Large-Scale Deterministic Network (LDN)

draft-qiang-detnet-large-scale-detnet-00

Li Qiang (Christina) qiangli3@huawei.com, Liu Bingyang (Bryan) liubingyang@huawei.com
Toerless Eckert tte+ietf@cs.fau.de (Huawei USA),
Liang Geng gengliang@chinamobile.com, Lei Wang wangleiyjy@chinamobile.com

v1.2

Goals

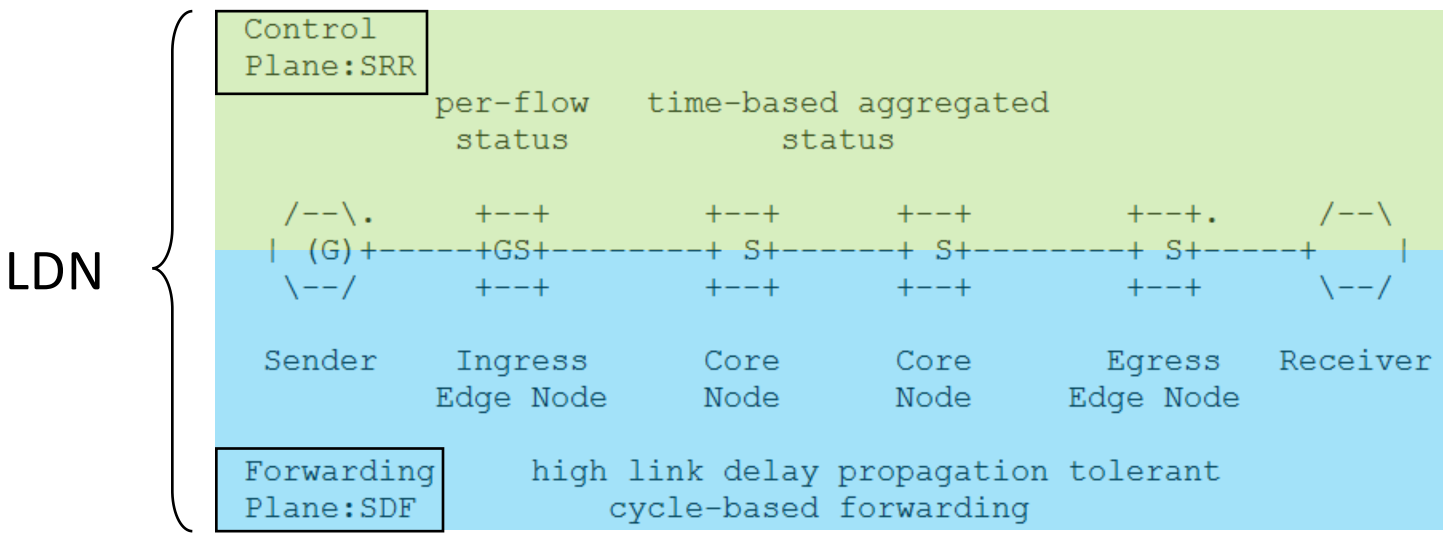
- Propose work for bounded delay solution with IP/MPLS/SR forwarding
  - Made specific to match DetNet requirements.
  - ‘Tightly bounded’ delay:
    - [min..max] range for end-to-end propagation delay. Small (max – min) [“jitter”]
    - Delay may be large, eg: 5 msec – but most link propagation delay (speed of light)

- Control Plane and Forwarding Plane modular from each other
  - Can start working/finalizing forwarding independent of control plane
    - Might have different control planes (central SDN .. RSVP modifications .. Better)

- Forwarding Plane derived from TSN/CQF (Cyclic Queuing Forwarding) principles
  - No per-flow state in forwarding plane.
    - Sender or first-hop router have “per-flow gate” as in TSN (time controlled shaper/scheduler).
  - Remove (TSN) need for short link propagation delay and tight clock synchronization
    - Want to be able to deal with links with usec…msec delays in WAN L3 networks

Overview

- Large-scale Deterministic Network (LDN) consists of
  - Scalable Resource Reservation (LDN-SRR) at control plane
  - Scalable Deterministic Forwarding (LDN-SDF) at forwarding plane
- SRR and SDF can be used independently

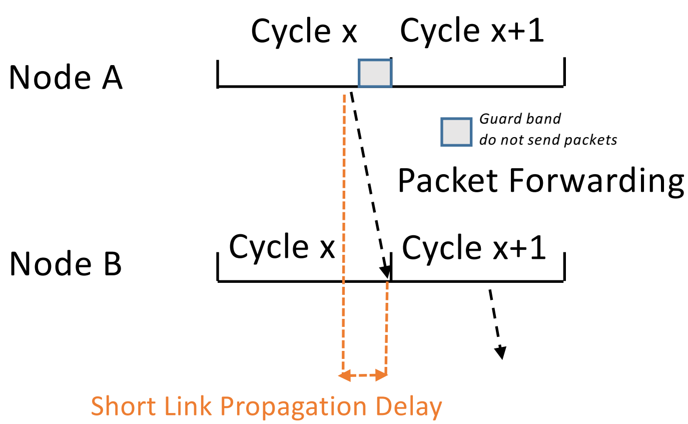


LDN-SDF(scalable deterministic forwarding) –
long link propagation tolerance, bonded latency

