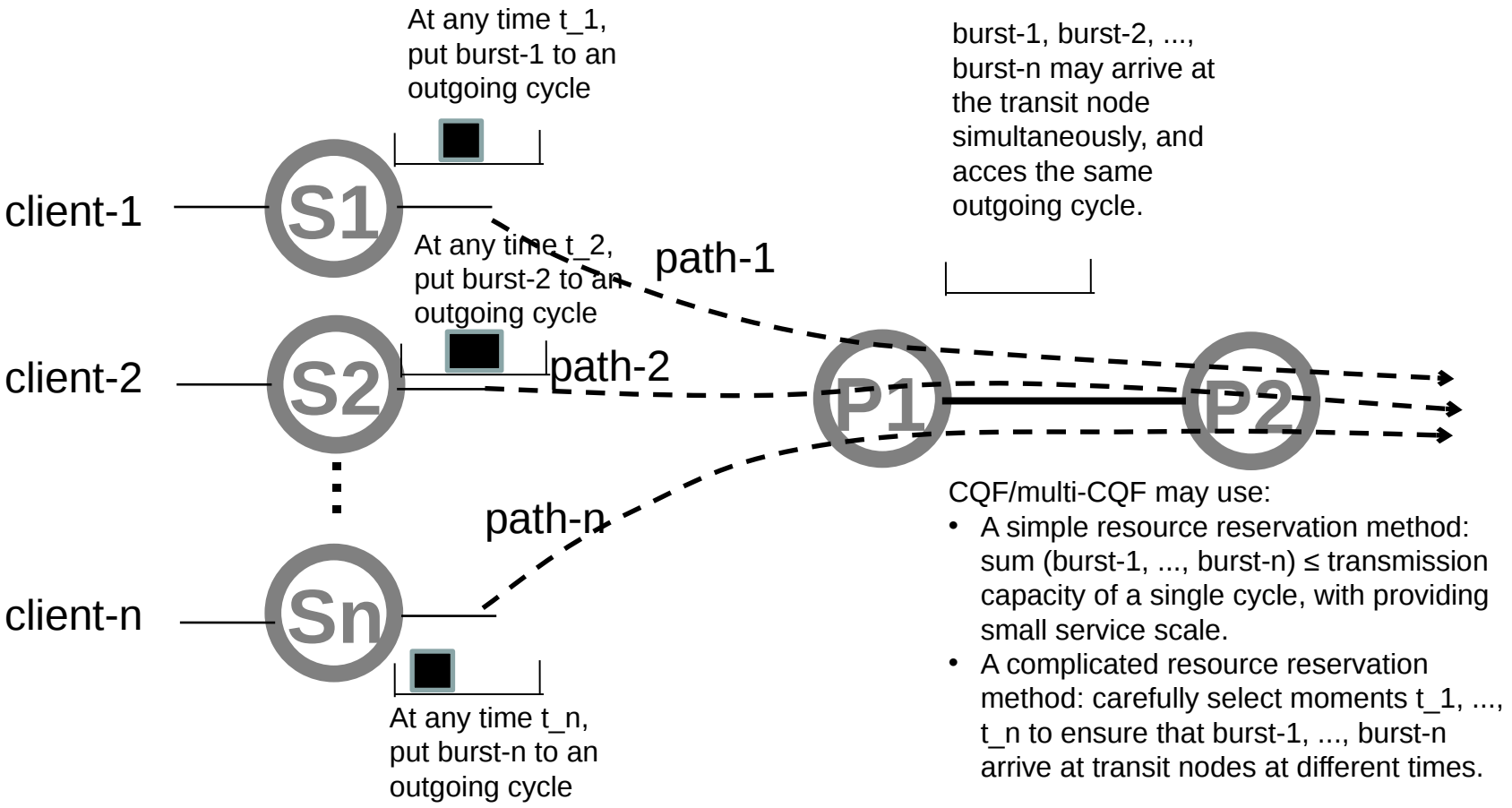
Timeslot Queueing and Forwarding (TQF)

