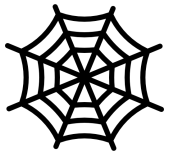
OrbWeaver: 

Using IDLE Cycles in Programmable Networks for Opportunistic Coordination

Liangcheng Yu, John Sonchack, Vincent Liu

Networks are woven from packets

- A primary goal of computer networks: ***deliver packets***
  - ***User application***: video streaming, web browsing, file transfer…
  - ***Non-user application***: control messages, probes about network state, keep alive heartbeats…

Networks are woven from packets

- A primary goal of computer networks: ***deliver packets***
  - ***User application***: video streaming, web browsing, file transfer…
  - ***Non-user application***: control messages, probes about network state, keep alive heartbeats…

- Often, two classes of traffic ***multiplex*** the same network

When introducing a new in-band application…

To consume ***extra BW*** for ***fidelity*** *(of the control application)*, or not to?

- *Time synchronization*: clock-sync rate → precision
- *Failure detector*: keep alive message frequency → detection speed
- *Congestion notification*: signaling data and rate → measurement accuracy

When introducing a new in-band application…

To consume ***extra BW*** for ***fidelity*** *(of the control application)*, or not to?

- *Time synchronization*: clock-sync rate → precision
- *Failure detector*: keep alive message frequency → detection speed
- *Congestion notification*: signaling data and rate → measurement accuracy

Is the trade-off between fidelity and overhead necessary?

When introducing a new in-band application…

To consume ***extra BW*** for ***fidelity*** *(of the control application)*, or not to?

- Failure detector: keep alive message frequency → detection speed
-
-

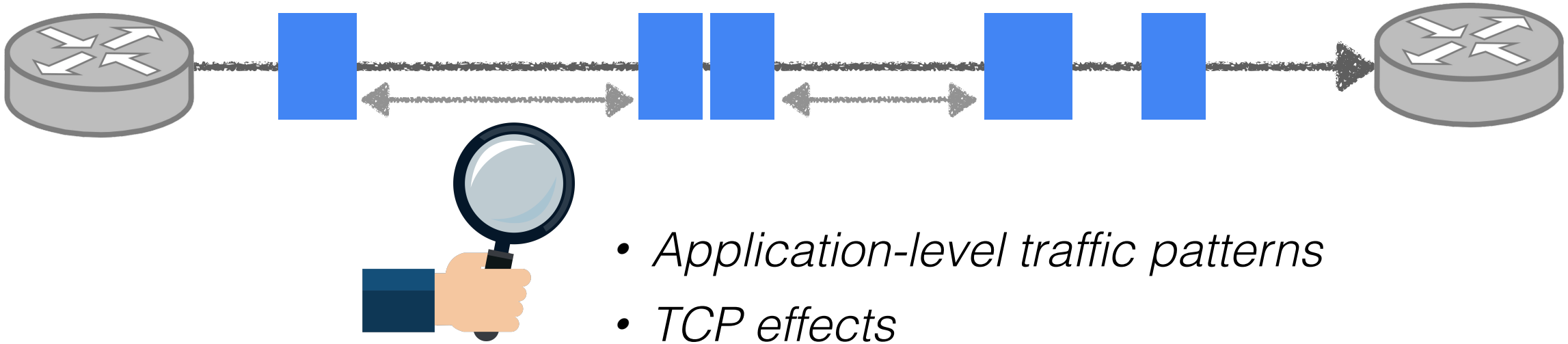
Can we coordinate at **high-fidelity** with a **near-zero cost** (to usable bandwidth, latency…)?

Can we coordinate at **high-fidelity** with a **near-zero cost** to usable bandwidth and latency?

Idea: Weaved Stream

- Exploit ***every gap*** (*O(100ns)*) between user packets opportunistically
- Inject customizable ***IDLE packets*** carrying information across devices

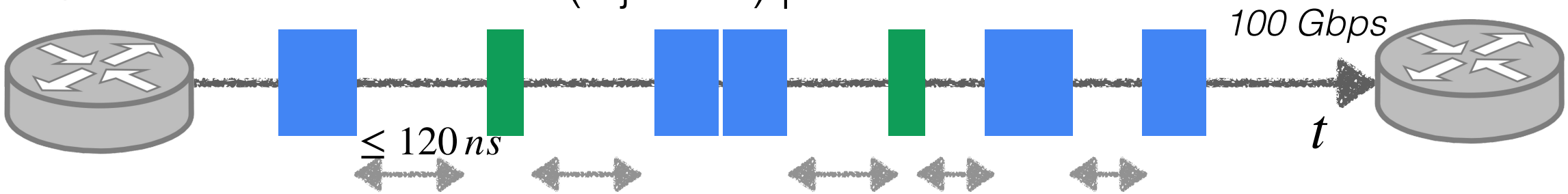
Opportunity: $< \mu s$ gaps are prevalent



- *Application-level traffic patterns*
- *TCP effects*
- *Structural asymmetry*
- *…*

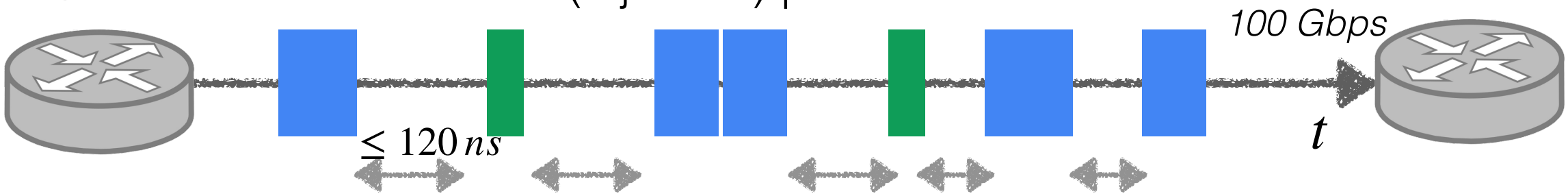
Abstraction: weaved stream

- Union of **user** and **IDLE** (injected) packets:



Abstraction: weaved stream

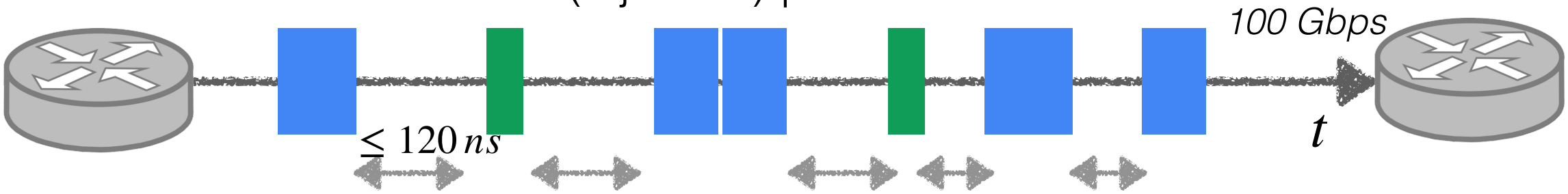
- Union of **user** and **IDLE** (injected) packets:



[R1 Predictability] Interval between ***any two consecutive*** packets $\leq \tau$

Abstraction: weaved stream

- Union of **user** and **IDLE** (injected) packets:



[R1 Predictability] Interval between ***any two consecutive*** packets $\leq \tau$

[R2 Little-to-zero overhead] Not impact user packets or power draw

Abstraction: weaved stream

- Union of user and IDLE (injected) packets:

Implement many ***in-network applications***
(failure detection, clock sync, congestion notification…)
for free!

1. [Predictability] Interval between ***any two consecutive*** packets $\leq \tau$
2. [Little-to-zero overhead] Weaved IDLE packets not impact user packets

Abstraction: weaved stream

- Union of **user** and **IDLE** (injected) packets:

1. Interv
2. Weaved IDLE packets incur ***little-to-zero*** impact to user packets

Crazy idea?

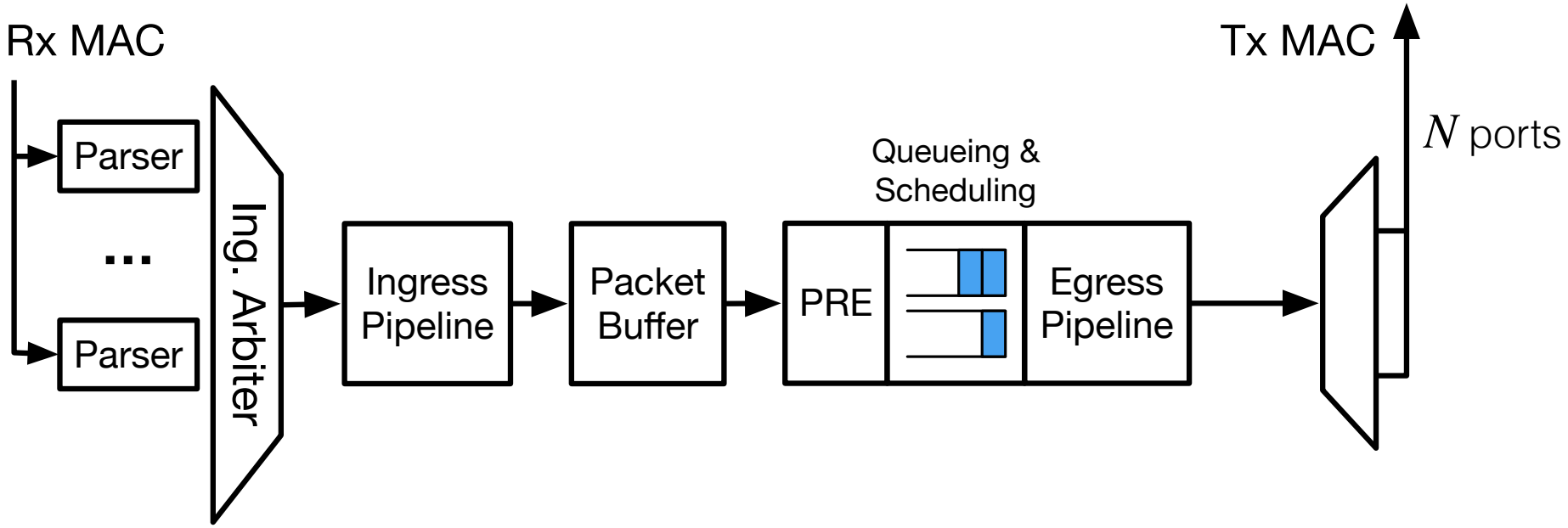
Extending IDLE characters to higher layers

