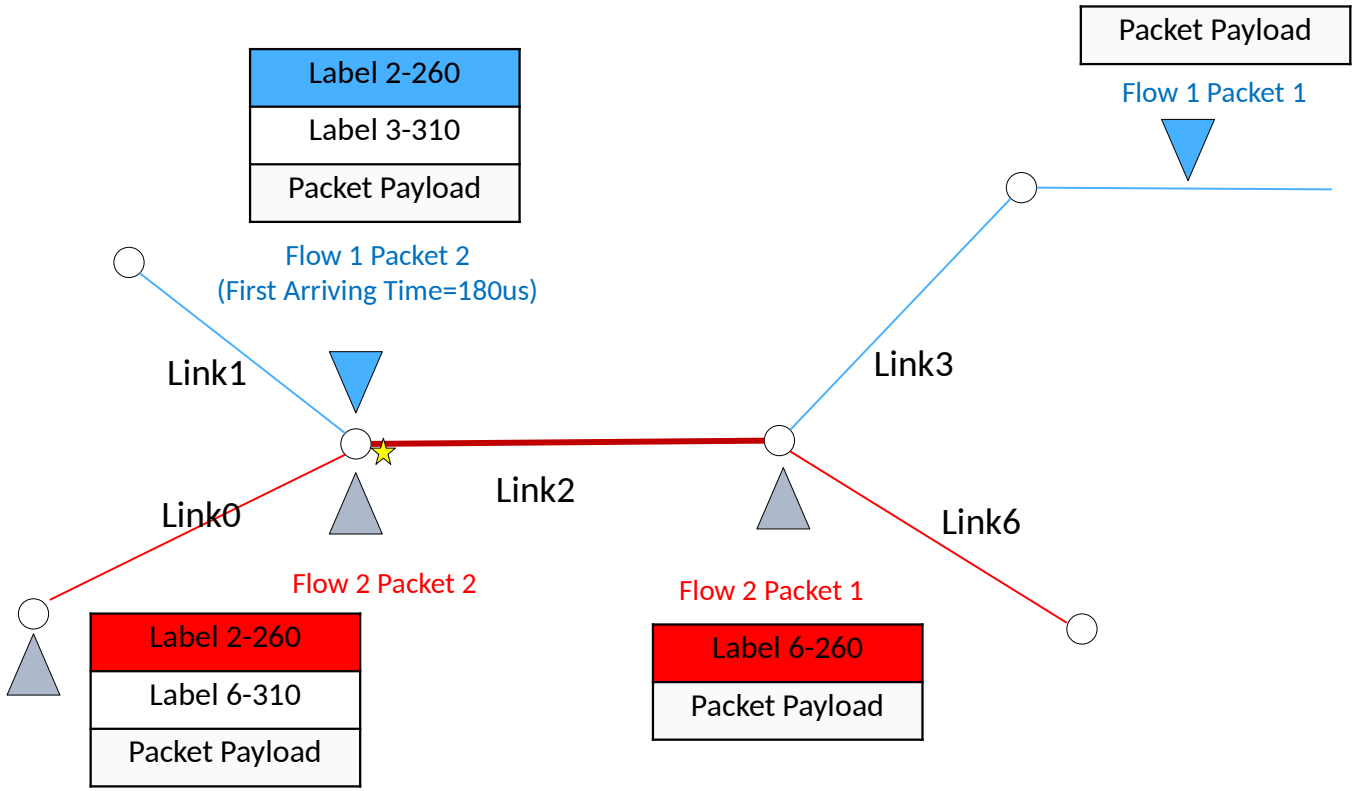
Step 4: Different Packets in Same DetNet Flow



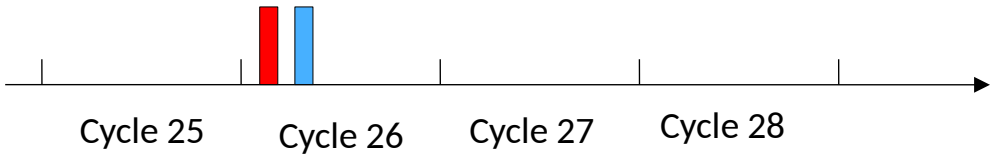
- The interval between packets is 100us
- There is also a pattern in the label stacks of the packets
- Although the outgoing port time of the message is different, the controller can issue a label stack group and the edge node picks the appropriate tag stack to encapsulate the packet