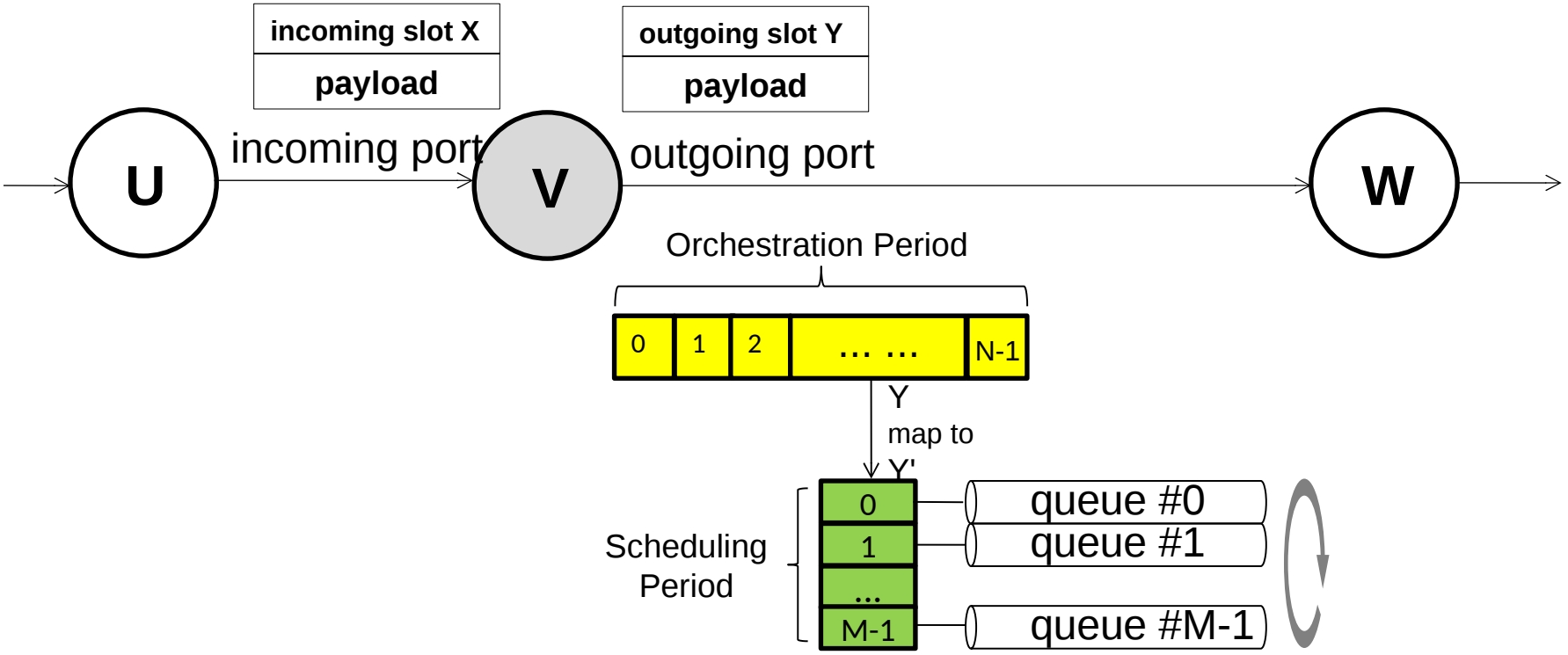
Shaofu Peng	ZTE
Peng Liu	China Mobile
Kashinath. Basu	Oxford Brookes University
Aihua Liu	ZTE
Dong Yang	Beijing Jiaotong University

Why introuce TQF ?

- This proposal is inspired by CQF/multi-CQF and provides a more flexible timeslot based resource reservation and scheduling method, to **improve the service scale that the network can support**.
  - existing CQF/multi-CQF