- Cyclic forwarding – the length of a cycle is T, irrelevant of time slot in control plane
- Long link propagation tolerance & bounded latency

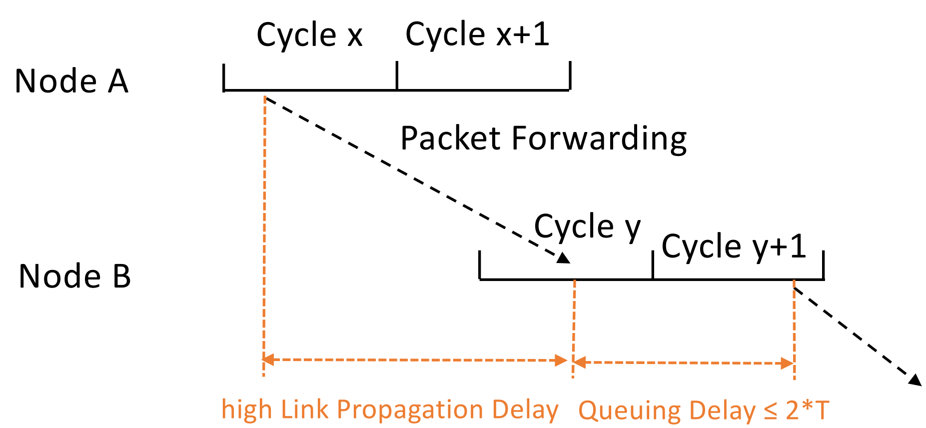
Pre-existing: TSN solution

TSN-CQF (time-synchronization)



Link propagation delay is required to small percentage of T,
T couldn't be too large since End-to-End Jitter's upper bounder is 2*T. Therefore link propagation delay couldn't be very high.

LDN-SDF (frequency-synchronization)



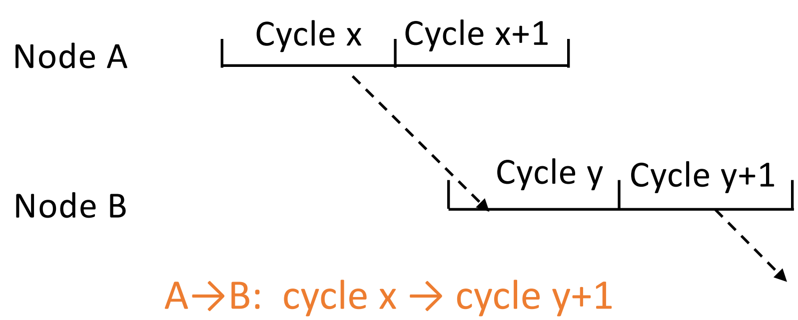
End-to-End Jitter ≤ 2*T
End-to-end Queuing delay ≤ 2*T*hops

SDF – cycle mapping, three queues

Simple (when ignoring non-queuing propagation delay variation)

- Link-propagation delay variation, in-node delay variation (non-queuing,…)

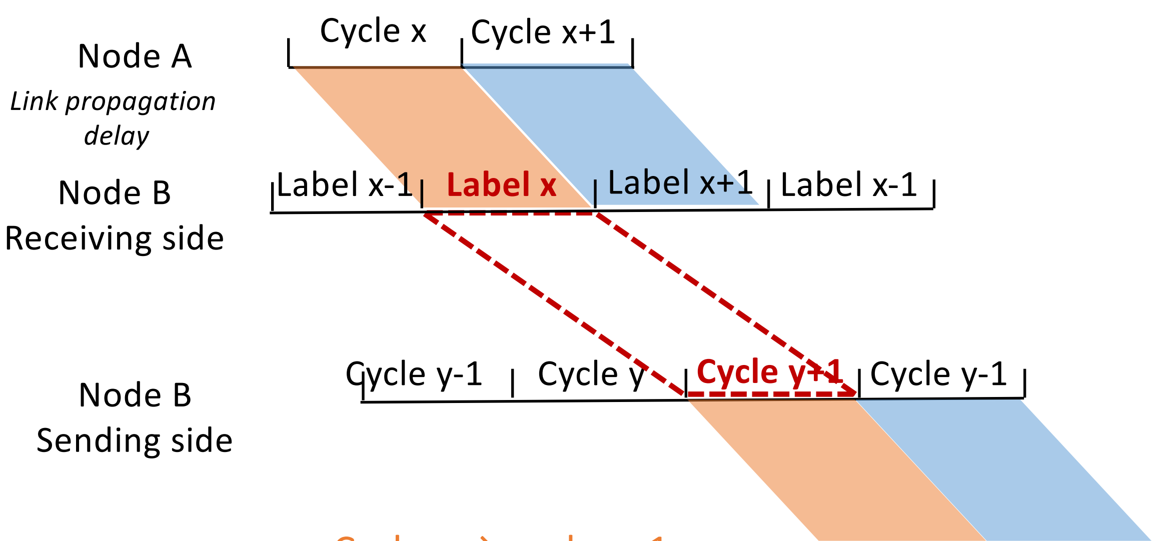
Each pair of neighboring nodes has a stable cycle mapping relationship that could be used to indicate the packet forwarding time