Step 4: Different DetNet Flows in Network

- Interval (Flow 1)=100us;
- Interval(Flow2)=50us

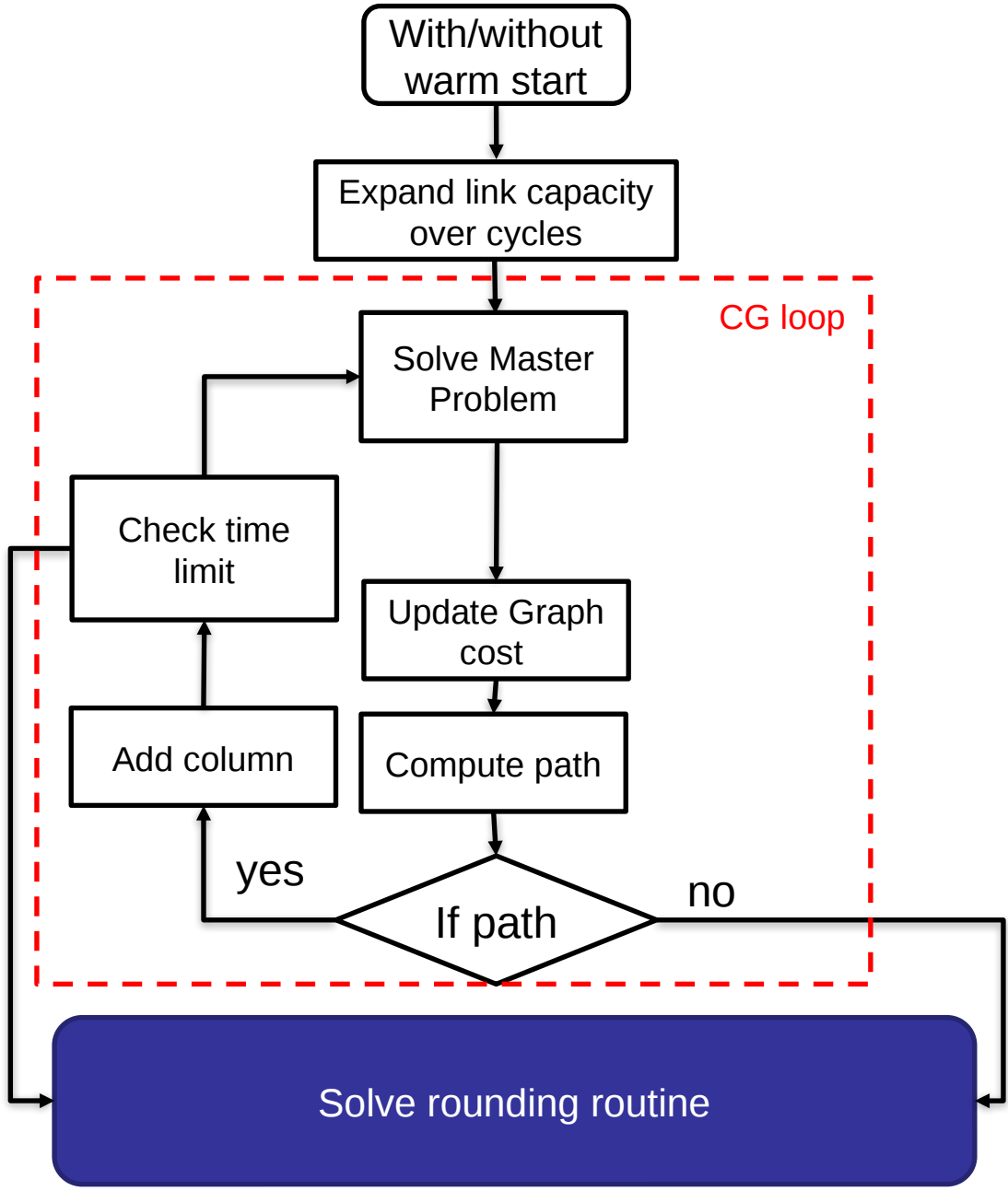


- Flow1 and Flow2, with different frequencies and different paths, share bandwidth at Link2 (same label);
- Intermediate nodes do not recognize flows and forward them only according to the label;
- Control the allocation of labels to ensure that the same cycle has sufficient resources and does not cause flow conflicts or congestion
- In the case of congestion, some packets may be unepxectedly delayed to other cycles introducing jitter



Offline planning

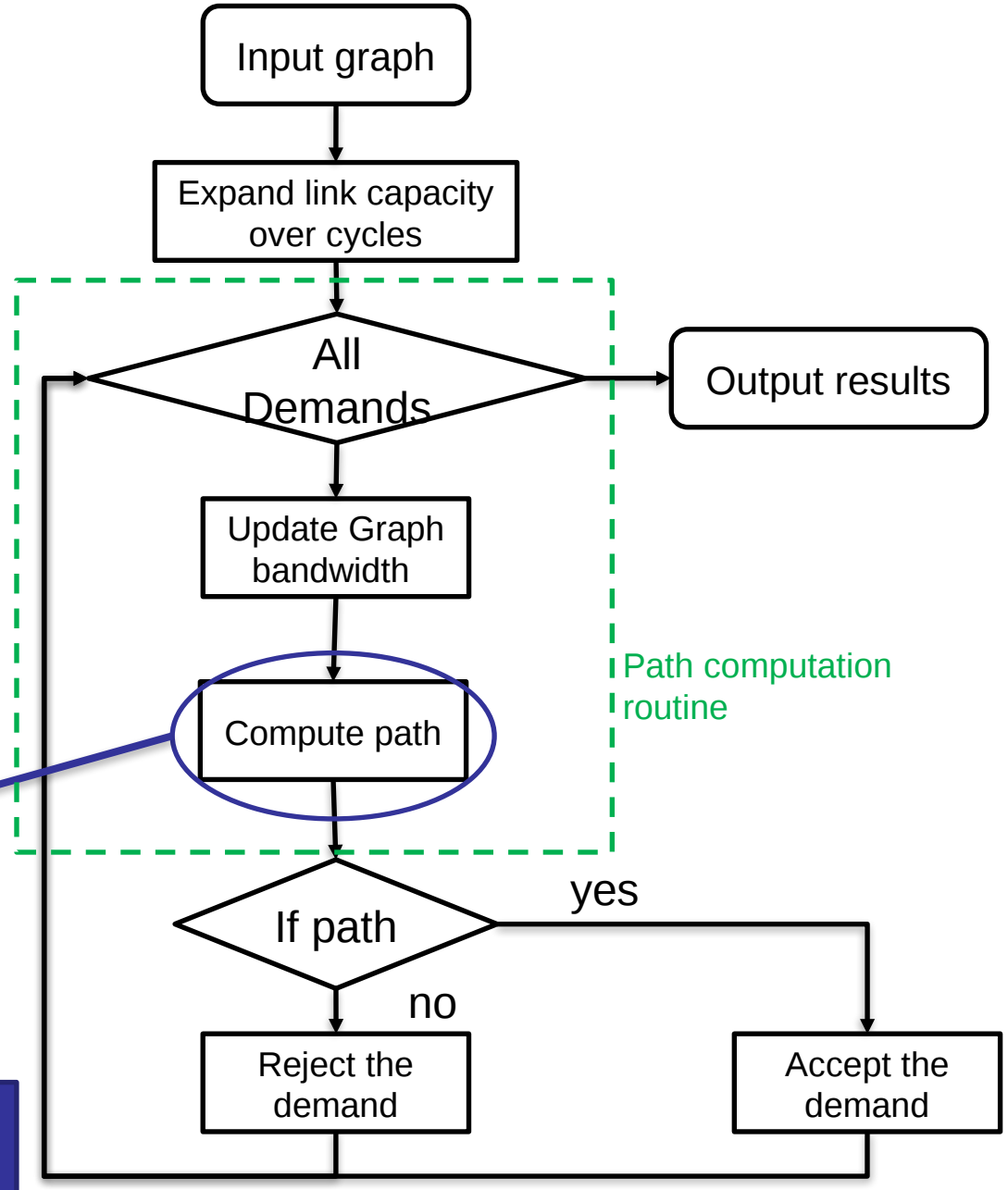
- **Based on Column Generation (CG) with Rounding**
- **Warm start** can be provide an initial feasible solution (e.g. via a greedy heuristic)
- **Graph expansion** to take into account the presence of cycles with different capacity
- **CG loop** to dynamically add only the relevant paths to the solution space
  - Compute linear relaxation
  - Solve pricing problem
  - Stop if there are no new columns to be added or the given time limit expires
  - It provides an almost optimal linearized solution (need for an integer one)
- **Rounding routine:** the output of the CG is a linear fractional solution that needs to be rounded to an integer one
  - Driven by the output of CG loop
  - Computed either optimally or quickly via a Randomized Rounding (RR) routine