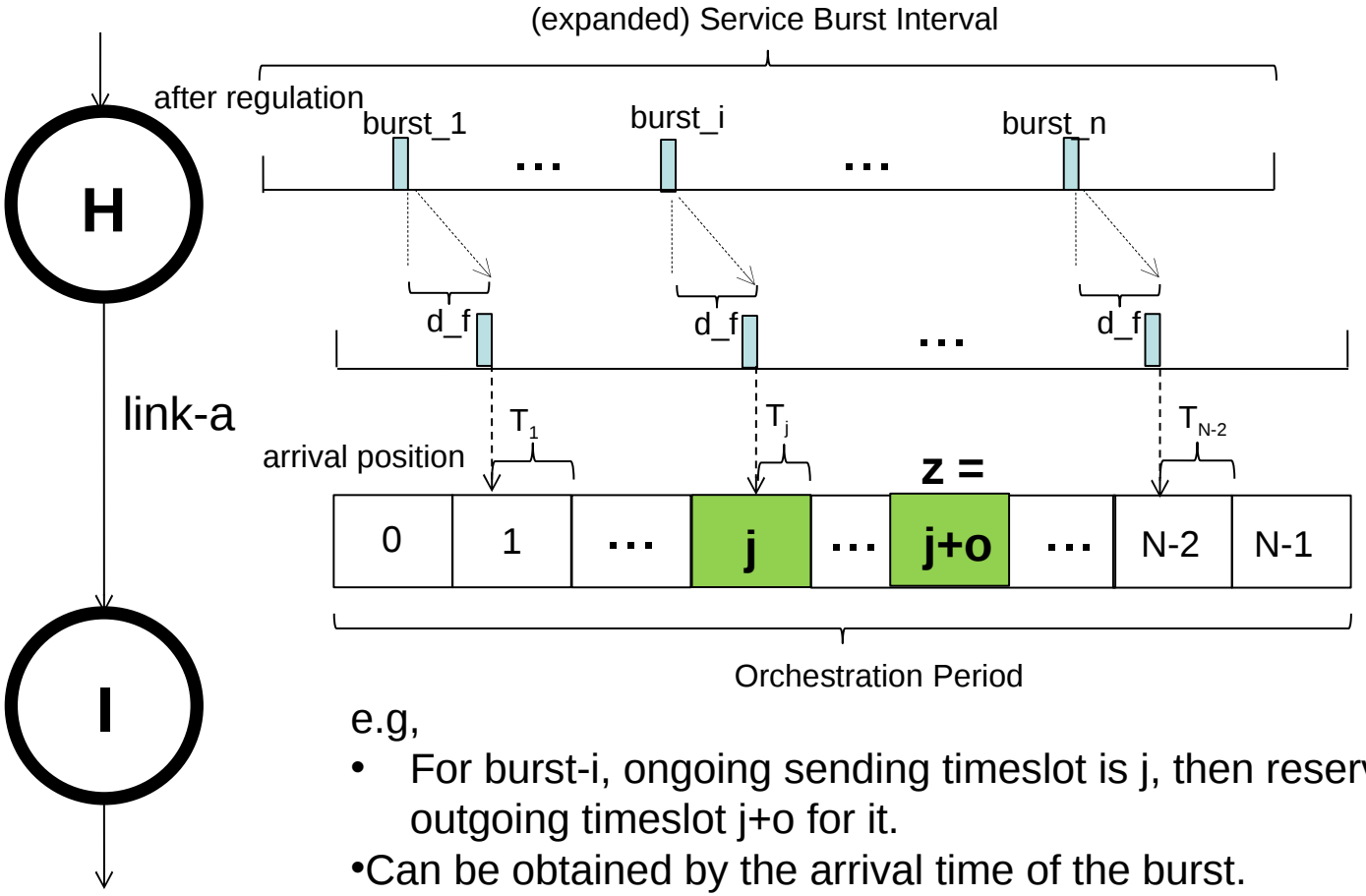
Overview of TQF Mechanism



- Decouple the Orchestration Period from the Scheduling Period. $N = k*M, k \geq 1$
  - The Orchestration Period includes slot resources that can be reserved, orienting towards service;
  - The Scheduling Period matches the actual capacity of the device, requiring only a few queues.
- Path calculation based on slot resource reservation. Obtain a flexible mapping relationship between the incoming and outgoing slots on each node.
  - The sum of all nodes' residence delay must meet the total residency delay budget.
- Map Orchestration Period slots to Scheduling Period slots, and access slot resources without conflicts.

1. Determine the Ongoing Sending Timeslot

- Before timeslot resource reservation on each node, it is necessary to determine the ongoing sending timeslot when the flow arrives at the node.
  - **On the headend node**



