

# Progress Report: Make NDN Congestion Control work in ndnSIM

6th NDN Hackathon

---

Klaus Schneider, Ashiqur Rahman, Chavoosh Ghasemi

May 13, 2018

The University of Arizona

# Motivation

Congestion Control crucial for high-performance simulations

# Motivation

Congestion Control crucial for high-performance simulations

- NFD congestion detection doesn't work in ndnSIM  
(no real TCP/UDP/Unix faces)

# Motivation

Congestion Control crucial for high-performance simulations

- NFD congestion detection doesn't work in ndnSIM  
(no real TCP/UDP/Unix faces)

⇒ Task: Fix that.

# Solution Steps

1. **ndnSIM doesn't use real TCP or UDP faces.**  
⇒ NetDeviceTransport: override virtual function(s) for congestion control (Junxiao's solution).

# Solution Steps

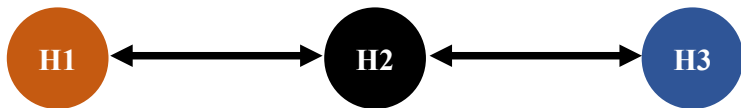
1. **ndnSIM doesn't use real TCP or UDP faces.**  
⇒ NetDeviceTransport: override virtual function(s) for congestion control (Junxiao's solution).
2. **CongestionMarks sent over NDNLP:** Already works!

# Solution Steps

1. **ndnSIM doesn't use real TCP or UDP faces.**  
⇒ NetDeviceTransport: override virtual function(s) for congestion control (Junxiao's solution).
2. **CongestionMarks sent over NDNLP:** Already works!
3. **Implement Consumer App** that reacts to congestion marks (AIMD and TCP CUBIC)

# Evaluation Scenario

Very simple scenario:



- 1 Consumer, Runtime: 40s
- RTT: 40ms
- Bottleneck capacity: 50 Mbit/s



# How ndnSIM performs right now

ConsumerWindow App:

- On Data:  $m_{cwnd}++$

# How ndnSIM performs right now

ConsumerWindow App:

- On Data:  $m_{cwnd}++$  (exp. increase / constant slow start!)

# How ndnSIM performs right now

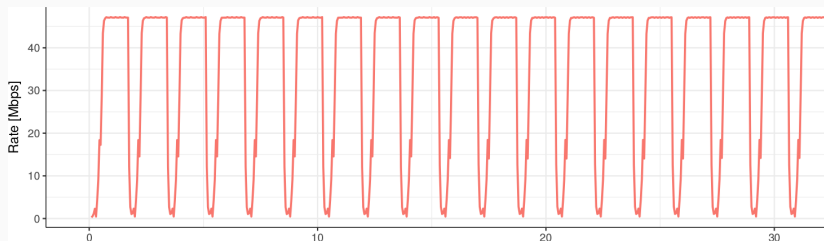
## ConsumerWindow App:

- On Data:  $m_{cwnd}++$  (exp. increase / constant slow start!)
- On Timeout:  $m_{cwnd} \leftarrow INITIAL\_CWND$  (2)

# How ndnSIM performs right now

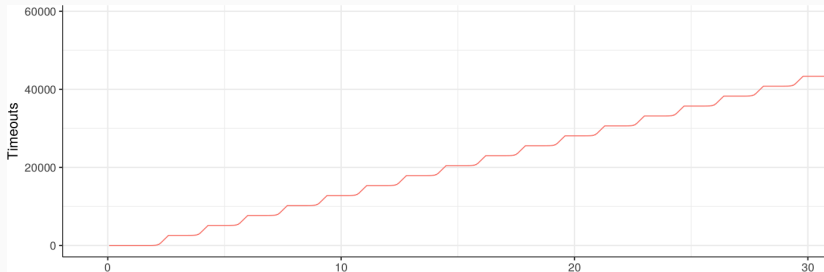
## ConsumerWindow App:

- On Data:  $m_{cwnd}++$  (exp. increase / constant slow start!)
- On Timeout:  $m_{cwnd} \leftarrow INITIAL\_CWND$  (2)

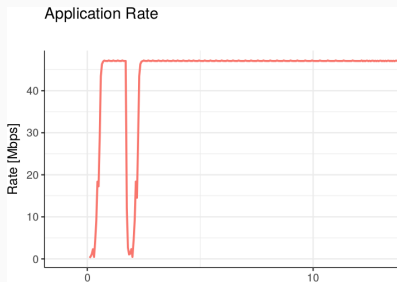


# How ndnSIM performs right now

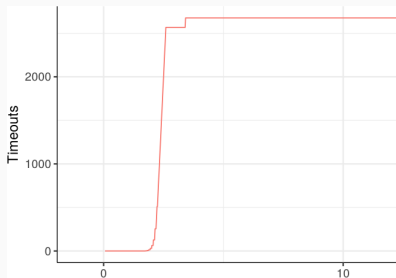
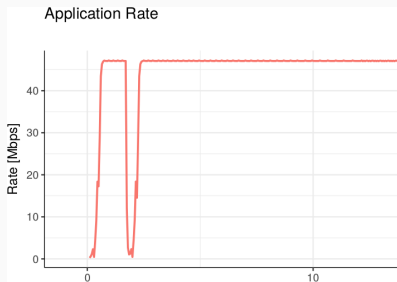
40,000 Timeouts!!!