- Data plane packet generator
- Replication engine
- Data plane programmability
- Flexible switch configuration (priorities, buffers…)

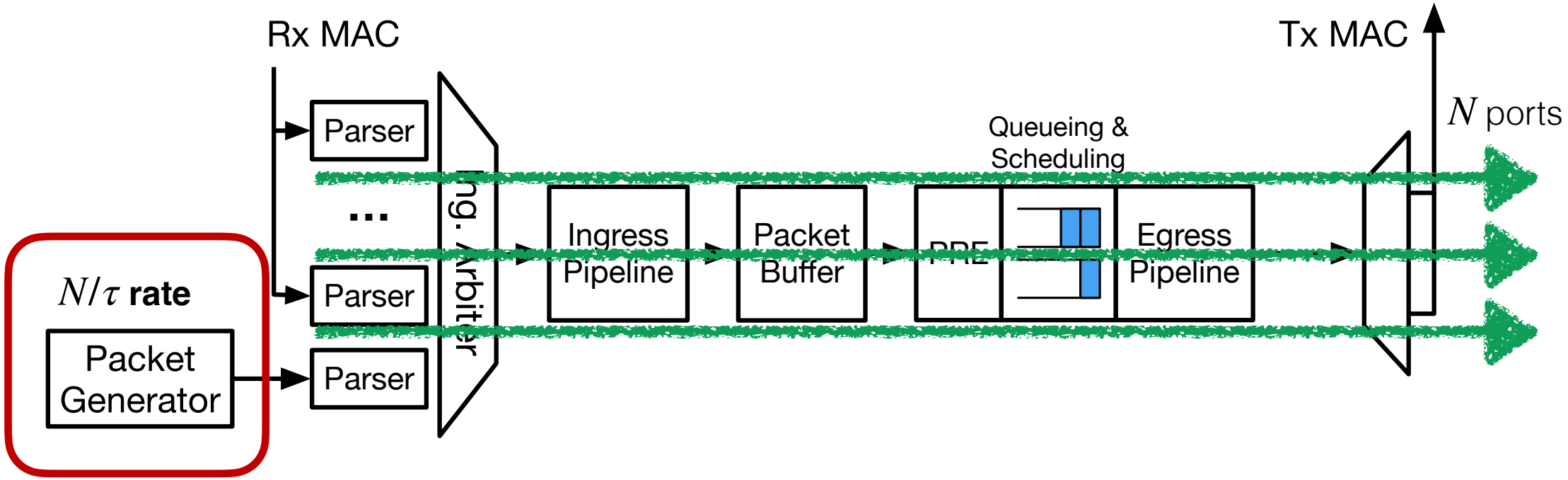
Outline

1. Switch data plane architecture
2. Weaved stream generation
3. OrbWeaver applications

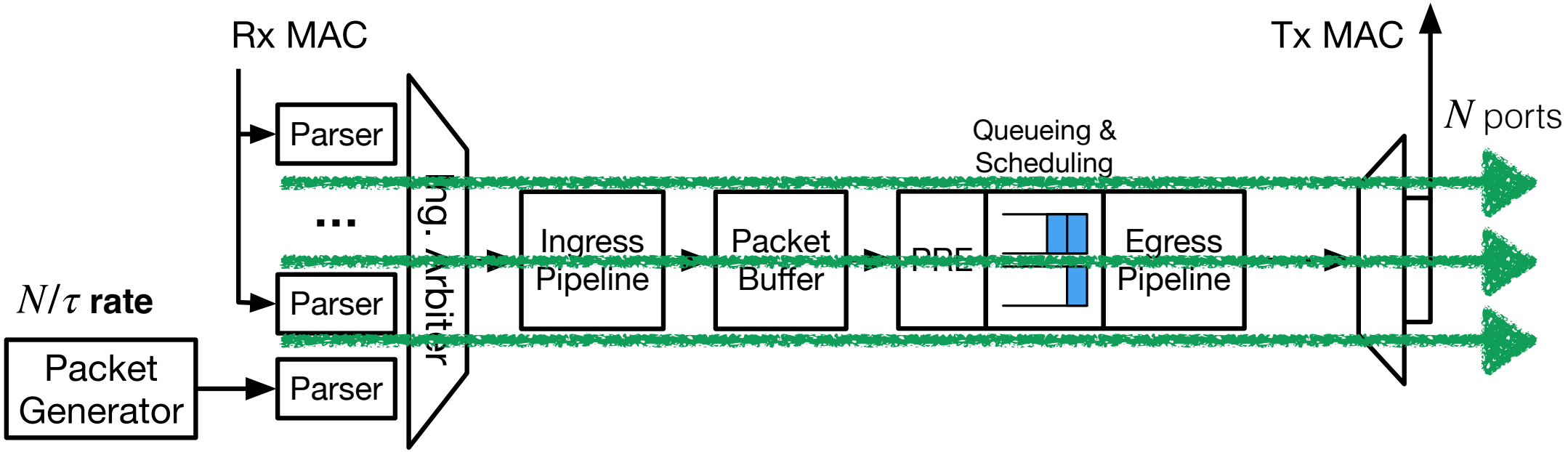
RMT switch model



Naive weaved stream generation

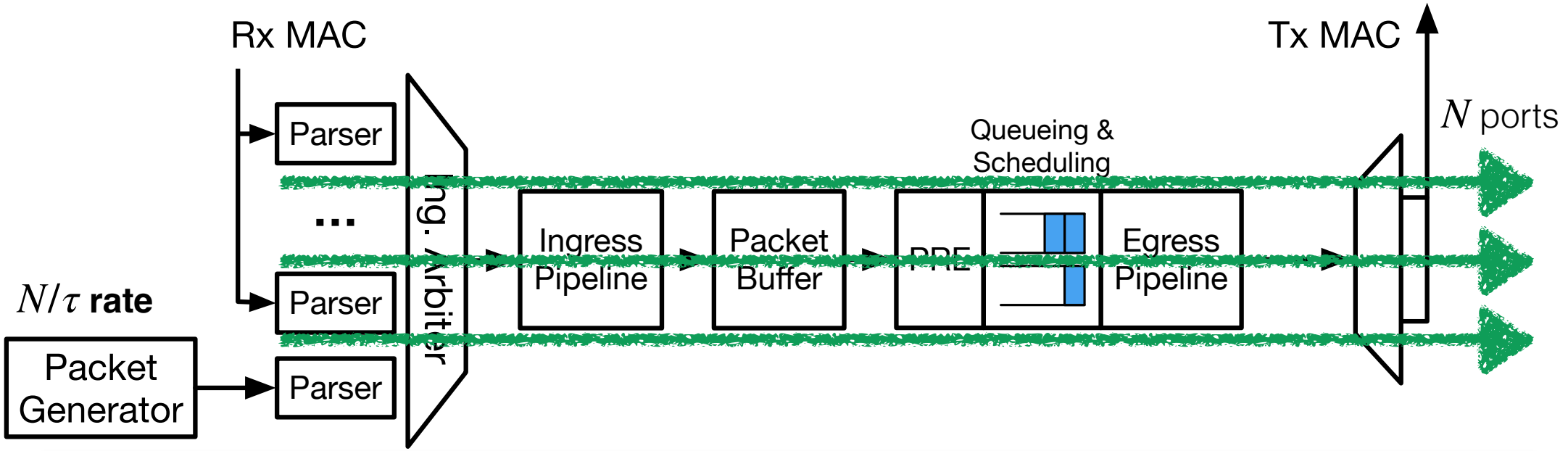


Naive weaved stream generation



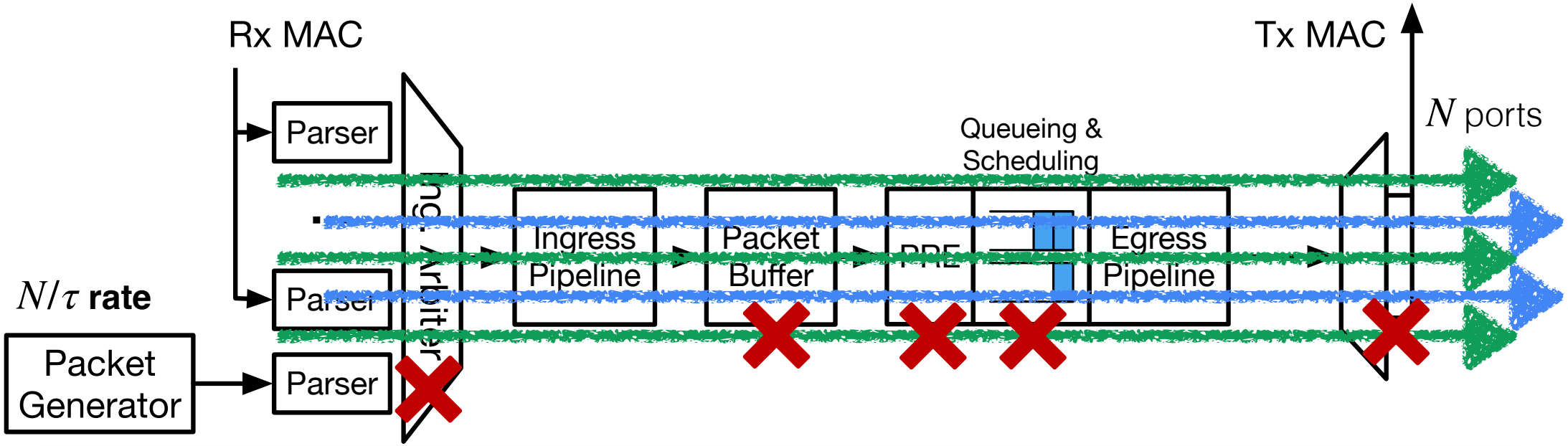
Predictability even there is no user traffic ✅