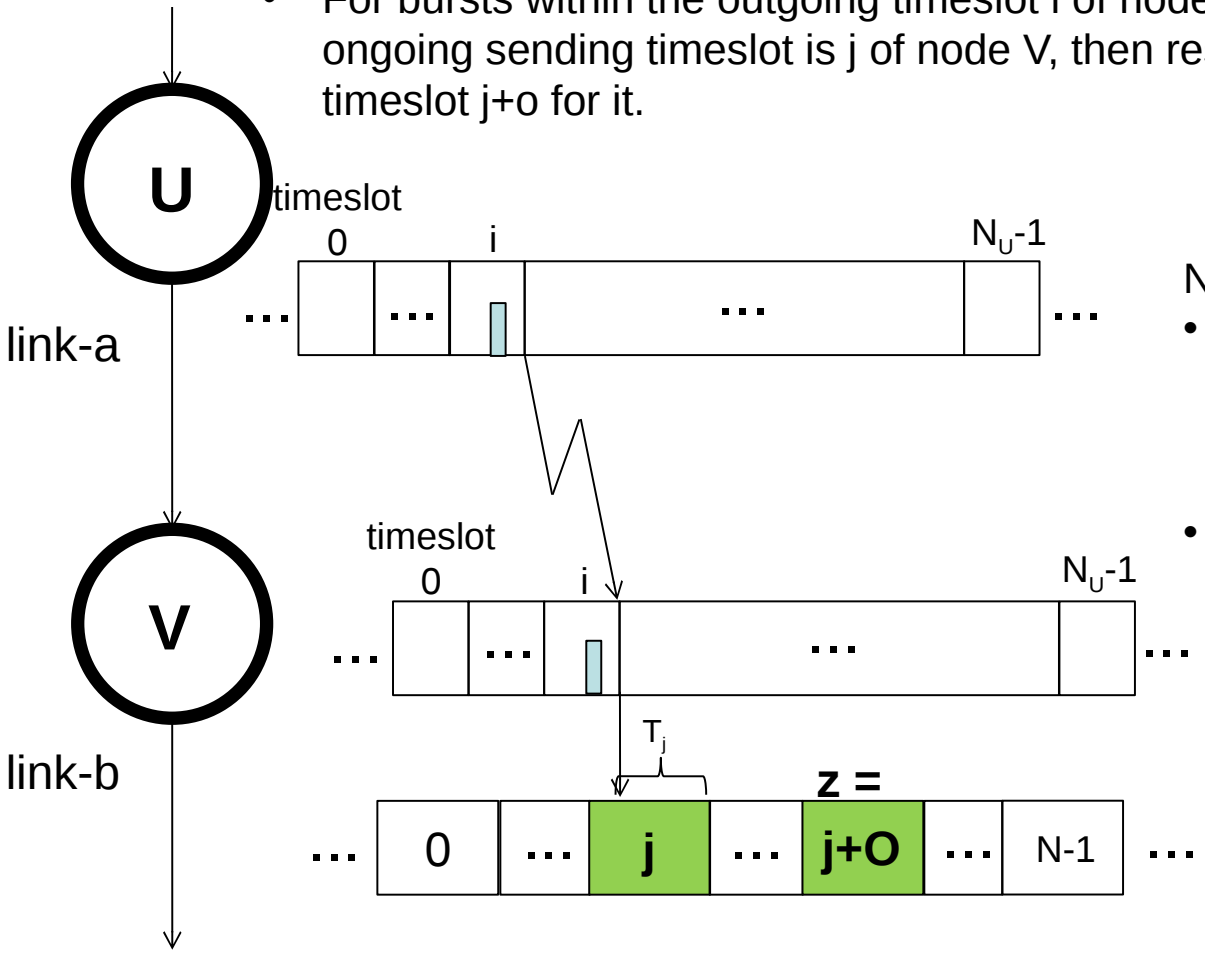
e.g,

- For burst-i, ongoing sending timeslot is j, then reserve outgoing timeslot j+o for it.
- Can be obtained by the arrival time of the burst.

- **On the transit nodes**

e.g,

- For bursts within the outgoing timeslot i of node U, the mapped ongoing sending timeslot is j of node V, then reserve outgoing timeslot j+o for it.

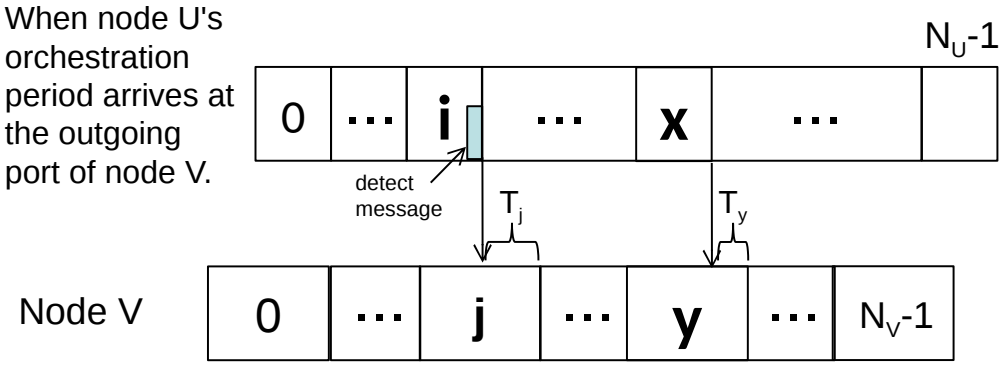


Note:

- The basic mapping i -> j is irrelated with any flows.
  - Can be obtained by detecting messages.
- However, the forwarding mapping i -> j+o is related with the path established for flows.

1.1. Incoming -> Ongoing Sending Timeslot

- Deduced by a single timeslot mapping detection

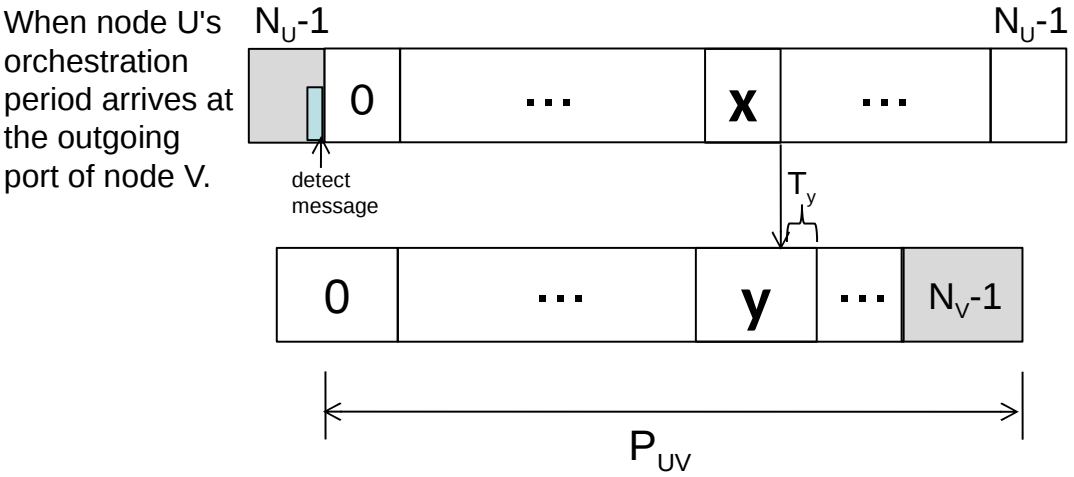


If i -> j is detected, for any x, we have x -> y, where:

