Deterministic Networking (DetNet) Data Plane
Flow interleaving

for scaling detnet data planes with minimal end-to-end latency and large number of flows

draft-eckert-detnet-flow-interleaving-01

Toerless Eckert, Futurewei Technologies USA (tte@cs.fau.de),

IETF 118 DetNet WG meeting , rev 0.1

https://github.com/toerless/detnet/tree/main/slides

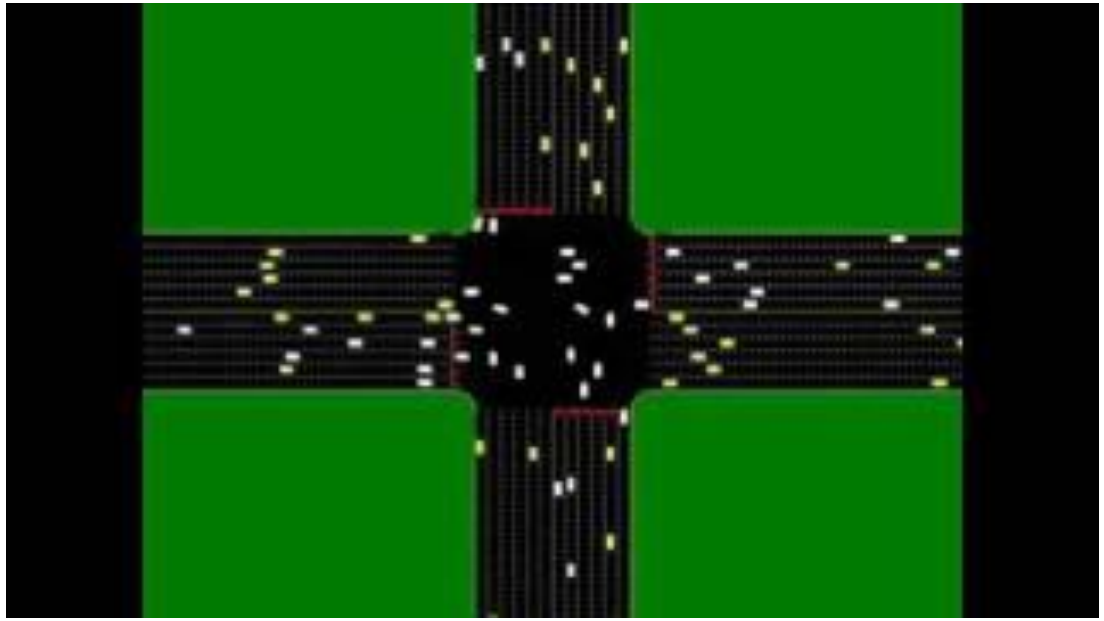
Draft Goal / Summary

- To achieve high utilization for bounded latency traffic:
  - We need a per-hop tightly bounded jitter forwarding mechanism
- This presentation improvement only in slides (no draft update)
  - Taking into account results from discuss during interims

- Yikes – is is TDM ? Yes, it is, but primarily in the control plane.

TDM is also becoming popular for reducing delay, incerasing throughput in city traffic (research)

Autonomous cars allow to be timed such that they can drive simultaneously across intersections without colliding.