Problems with blind injection



Scalability: overwhelm packet generator capacity to satisfy target rate

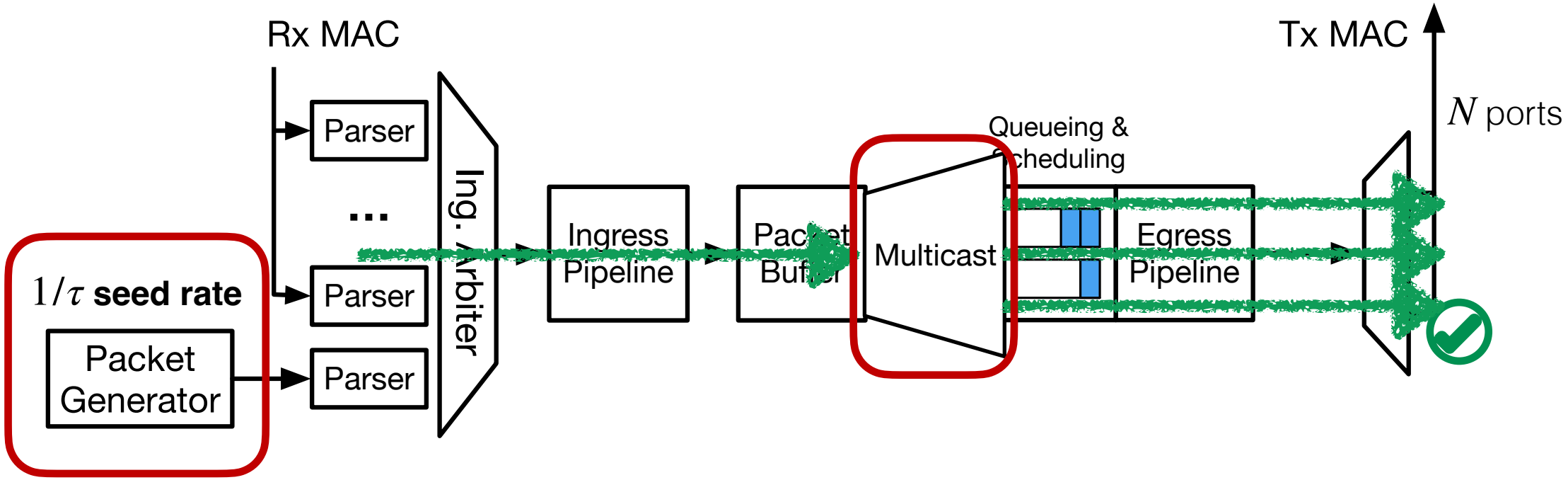
Problems with blind injection



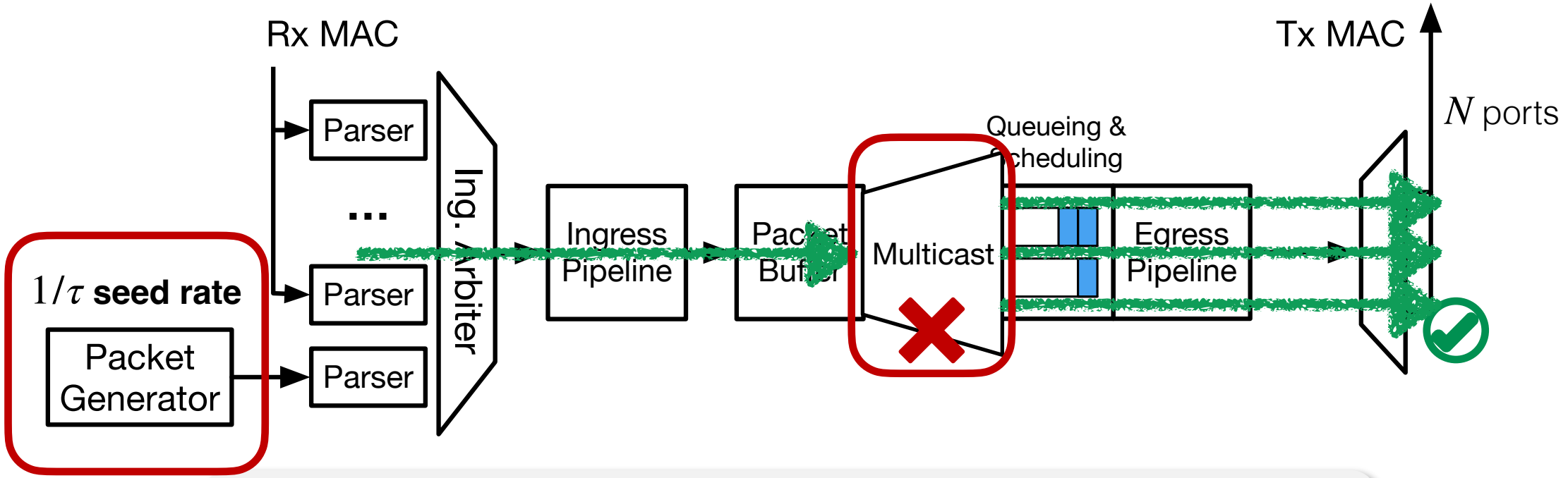
Scalability: overwhelm packet generator capacity to satisfy target rate

Interference upon cross-traffic: throughput, latency, or loss of user traffic!

Amplify seed stream

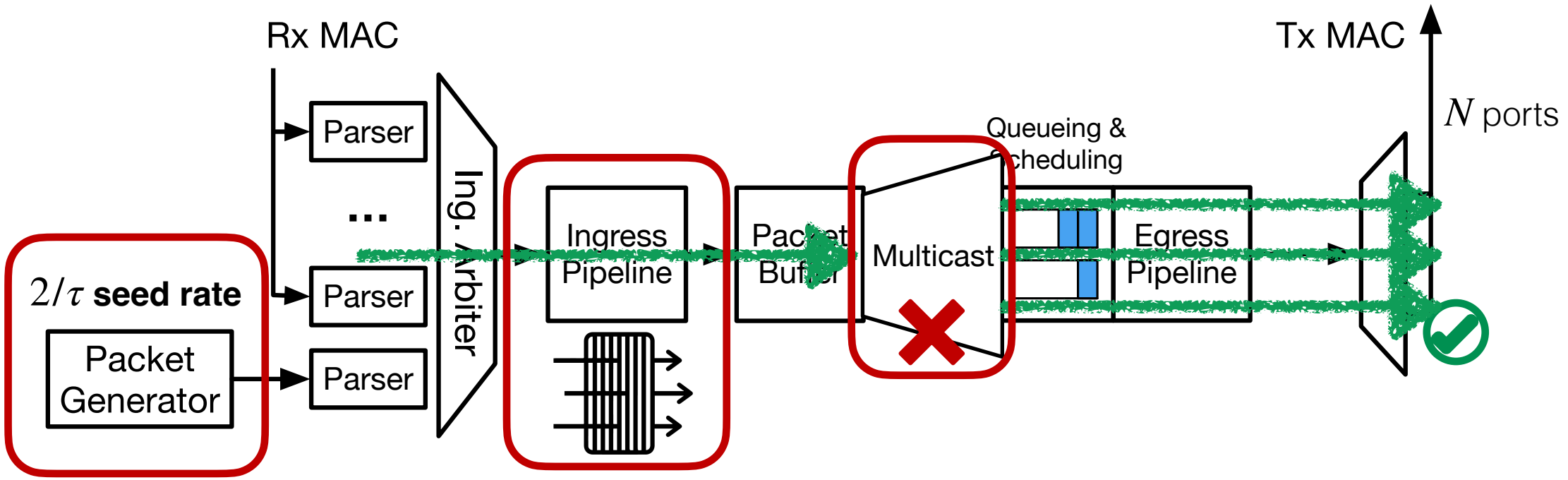


Amplify seed stream



Monopolize usage and waste PRE packet-level BW!