$y = (j + ((N_U+x-i)*L_U-T_j)/L_V + 1) \% N_V$

$T_y = L_V - ((N_U+x-i)*L_U-T_j)\%L_V$

- Deduced by phase difference of orchestration period (P_{UV})



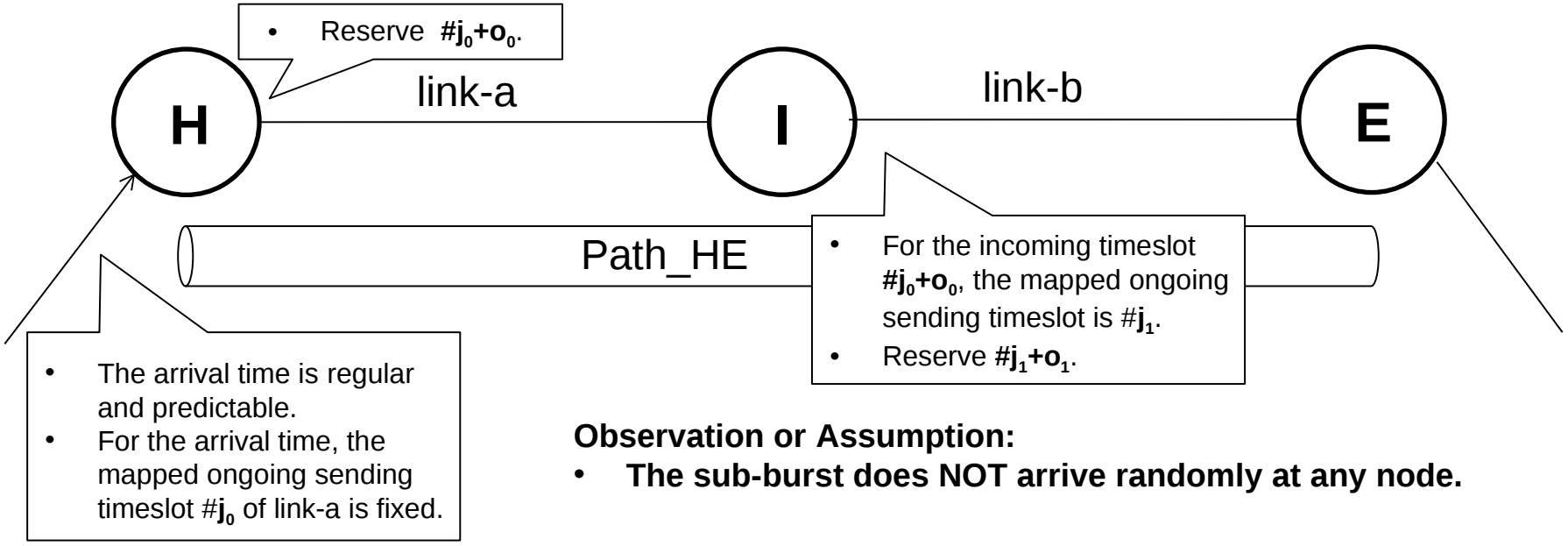
If P_{UV} is detected, for any x, we have x -> y, where:

$y = ((LOP+(x+1)*L_U-P_{UV})/L_V) \% N_V$

$T_y = L_V - (LOP+(x+1)*L_U-P_{UV})\%L_V$

2. Reserve Timeslots on the Control Plane

- Maintain a reservation sub-task for each sub-burst (note: there may be multiple sub-bursts in the service burst interval, and each may consume different timeslot).
  - Firstly allocate a fixed outgoing timeslot **(i.e., #j_0+o_0)** for the sub-task on the headend, according to the fixed arrival position **(i.e., #j_0)** of that burst within the orchestration period.
  - Then, the fixed outgoing timeslot of the headend node, will map to a fixed ongoing sending timeslot **(i.e., #j_1)** on the first transit node, and continue to allocate a fixed outgoing timeslot **(i.e., #j_1+o_1)** on that transit node.
  - And so on, till the endpoint.