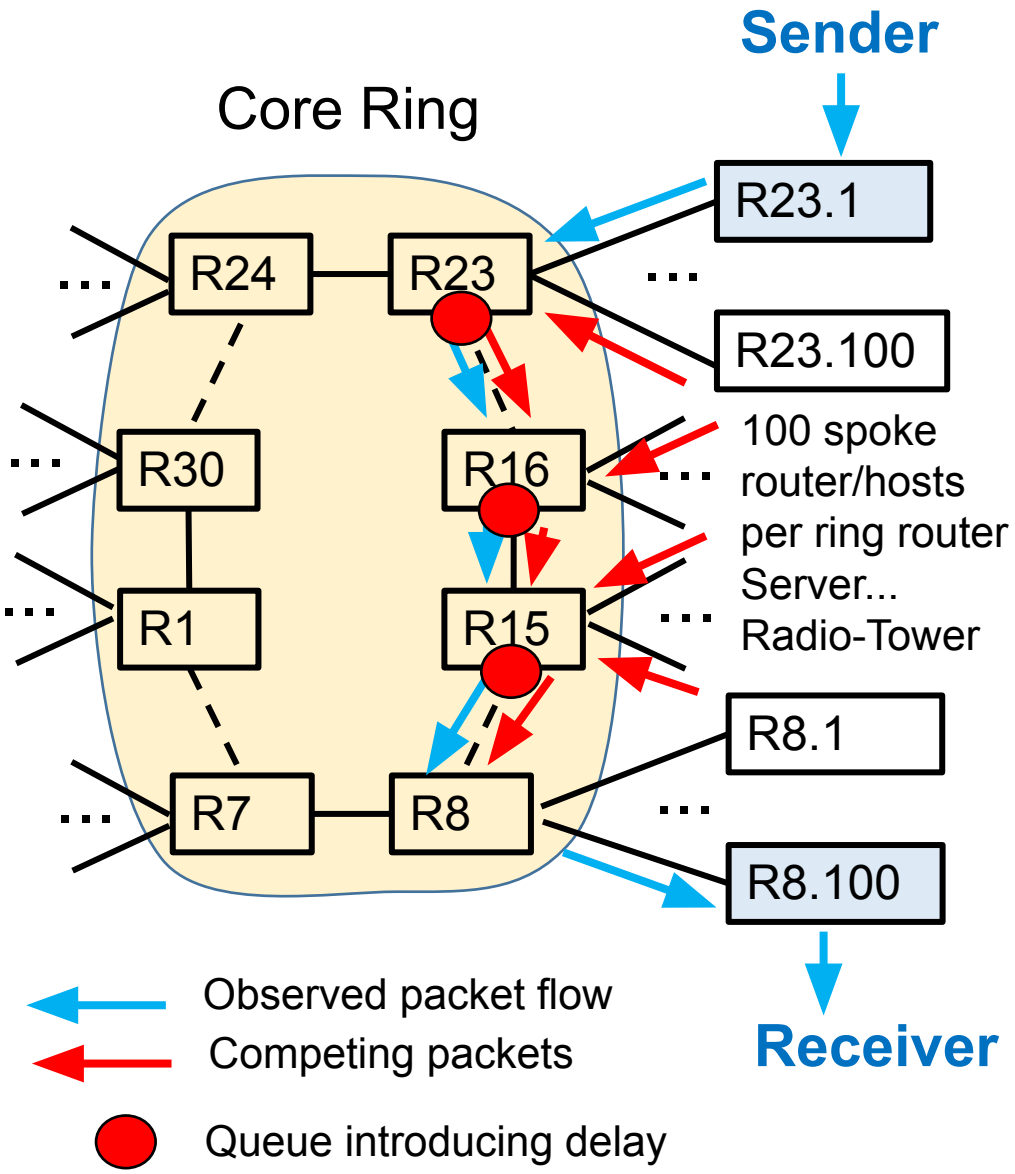


- **Autonomous Intersection Management**
  e.g.: youtube.com/watch?v=r7_lwq3BfkY

Example (worst case) use-case

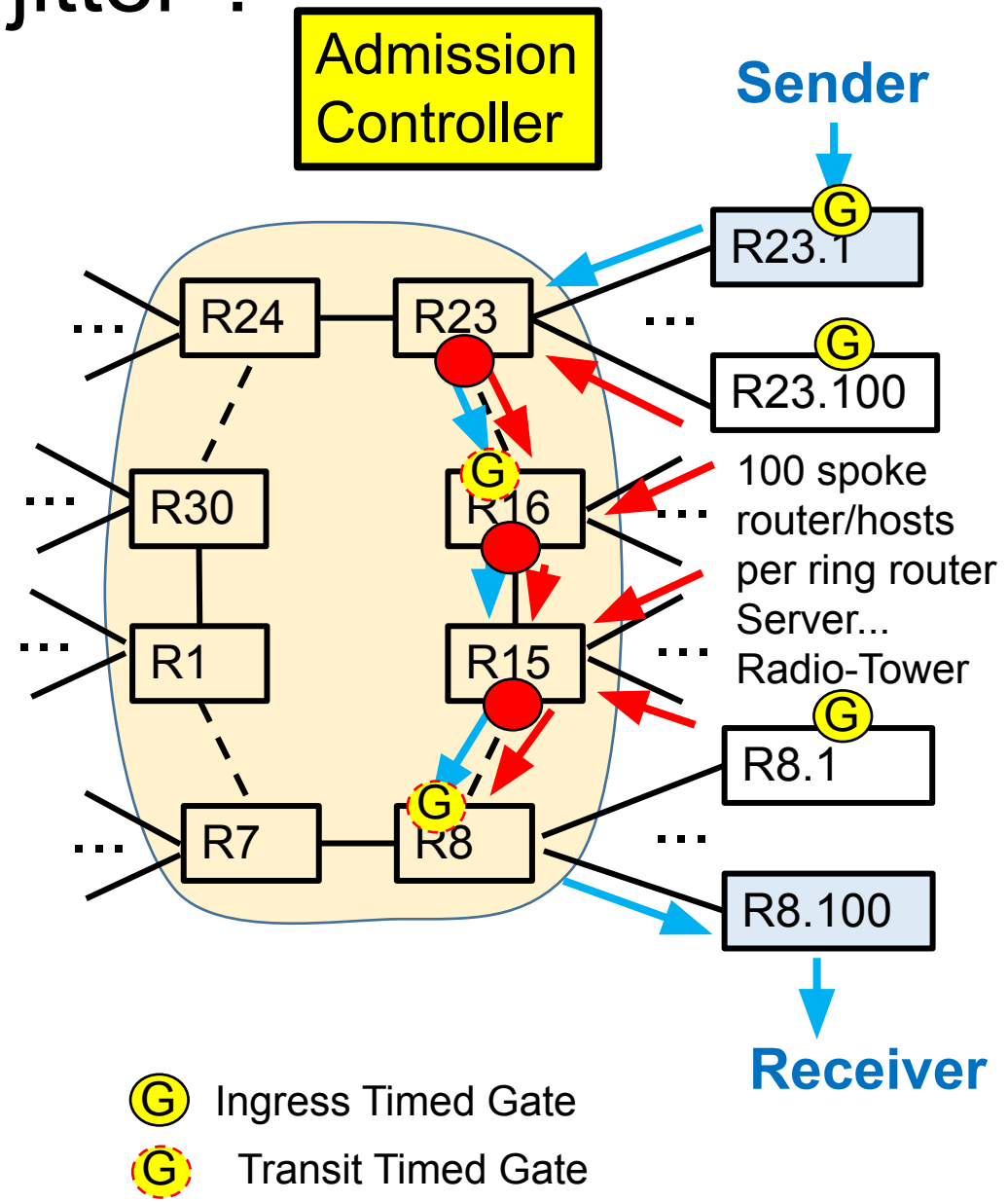
- Metro-Ring, e.g.: 30 hops , 100 spoke nodes/ring router. Assume all links equal speed.
- 1 flow observed. across 14 ring hops
- On each ring router, worst case we receive **simultaneously** a packet from each input interface (maximum burst collision)
- Packet of observed flow may need to be queued behind 100 other packets on this hop.
  - WFQ (IntServ) does not mitigate this delay
- 14 hops * 100 = 1400 packet serialization time worst case unavoidable minimum **jitter** (jitter, because this delay does not occur in absence of competing packets).

- Problem increases if flows need to be allowed to have large bursts (back-to-back packets). E.g.: 10 back to back packets.
  - Jitter also increases x10.
- Depending on queing algorithm used, the per-hop jitter can even be higher (e.g.: FIFO queuing in TSN-ATS)