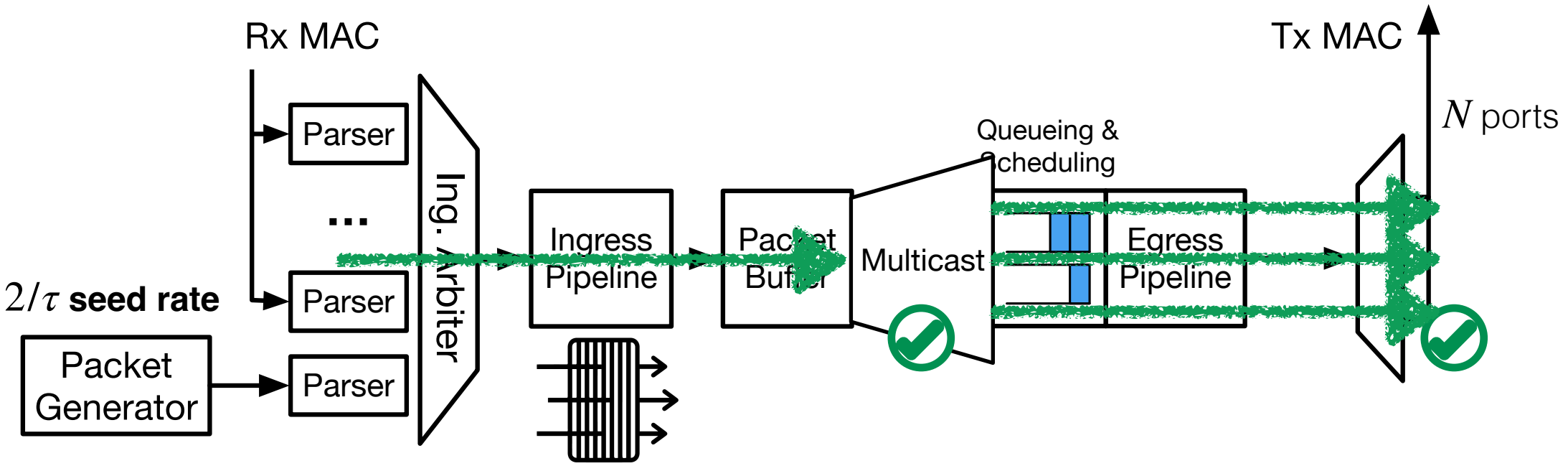
Amplify seed stream on demand



Selective filtering

- (Tiny) sending history state of past cycle to each egress port
- Create an IDLE packet to a port ***only if we need an IDLE packet***

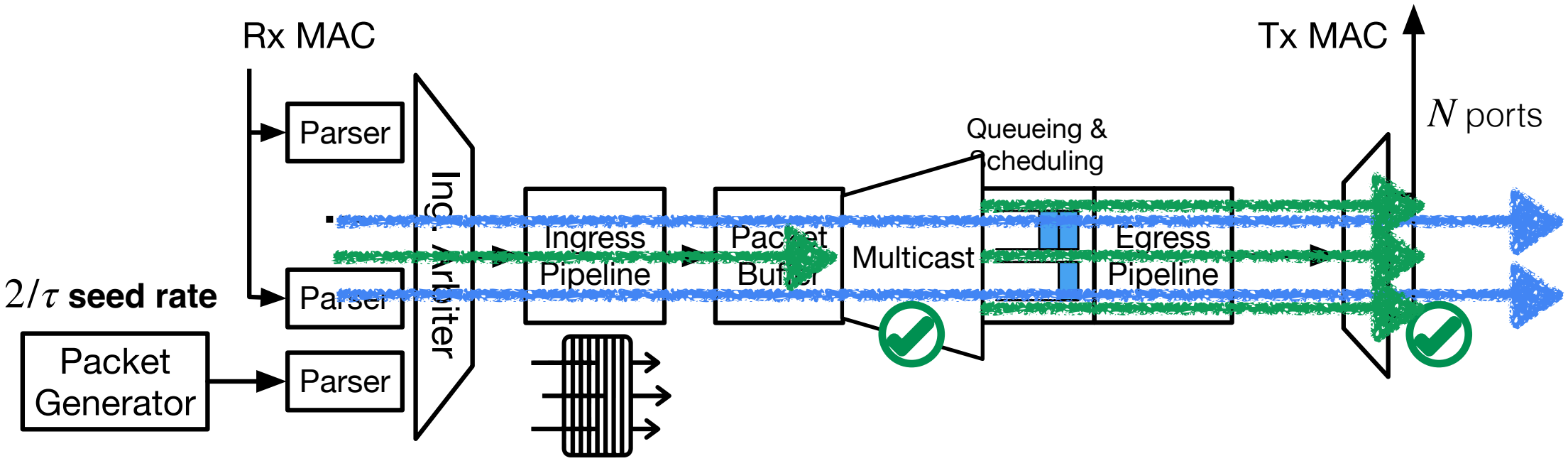
Amplify seed stream on demand



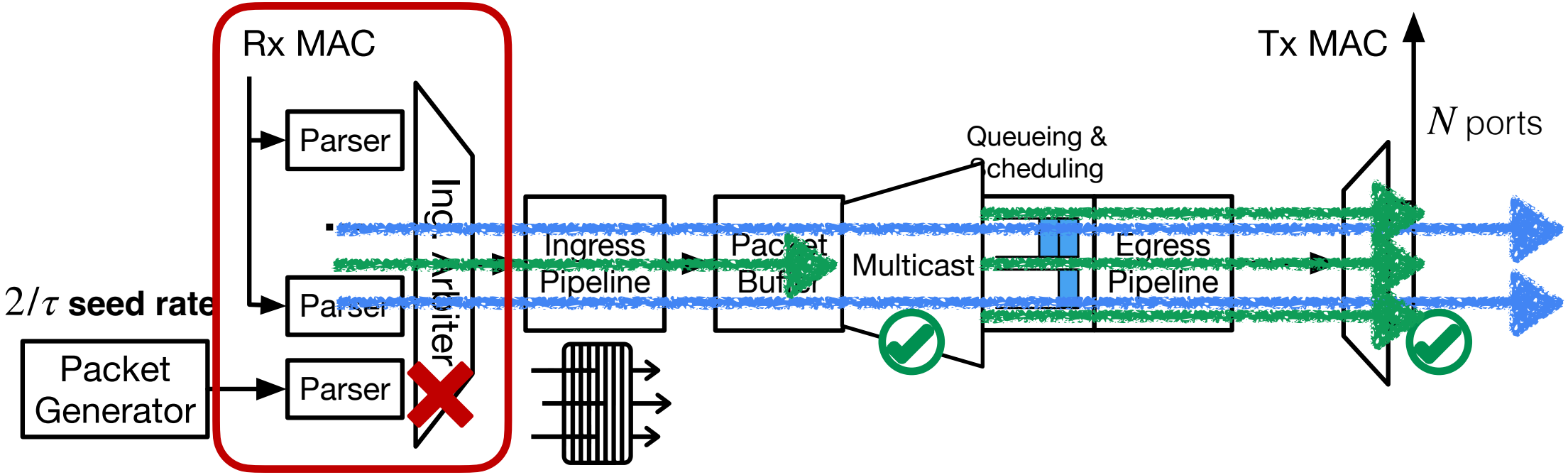
Selective filtering

- (Tiny) sending history state of past cycle to each egress port
- Create an IDLE packet to a port ***only if we need an IDLE packet***