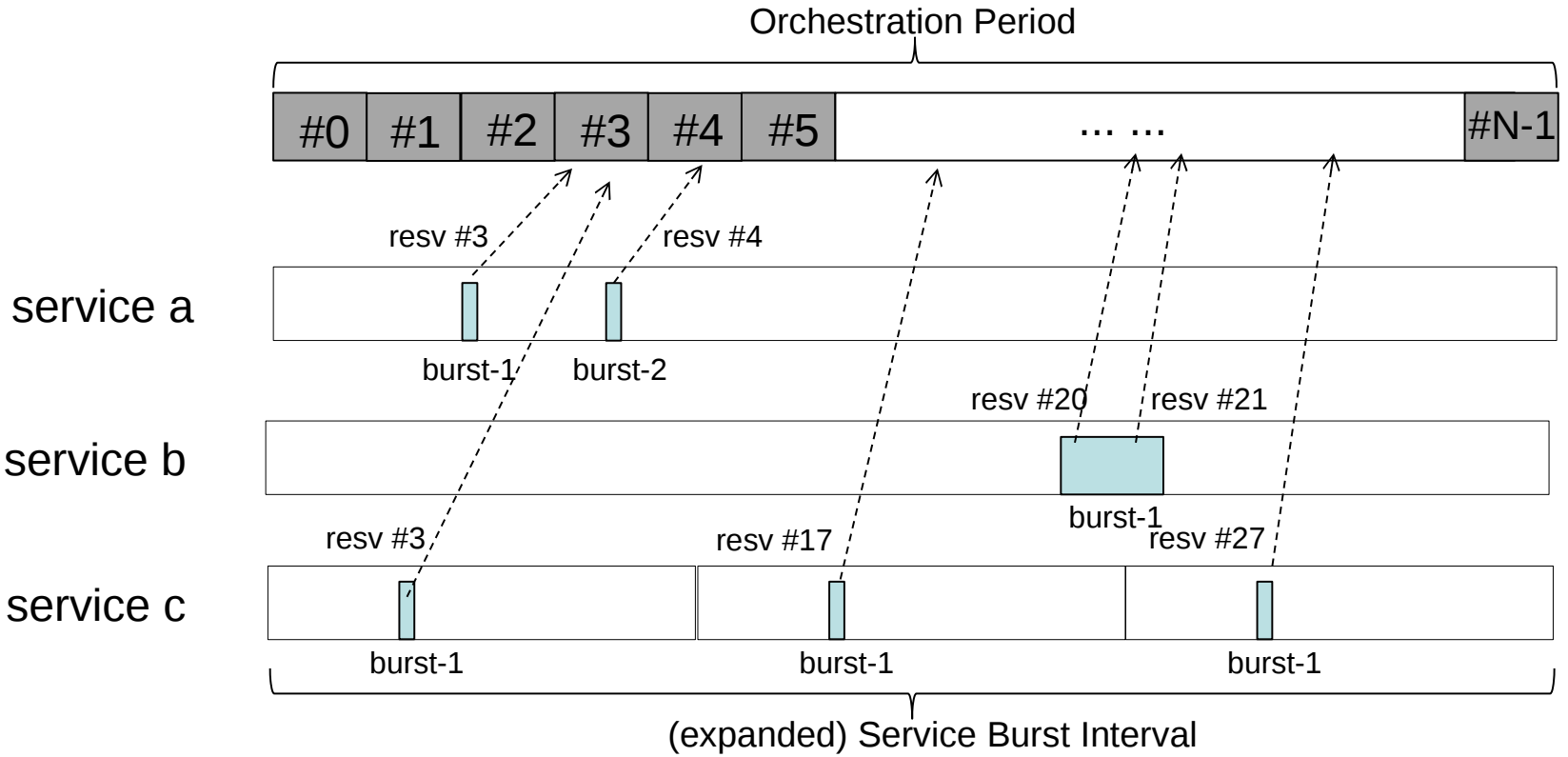


Observation or Assumption:

- **The sub-burst does NOT arrive randomly at any node.**

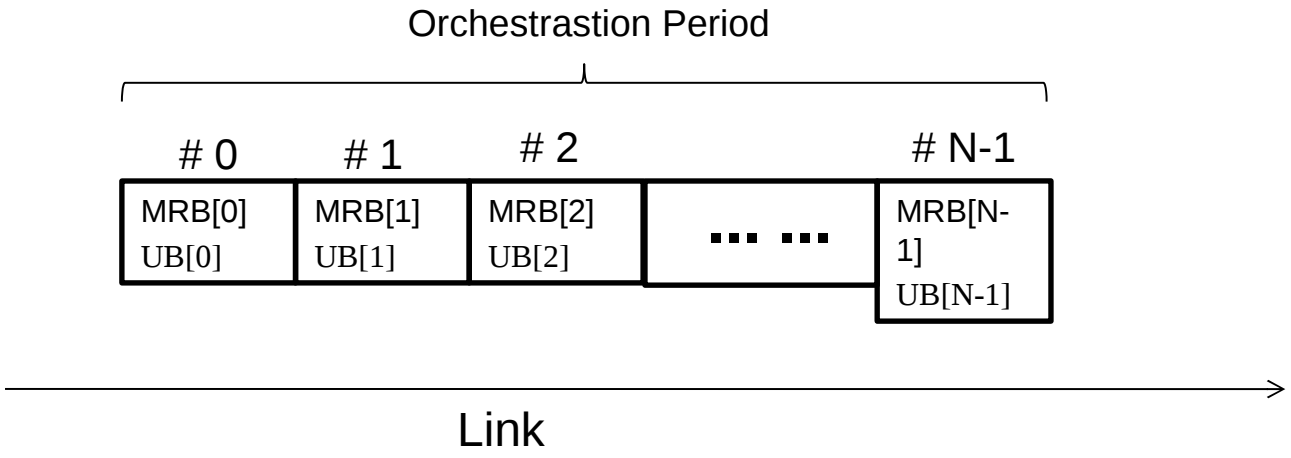
2.1. Fixed Arrival Position, is this really the case?