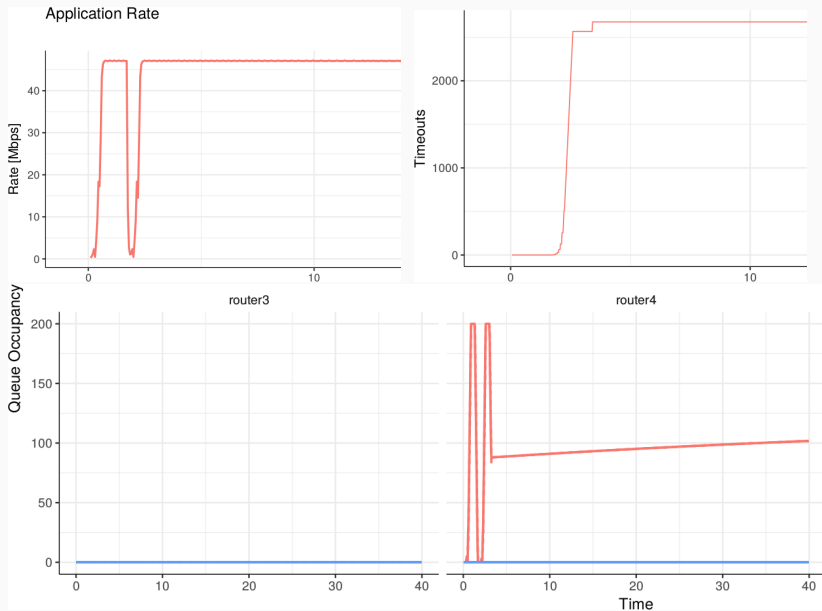
# Improved ConsumerWindow (no congestion marks)



# Improved ConsumerWindow (no congestion marks)

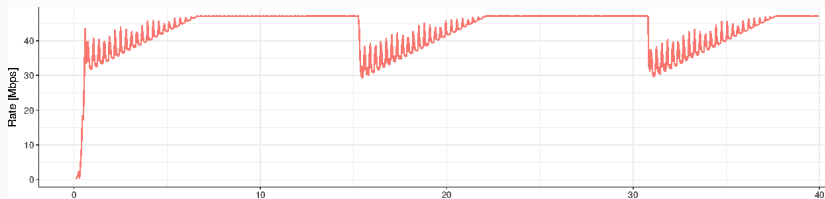


# Improved ConsumerWindow (no congestion marks)



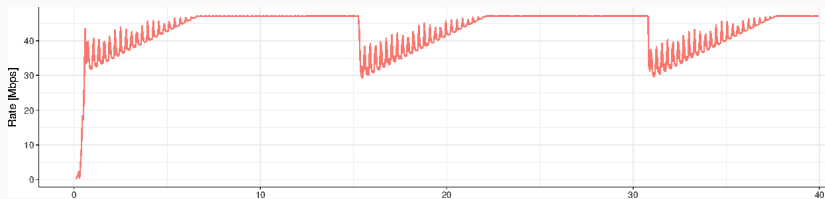


# ConsumerPCON – AIMD

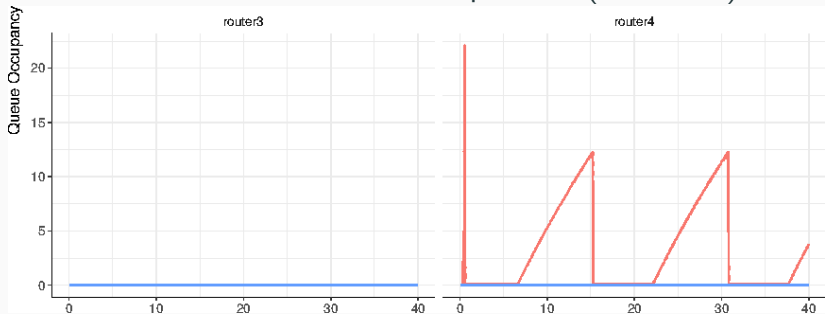


No Timeouts!, Shorter “sawtooth” patterns (about 15s)