For more details on advanced implementations, refer to Krolikowski et al., “Joint routing and scheduling for large-scale deterministic IP networks,” Computer Communications 2021

Online traffic acceptance

- **Graph expansion**: for each link we consider a capacity for each cycle
- **Path computation routine:** for each demand *d*
  - Step 1: compute a path for the demand *d* according the remaining bandwidth on each cycle
  - Step 2: if there exists a path, accept the demand, otherwise reject it
- **Install** the demand in the network and **update the residual capacity**
  - On each link used by the path
  - On each cycle used in the links of the path



To find a path we use a depth first search algorithm. When we find the first path respecting the delay and the bandwidth on each cycle then we stop the search.

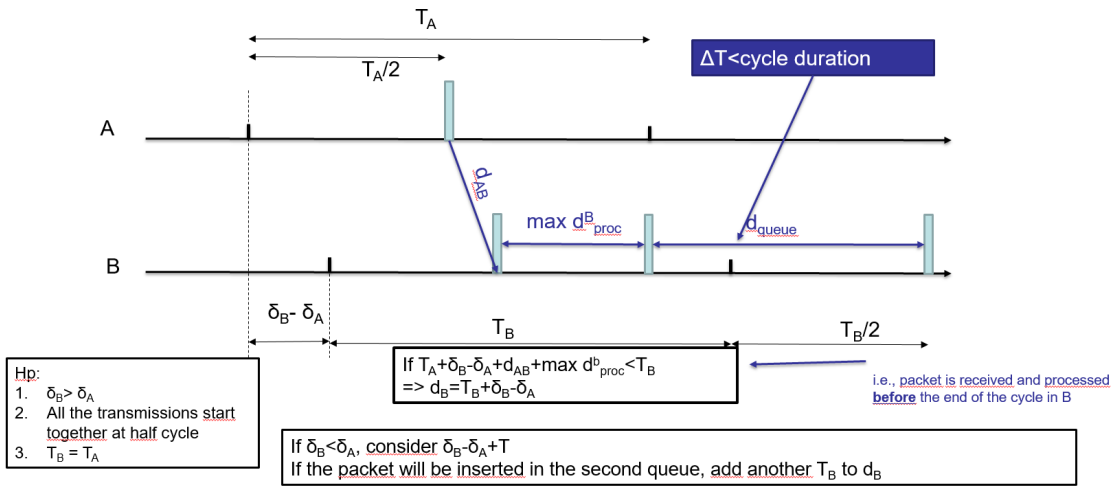
Proper cycle usage is implicitly considered in the exploration of the tree

Q&A

- Do all three drafts define a common queueing/scheduling mechanism?
- Yes the underlying mechanism is the same.

- What is the difference?
- The mapping relationship is calculated by the controller rather than maintained in the device.
- The time slot can by calculated flexibly based on the reservation status.

Example of transition



Thanks