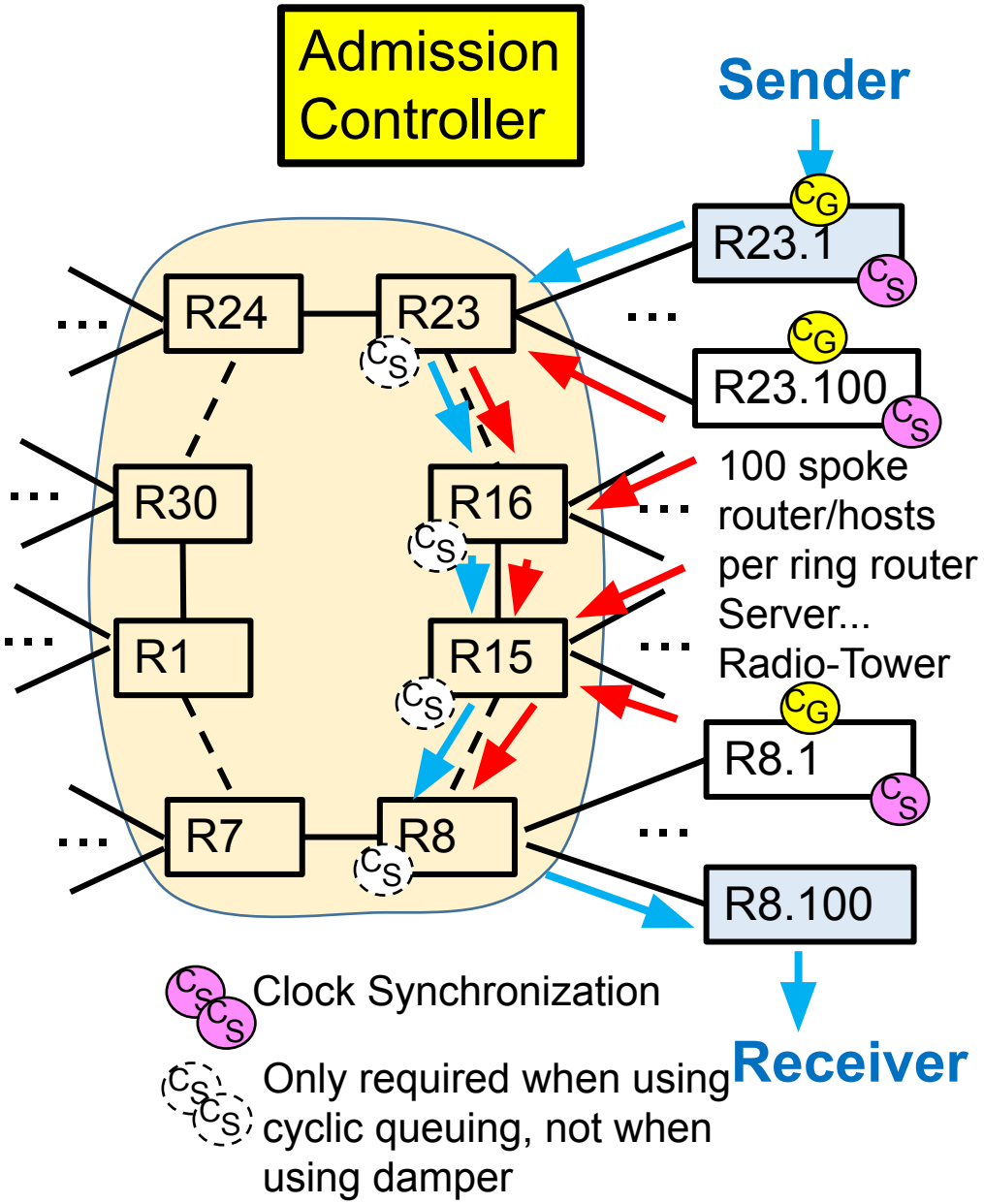
How would TSN avoid this delay/jitter ?

- **Timed Gates (G)** can delay packet of a flow up to specific times and release them (e.g.: periodic at certain time offsets).
- **Admission controlle**r calculates arrival times of each flows packet at each competing queue and deduces / calculates an appropriate
  - If impossible to resolve, additional gates on transit routers (e.g.: R16, R8) need to be added.
- Result: Flow interleaving packet-by-packet on each queue
  - Without incurring queue delay (under burst collision)
  - Except on transit timed gates
- Not applicable for DetNet scale (IMHO)
  - Can not have per-flow gates on transit nodes (scale, Opex)
  - Accuracy of calculation of arrival time at queues limited: delay variation because of (complex) node processing variations, link delay variations (FEC, restransmit, length changes), clock jitter relative to serialization speeds (100Gbps links or more).



Damper: beyond cyclic queuing

- **Cyclic Queuing** requires (cycle time level accuracy) clock synchronization across all hops.
- Damper solution (such as gLBF) aims to eliminate this requirement.
  - Only requires clock frequency synchronization across single hops with relevant packet-to-packet jitter(e.g.: radio links)
  - gLBF uses TSN-ATS calculus/queuing, but this applies to any queuing/latency calculus when a damper is added.
- Ingress (Cycle) gates still requires clock synchronization between each other.
  - But if these are eg: on/related to radio towers, they need that anyhow
- Dampers can be used with ingress cycle gates because arrival time on each hop can accurately be calculated (to the degree of cross-hop clock accuracy).
  - To be able to use cycle based behavior, per-hop dampened latency L needs to be increased to be multiple of the choosen cycle time at the CG/admission controller
  - L+= C - L mod C (where C is cycle time)



Summary
in-time queuing enables flow interleaving = higher utilization at lower latency and jitter

- Queuing mechanisms with per-hop low-jitter (“in-time forwarding”), cyclic-queuing and damper based already have the benefit of providing low jitter without additionally large “play-out buffers” on receivers or receiver edge-router.
  - This is already beneficial / required for many applications
- But because they allow for admission controller to calculate arrival times (cycle / time-window) at each hop, it allows admission controller to increase utilization of network with DetNet traffic
  - By combination with ingress timed gates – operating e.g.: cycle periodic.
- Adding such functionality to admission controller seems easy (compared to other admission control problems).
  - This is only an option – when needed due to increasing utilization with DetNet.
  - In-time queuing mechanisms do not give us this option.