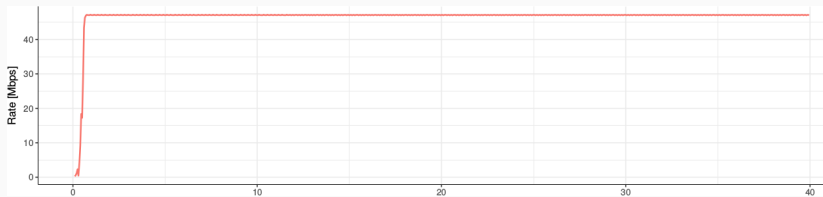
# ConsumerPCON – AIMD



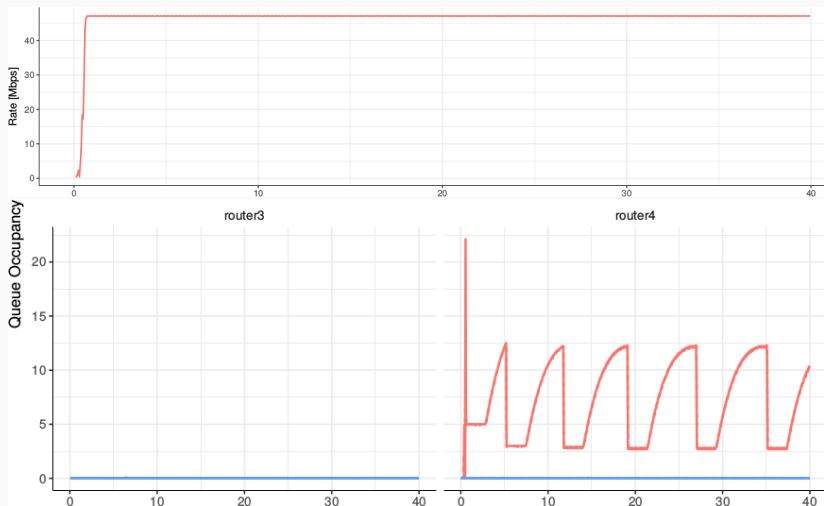
No Timeouts!, Shorter “sawtooth” patterns (about 15s)



# ConsumerPCON – CUBIC



# ConsumerPCON – CUBIC



Even Shorter Sawtooths (about 7s)!

## Future Work

Congestion Detection via **adapted CoDelQueue**:

- Worked well in ns3 3.23 (PCON simulation code)

# Future Work

## Congestion Detection via **adapted CoDelQueue**:

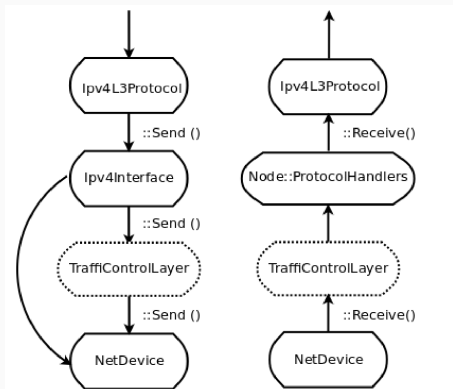
- Worked well in ns3 3.23 (PCON simulation code)
- Doesn't work anymore in ns3 3.27 (current ndnSIM)

# Future Work

## Congestion Detection via **adapted CoDelQueue**:

- Worked well in ns3 3.23 (PCON simulation code)
- Doesn't work anymore in ns3 3.27 (current ndnSIM)

NS-3 separated queuing in traffic-control module as:



# Summary

Current ndnSIM consumer apps very limited!

⇒ **Works much better now!**



# Summary

Current ndnSIM consumer apps very limited!

⇒ **Works much better now!**

Mechanisms:

1. Slow start + Congestion avoidance

# Summary

Current ndnSIM consumer apps very limited!

⇒ **Works much better now!**

Mechanisms:

1. Slow start + Congestion avoidance
2. Fast recovery + Conservative Window Adaptation

# Summary

Current ndnSIM consumer apps very limited!

⇒ **Works much better now!**

Mechanisms:

1. Slow start + Congestion avoidance
2. Fast recovery + Conservative Window Adaptation
3. **Explicit Congestion Marks**

# Summary

Current ndnSIM consumer apps very limited!

⇒ **Works much better now!**

Mechanisms:

1. Slow start + Congestion avoidance
2. Fast recovery + Conservative Window Adaptation
3. **Explicit Congestion Marks**
4. **CUBIC > AIMD**

# Summary

Current ndnSIM consumer apps very limited!

⇒ **Works much better now!**

Mechanisms:

1. Slow start + Congestion avoidance
2. Fast recovery + Conservative Window Adaptation
3. **Explicit Congestion Marks**
4. **CUBIC > AIMD**

<https://github.com/6th-ndn-hackathon/congestion-control>

# Any Questions?

Klaus Schneider, Ashiqur Rahman, Chavoosh Ghasemi