- Position deviations may happen, but must be limited. Should consider this deviation during reservation procedure.
- If position is completely random, it is difficult to allocate timeslots, more work:
  - An explicit buffer can be used to store packets to get the fixed position.
  - Or let scheduling period = orchestration period, to directly store packets in the queue.
  - In any options, extra delay may be introduced and control plane does not aware that.



2.2. Timeslot Resource Definition

- Each link has its own timeslot resources information.
  - Timeslot Length **(L_T)**
  - Length of Orchestration Period (**LOP**)
  - Length of Scheduling Period (**LSP**)
  - For each timeslot within the orchestration period, it continues to include:
    - Maximum Reservable Bursts (**MRB**), < L_T * C
    - Unreserved Bursts (**UB**)



2.3. A Possible Reservation Algorithm

- Get **Total Residence Budget**, which equals to the end-to-end delay minus the sum of all link propagation delay.
- Set **Individual Residence Budget** for each node along the path.
  - A single average budget value for all nodes
  - Or, a list containing different budget value for different node.
- Allocate outgoing timeslot on each node according to the reslut of the **Individual Residence Budget** plus the **Accumulated Residence Deviation**.
  - Get the **Residence Evaluation** on that node by the following formula.

    headend: $D_f + T_j + (2o-1)*L_H/2$

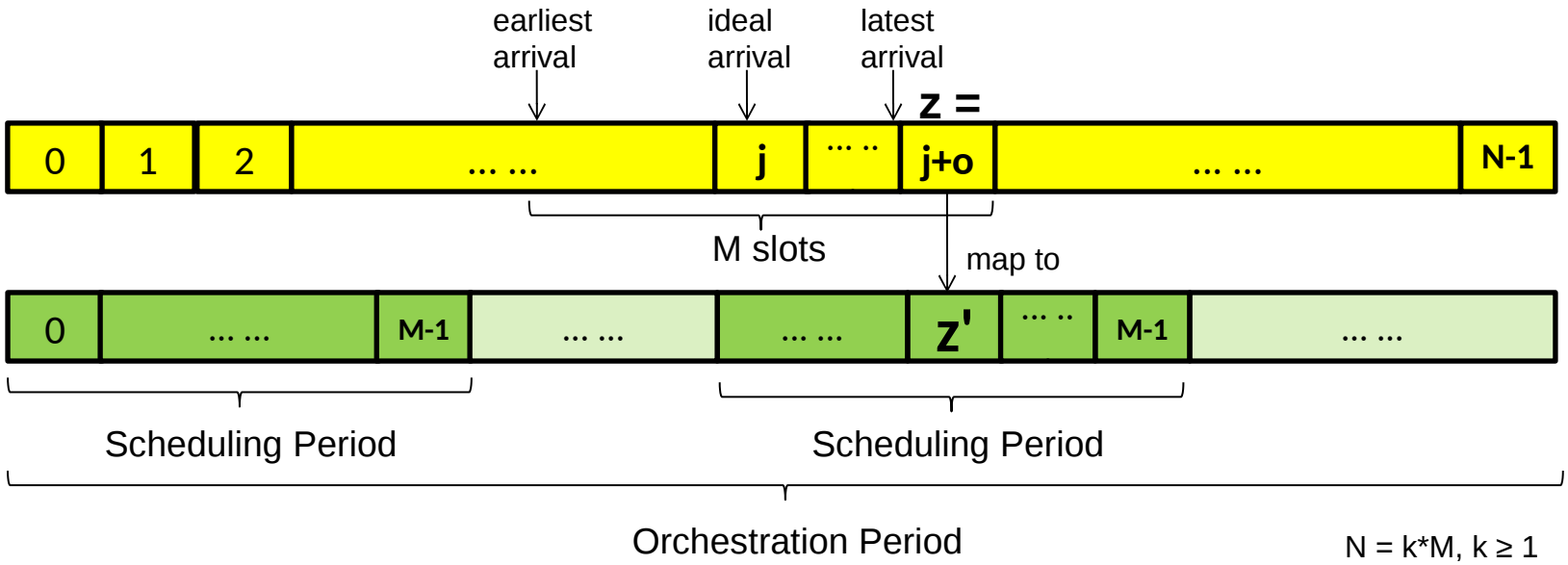
    transit: $D_f + T_j + (L_U+(2o-1)*L_V)/2$

    D_f: intra node forwarding delay
    T_j: remaining time of the ongoing sending slot.
    o: offset of outgoing slot related to ongoing sending slot
    L_H: timeslot length of the headend node.
    L_U: timeslot length of upstream node U.
    L_V: timeslot length of this node V.