Cross-traffic contention

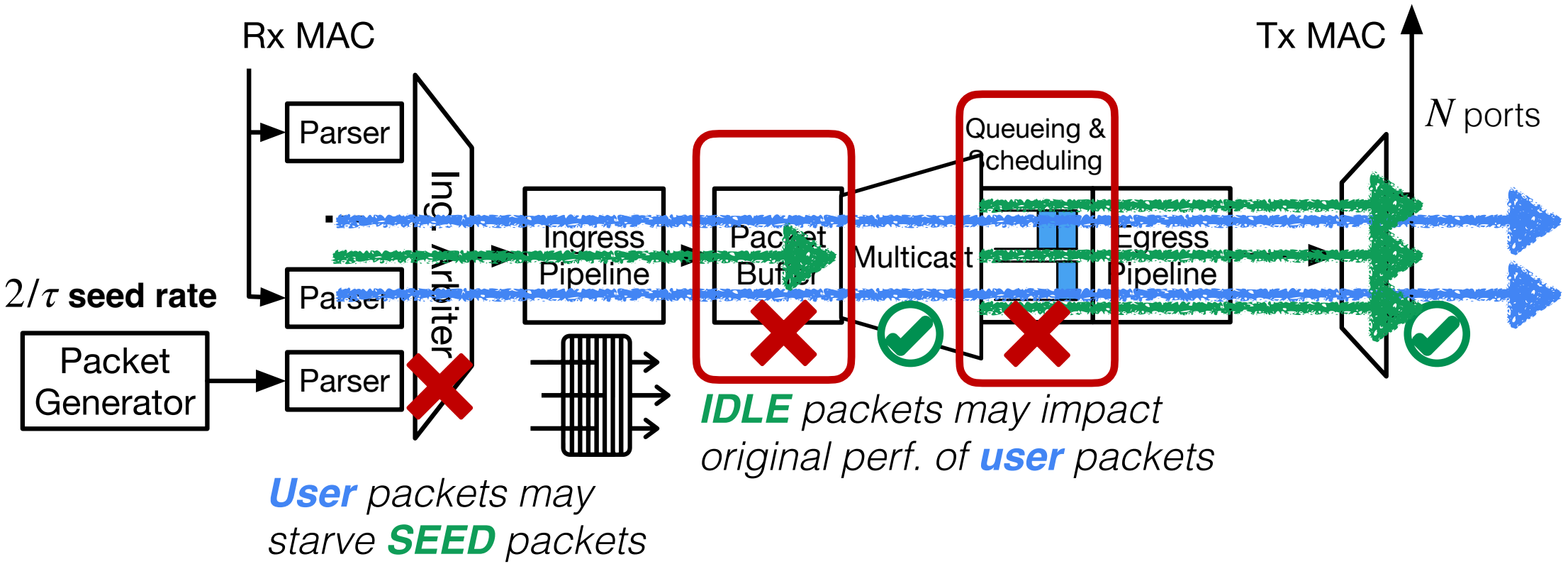


Cross-traffic contention

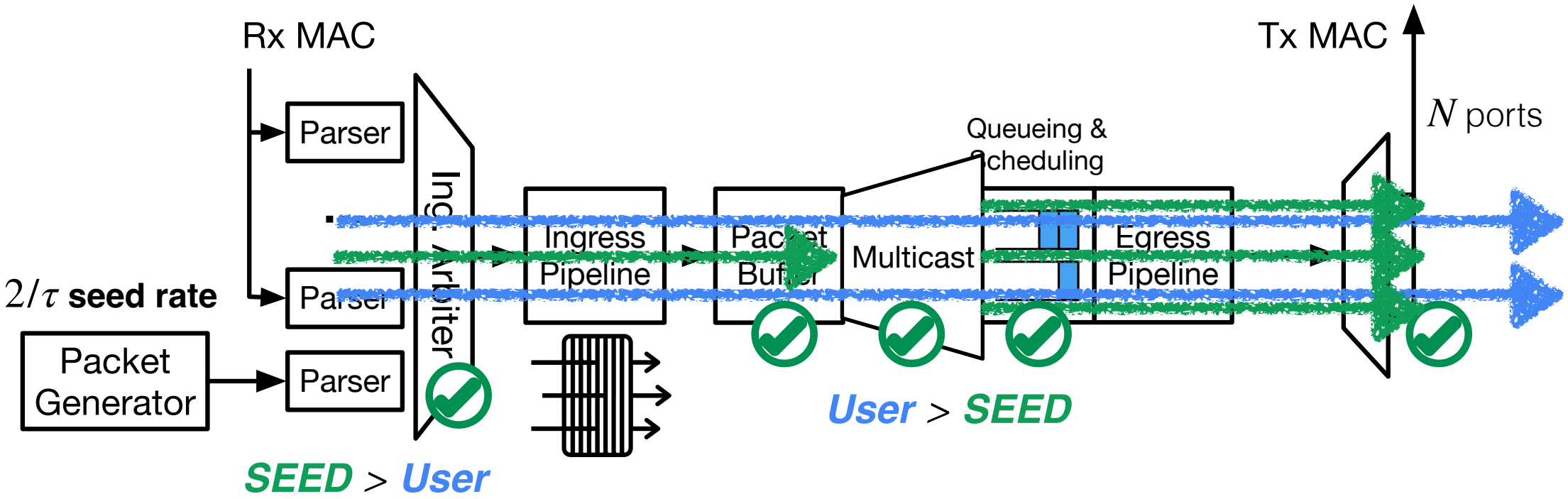


User *packets may starve* ***SEED*** *packets*

Cross-traffic contention



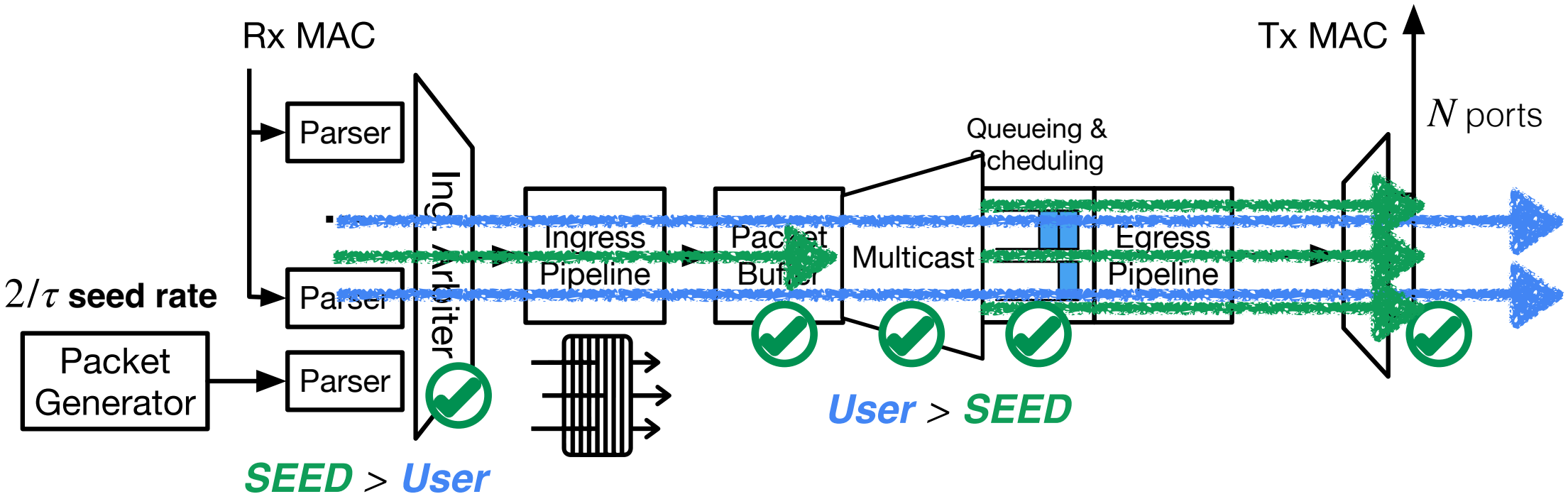
Preventing contention



Rich configuration options for priorities and buffer management

- *Zero impact of weaved stream predictability*
- *Zero impact of **user traffic** throughput or buffer usage*

Preventing contention

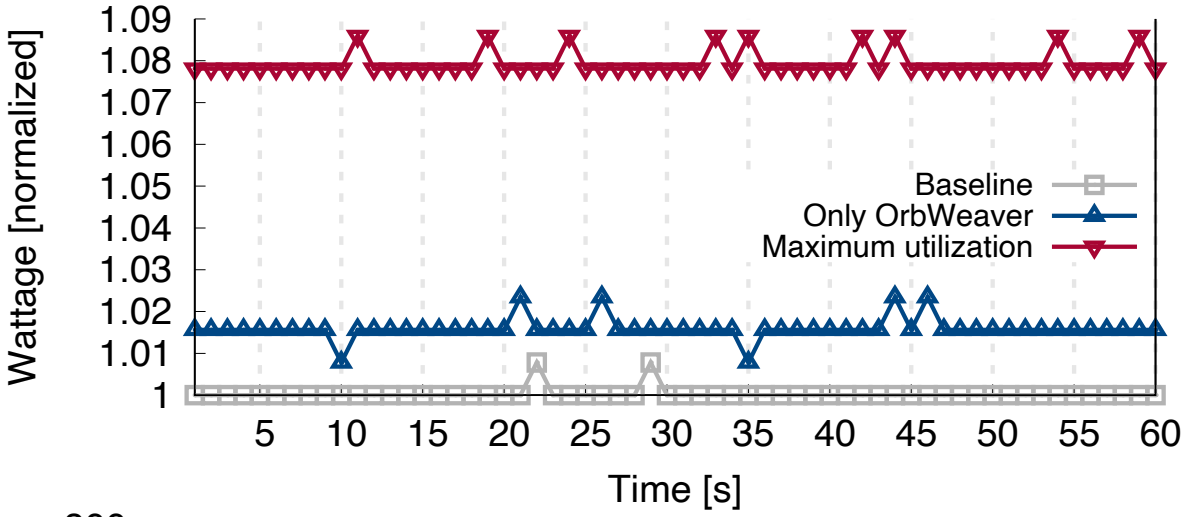


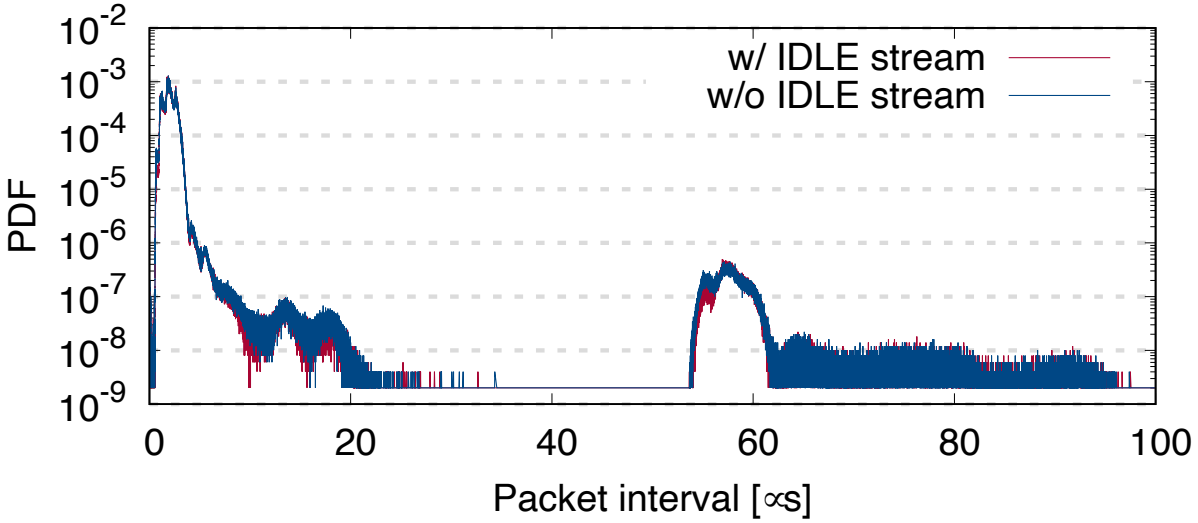
Rich configuration options for priorities and buffer management