The cycle mapping relationship could be notified by control plane, also can be self-studied in a distributed way

SDF – cycle mapping, three queues

- Cycle y may receives packets sent from two cycles (x and x+1) due to loose time-synchronization, hence needs two receiving queues



three cyclic queues at each output port

Cycle x → cycle y+1
cycle x+1 → cycle y-1
cycle x-1 → cycle y

2 bits to carry cycle-identifiers

- ✓ DSCP of IPv4 Header
- ✓ Traffic Class of IPv6 Header
- ✓ TC of MPLS Header (used to be EXP)
- ✓ EtherType of Ethernet Header
- ✓ IPv6 Extension Header
- ✓ TLV of SRv6
- ✓ TC of MPLS-SR Header (used to be EXP)
- ✓ Three labels/adjacency of SIDs for MPLS-SR
- ✓ Etc.

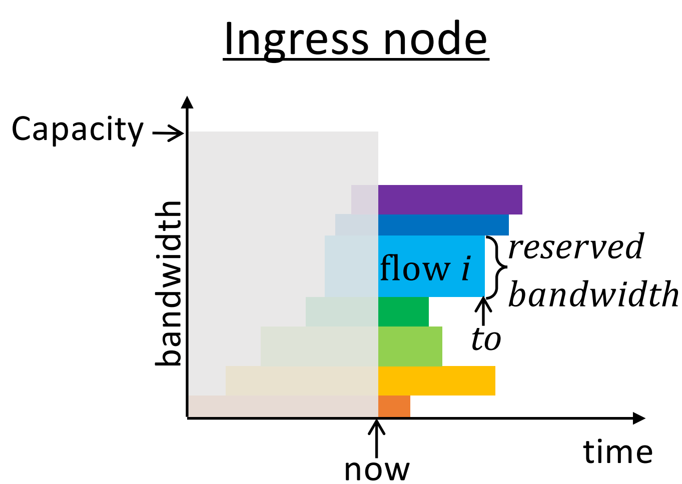
Sync requirements & slot requirements

- Frequency synchronization
  - Do not want drift between slot mappings => ideal clock frequencies are synchronized!
  - If there is known clock drift, additional cycles could be used and keep the cycle mapping stable for longer time.
    - Map to +3 cycles -> over time receiver clock runs faster, delay for sending cycle +3 shrinks.
    - Remap (after hopefully long time and when there is no traffic).

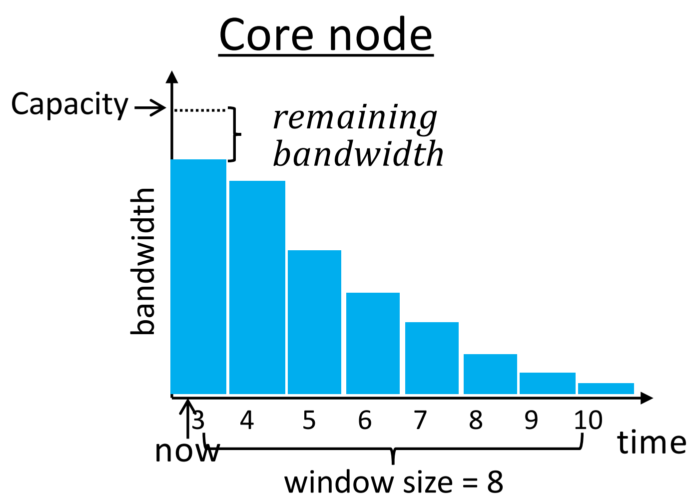
- Link delay variation
  - Assume slot is 10 usec. If link propagation delay varies over time and is larger than 10 usec, what then ?
  - Inband measure long term min/max delay (by received slot labels in packet)
  - Need to use more slots: Need to map slot so that all packets from source slot can make it into the destination slow with both min and max propagation delay
  - Reasons for link propagation delay variation ?
    - Hanging copper cables on poles in the heat (>= 10% variation)

SRR(scalable resource reservation) – Scalable at Core Node

- Ingress edge node maintains per-flow resource reservation states
- Core node aggregates per-flow resource reservation states in time slots
- Core node refreshes ABRW according to the per-flow information maintained at Ingress node through RSVP



Per-flow Bandwidth Reservation Status



Aggregated Bandwidth Reserved Window

Summary: LDN (large-scale deterministic network)

- Forwarding Plane – SDF (scalable deterministic forwarding)
  - ✓ no per-flow status at core nodes
  - ✓ loose time-synchronization
  - ✓ end-to-end jitter ≤ 2*T
  - ✓ end-to-end queueing delay ≤ 2*T*hops
  - ✓ queue size of each output port= 3*port rate*T

- Control Plane – SRR (scalable resource reservation)
  - ✓ no per-flow status at core nodes

Thank You!

Demo at Coffee Break Main Area Now