  - Update the Accumulated Residence Deviation, according to the above result minus the **Residence Evaluation**.
- The sum of all nodes' **Residence Evaluation** must meet the **Total Residence Budget**.

3. Access the Timeslots on the Data Plane

- The data packets carry timeslot z@orchestration period.
- Map timeslot z@orchestration period to timeslot z'@scheduling period.
  - If the constraint o < M is always followed during reservation procedure, the mapping rule is **z' = z mod M**.
  - The ideal arrival time of the burst will let the burst fall into the ongoing sending timeslot #j.
    - The latest arrival that can be tolerated fall into #j+o-1, i.e., the maximum position deviation caused by late arrival is o-1.
    - The earliest arrival that can be tolerated fall into #j+o-(M-1), i.e., the maximum position deviation caused by early arrival is M-o-1.



4. Admission Control on the Ingress

- Traditional traffic regulation per flow on the ingress node.
  - Leaky bucket depth(CBS) is set to cover the reserved burst resource.
  - Leaky bucket rate(CIR) is set to cover the reserved bandwidth resource.
- Maintain state for each sub-burst of the flow, since different sub-bursts may consume different timeslots.
- Optional:
  - If the received traffic has random arrival time, an explicit buffer may be placed before the TQF scheduler to let each sub-burst obey its fixed arrival position within the orchestration period.
  - Note that the asynchronous runtime timeslot mapping describe later does not relay fixed arrival position.

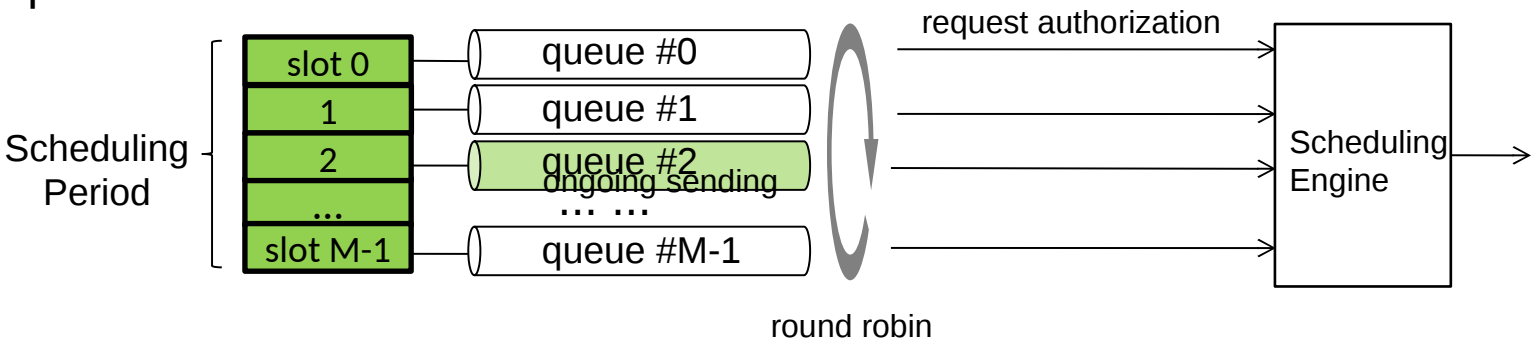
5. Global Timeslot Style

- When all nodes have the same timeslot length, and don't want to maintain trivial mapping states per path, global timeslot style may be useful.
- Feature:
  - for incoming timeslot i that mapped to any ongoing timeslot j (j ≠ i), always reserve outgoing timeslot i, i.e., with **constant o = (N+i-j)%N**.
    - If j is unfortunately equals to i, an explicit buffer should hold the packet to let it fall to a new ongoing sendding timeslot j .

Reservation Strategy	Timeslot Style	Referrence
Flexible o where 1 ≤ o < M	Local timeslot style	[this proposal]
Constant o where o = (N+i-j)%N	Global timeslot style	[this proposal]
Constant o where o = 1	Multi-CQF, a special case of local timeslot style	[Multi-CQF]

5.1. Asynchronous Runtime Timeslot Mapping

- This is a varaints of global timeslot style.
- Feature:
  - Even on the control plane the constant outgoing timeslot i is reserved for the incoming timeslot i (i.e., synchronous mapping i -> i), it may take asynchronous runtime timeslot mapping (i -> x) on the data plane, where x is a nearby slot than i.
  - With work-conserving behavior, not guarantee jitter.
- Implementation



  - Packets with incoming timeslot i are still insered to the queue for the timeslot i, so that there is no possibility of overflow.
  - Each queue if is not empty, request authorization to send packets. For all queues:
    - Based on round robin, the queue for the current ongoing sending timeslot has the highest priority, the queue for the next ongoing sending timeslot has the second highest priority, and so on.
    - The authorization bytes obtained each time may be 2000 bytes (a preset maximum size for DetNet service), so that to get a limited interference latency by low priority queue.
    - The queue for the ongoing sending timeslot may request multiple authorization to clean itself.

Questions & Comments ?