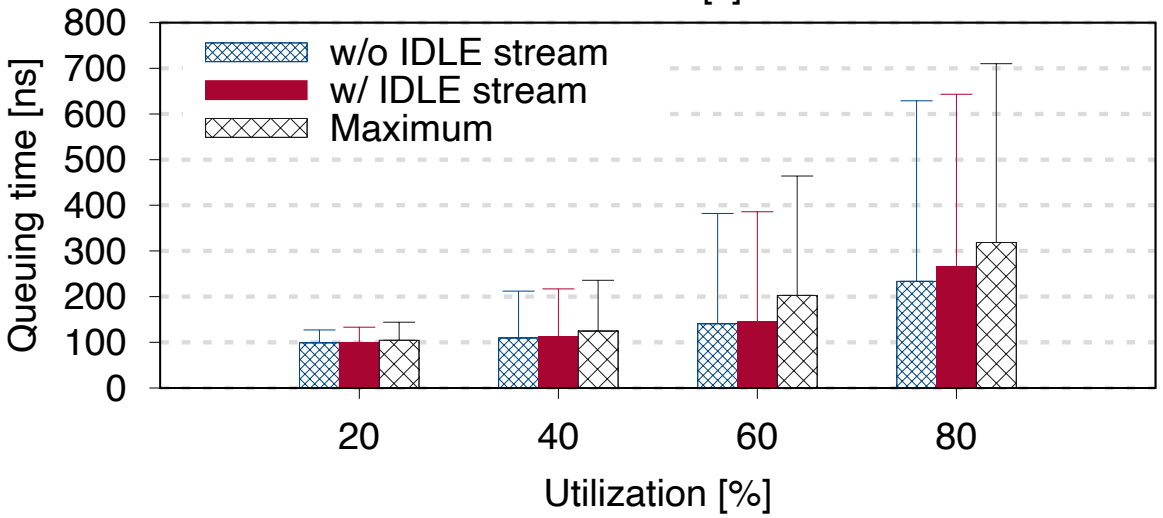
- *Zero impact of weaved stream predictability*
- *Zero impact of **user traffic** throughput or buffer usage*
- *Negligible impact of latency of **user packets***

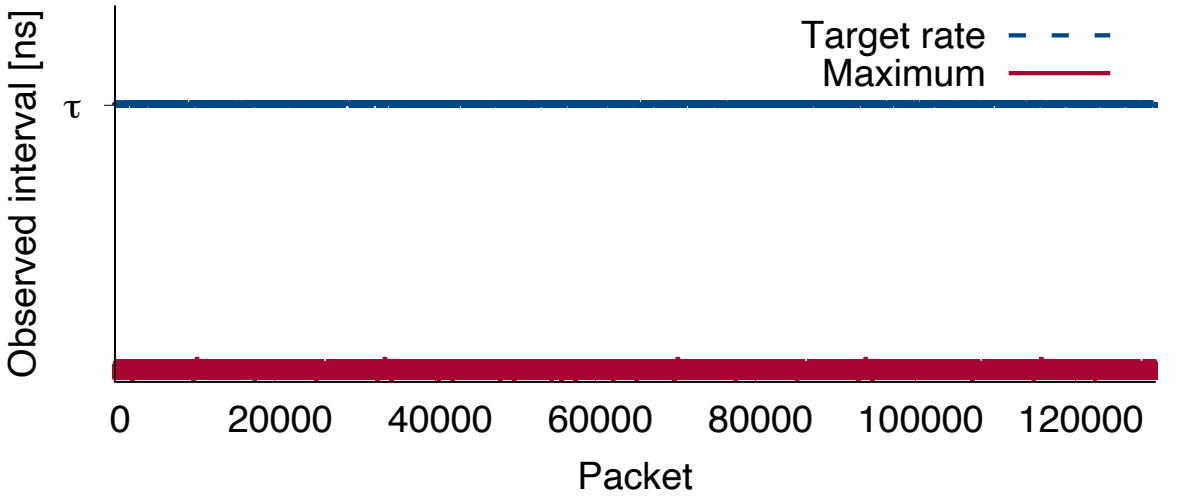
Implementation and evaluation

Hardware prototype on a pair of Wedge100BF-32X Tofino switches



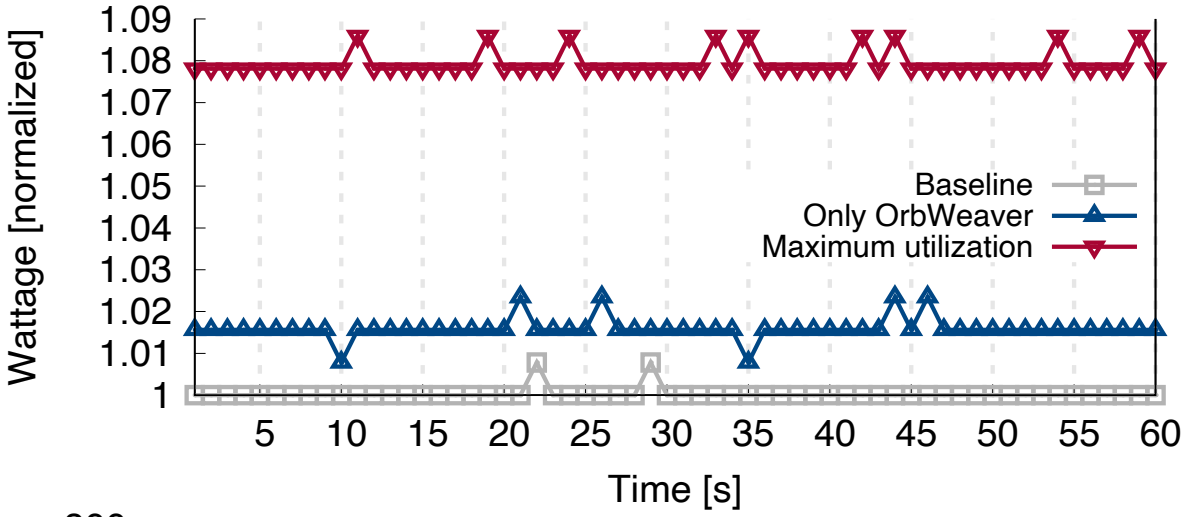


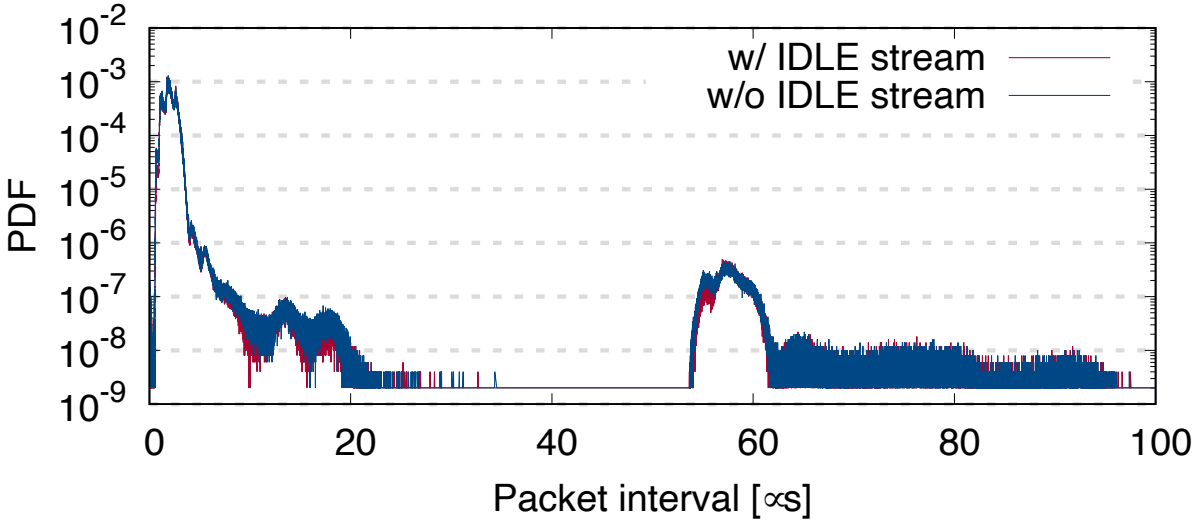


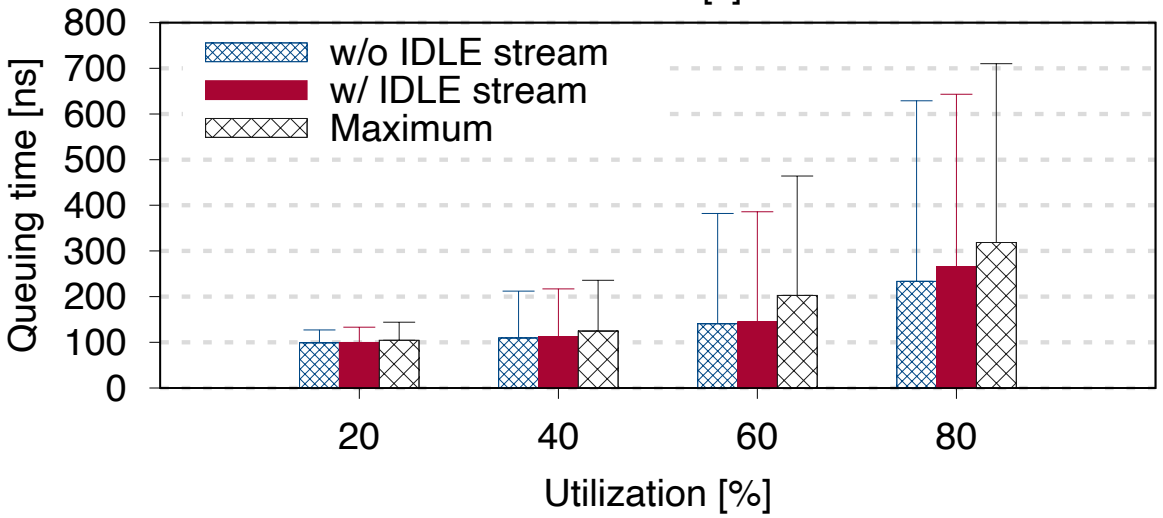


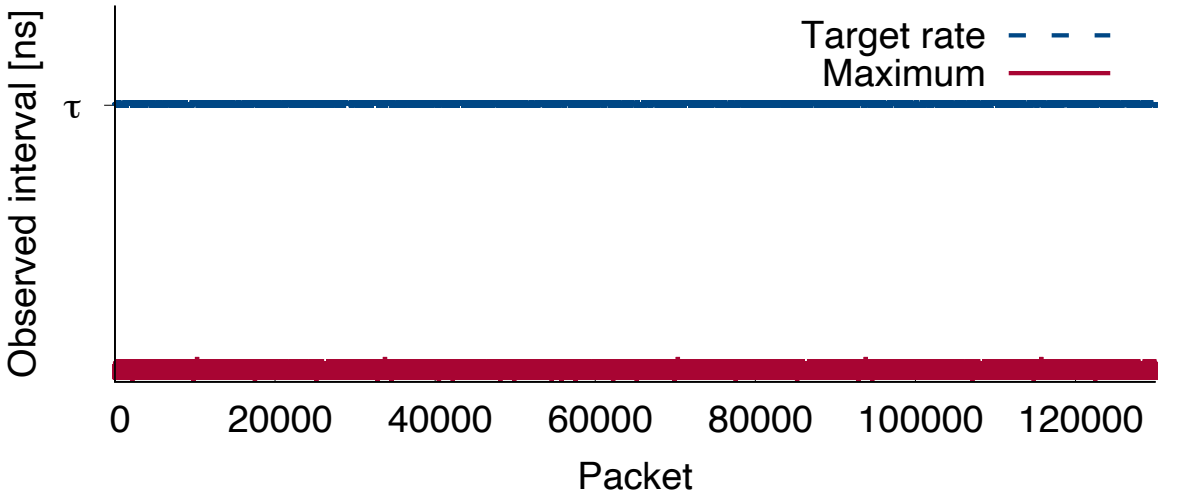
Takeaway: **Little-to-no impact** of power draw, latency, or throughput while guaranteeing **predictability** of the weaved stream!

Hardware prototype on a pair of Wedge100BF-32X Tofino switches









OrbWeaver use cases

Free information dissemination [R2]

Fine-grained network state inference [R1]

Performance aware routing

Flowlet load imbalance

Consistent replicas

Network queries

Latency localization

Header compression

Microburst detection

In-band telemetry

Event-based network control

Failure detection

Network queries

Packet forensics

Clock synchronization

OrbWeaver use cases

Free information dissemination [R2]

Fine-grained network state inference [R1]

Performance aware routing

Flowlet load imbalance

Consistent replicas

Network queries

Latency localization

Header compression

Microburst detection

In-band telemetry

Event-based network control

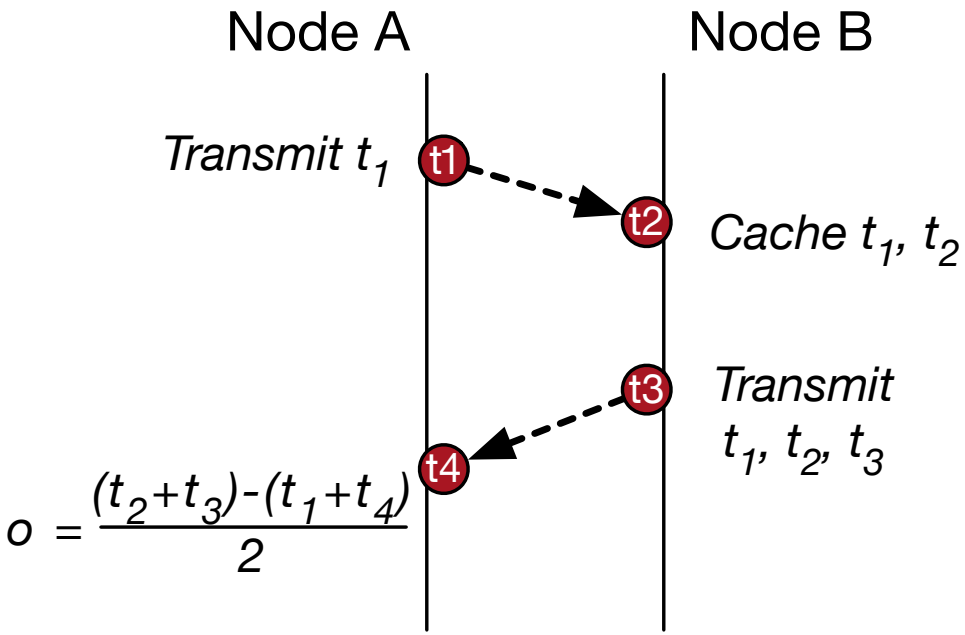
Failure detection

Network queries

Packet forensics

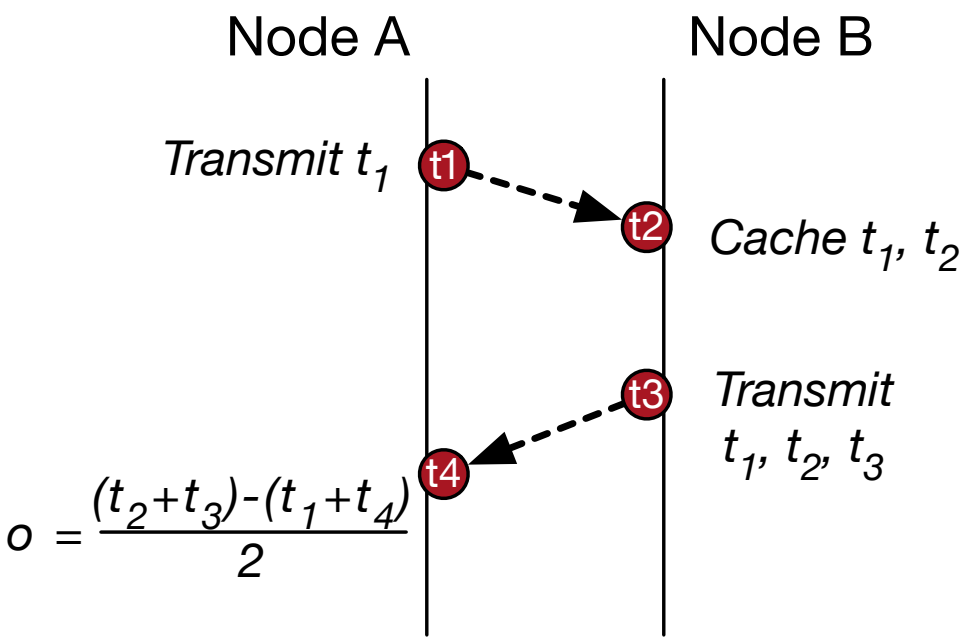
Clock synchronization

Example: time synchronization



Traditional two-way protocol

Example: time synchronization

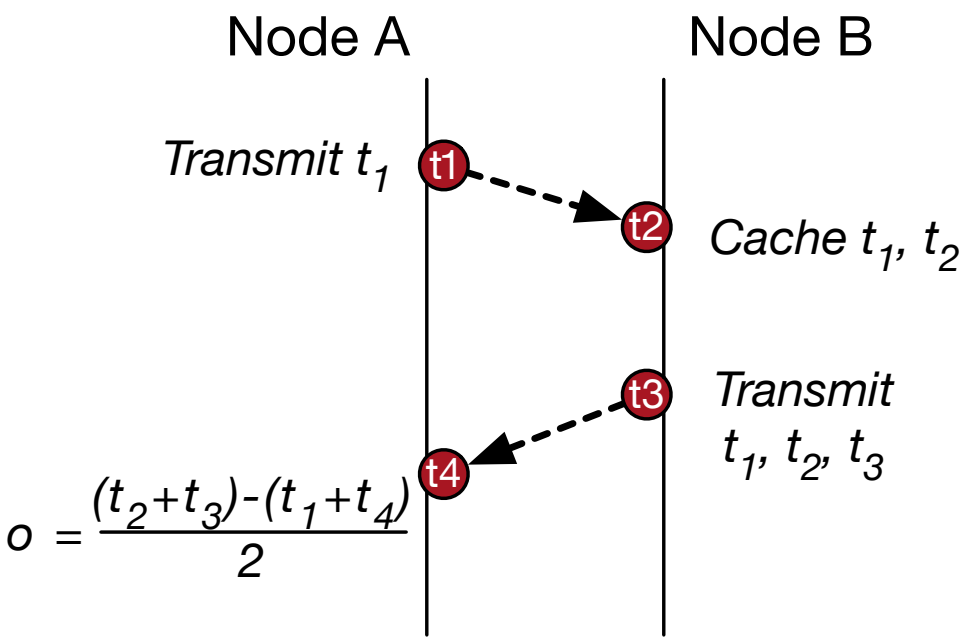


Traditional two-way protocol

Existing approaches for high precision

- Require special hardware (such as DTP)
- Require messaging overheads (such as DPTP)

Example: time synchronization



Traditional two-way protocol

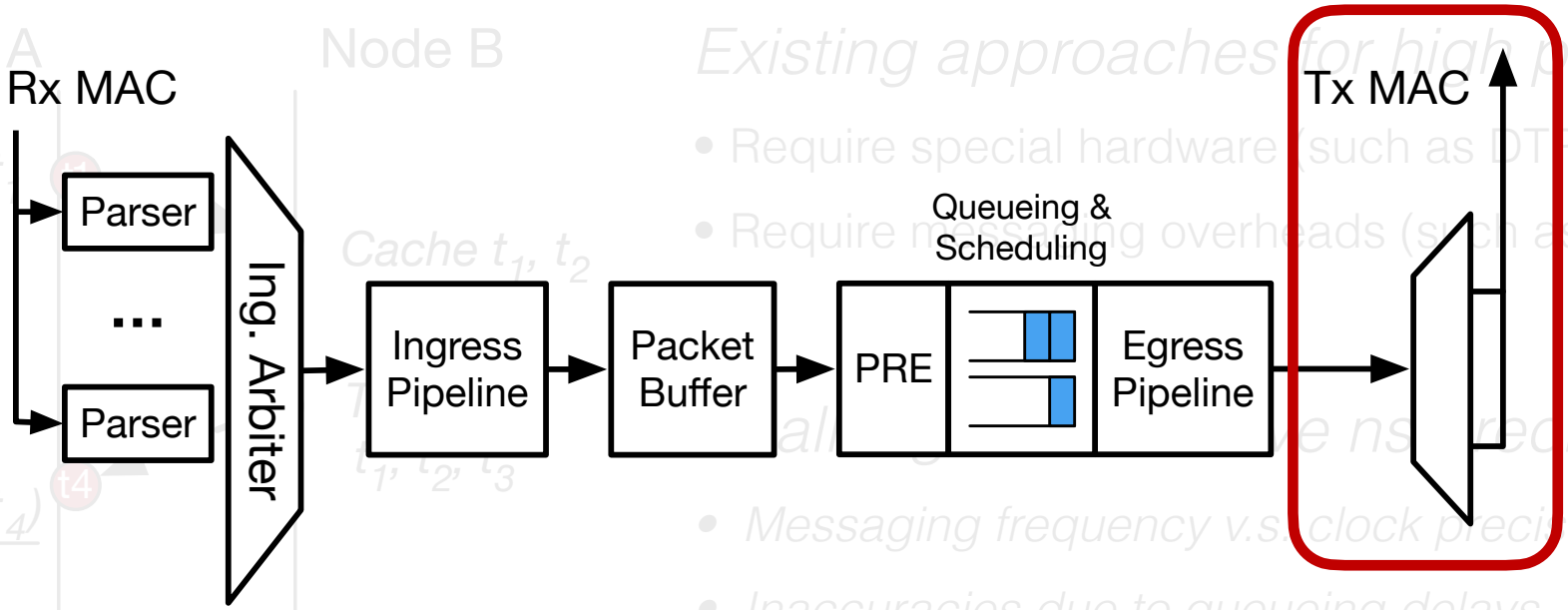
Existing approaches for high precision

- Require special hardware (such as DTP)
- Require messaging overheads (such as DPTP)

Challenges to achieve ns precision

- *Messaging frequency v.s. clock precision*
- *Inaccuracies due to queueing delays*

Example: time synchronization



Data plane timestamps don't capture the actual point of serialization

OrbWeaver Redesign

Key ideas:

1. Embed timestamp information in **free IDLE packets** [R2]

OrbWeaver Redesign

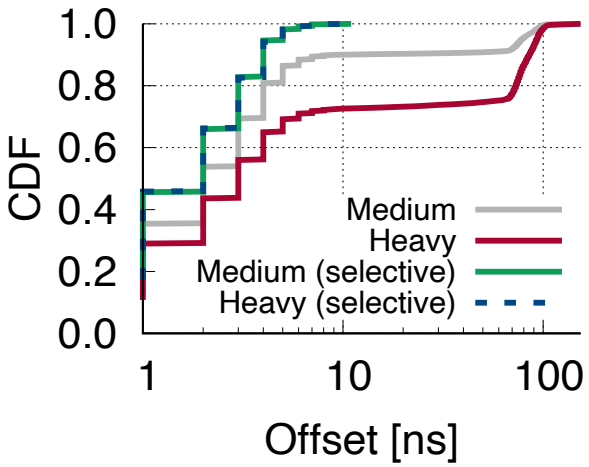
Key ideas:

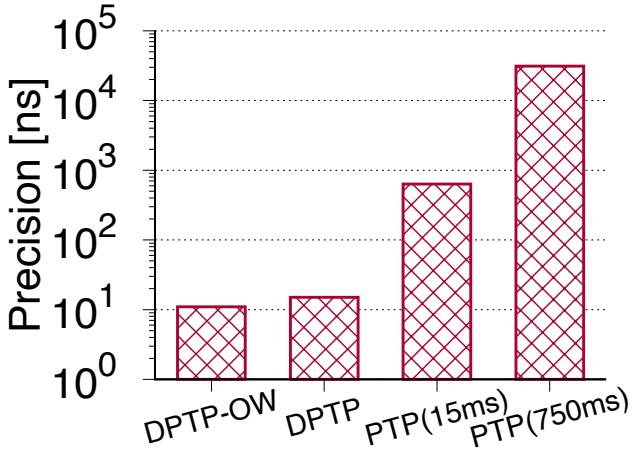
1. Embed timestamp information in **free IDLE packets** [R2]
2. Selective synchronization: **infer queue delay** from IDLE gaps and filter out **unreliable messages** [R1]

OrbWeaver Redesign

Key ideas:

1. Embed timestamp information in **free IDLE packets** [R2]
2. Selective synchronization: **infer queue delay** from IDLE gaps and filter out **unreliable messages** [R1]





Achieve same or better performance with close-to-zero overheads

Summary

- **Weaved stream abstraction** to harvest IDLE cycles
  - Guarantee predictability with little-to-zero overhead

Summary

- **Weaved stream abstraction** to harvest IDLE cycles
  - Guarantee predictability with little-to-zero overhead
- Generic support of a wide range of data plane applications for free
  - ***Don't*** need to choose between coordination fidelity and bandwidth overhead

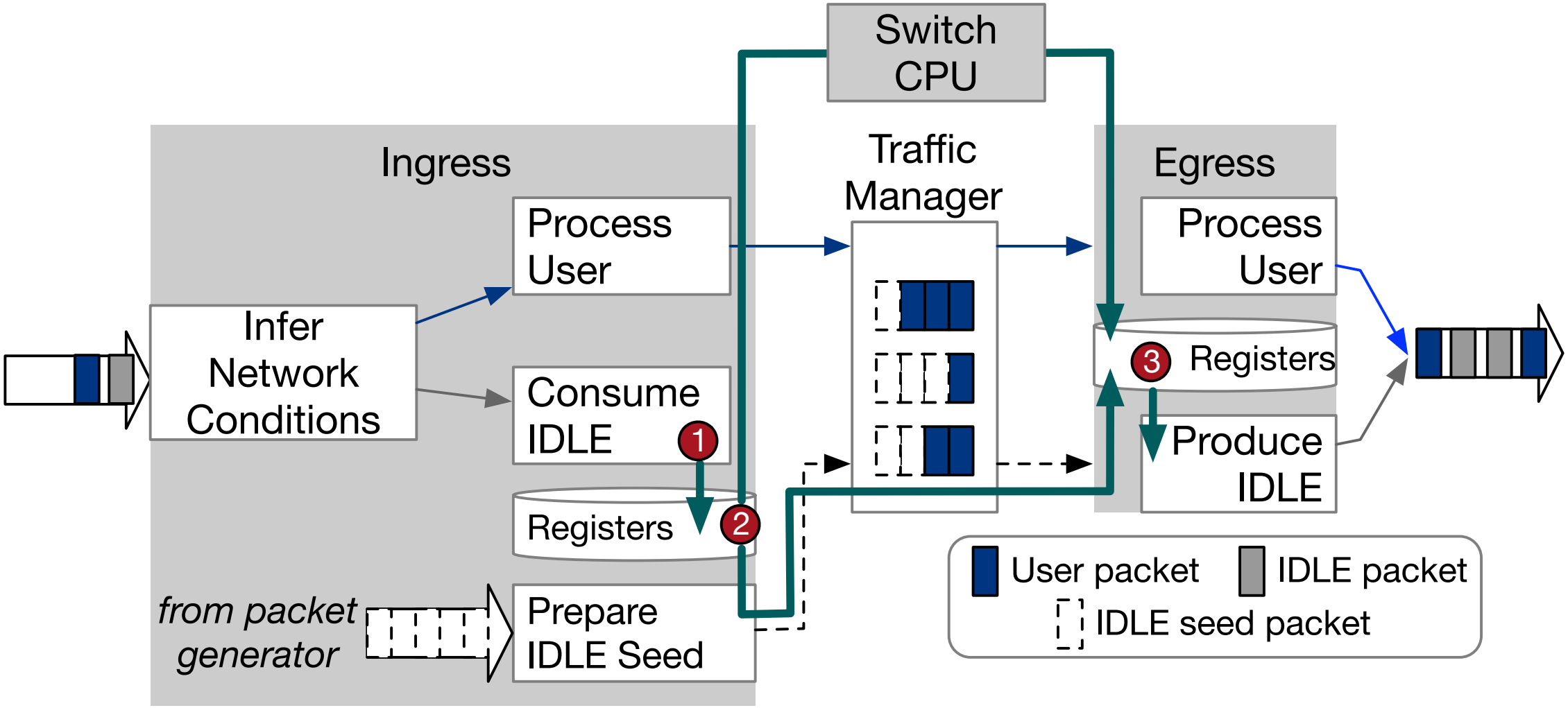
https://github.com/eniac/OrbWeaver

Thank you for your attention!

Q & A

Backup Slides

Using weaved stream



Optimal value of τ

$$\tau = B_{100Gbps}/MTU_{1500B} = 120ns$$