

NS-3 Implementation of ABB CoDel Queue Discipline

Gongbing Hong
Georgia College and State University

Outline

- Quick review of **existing queue disciplines** in ns-3
- **Why another** queue discipline?
- **ABB CoDel** and its implementation in ns-3
- Preliminary result
- **Experience and challenges** of ABB CoDel implementation in ns-3
- Future work



Review of existing queue disciplines in ns-3

- **Single-queue** queue disciplines (qdisc's)
 - Non-AQM: Fifo/pfifo-fast
 - **AQM**: RED/CoDel/PIE
- **Multi-queue** qdisc's for fair queueing
 - Fq-CoDel (by default, 1024 internal queues)
 - Fq-Pie (by default, 1024 internal queues)



Why another queue discipline?

- **Bandwidth allocation fairness** considerations (especially weighted fairness)
 - Single-queue qdisc's can **hardly** provide fair bandwidth allocation
 - Fq-CoDel / Fq-Pie does not provide **weighted** fair bandwidth allocation
- **Low delay**
 - Buffer bloat is a huge problem that has been recognized and dealt with
 - AQM qdisc's such as CoDel and PIE provide low delay while others do not
- Implementation **complexity**
 - Fair queueing disciplines such as WFQ often use **many internal queues – not scalable**
 - **Work required per packet** is high – $O(n)$ or $O(\log n)$



ABB CoDel ideas

- ABB stands for **A**daptive **B**andwidth **B**inning
 - Previous implemented in ns-2 and evaluated under DOCSIS 3.x
- ABB **approximates fair queueing** by:
 - Using only a few (e.g. 3~5) **internal queues** (called **bins**) for low implementation complexity
 - Using CoDel to manage queueing delays for each bin
 - **Periodically moving flows from one bin to another** depending on **flow service rates**:
 - Each flow can have a **flow weight** for bandwidth allocation
 - **Flows with similar normalized service rates are classified into the same bin**
 - Flows with lowest normalized services rates are classified into first bin
 - Flows with highest normalized services rates are classified into last bin



ABB CoDel implementation in ns-3

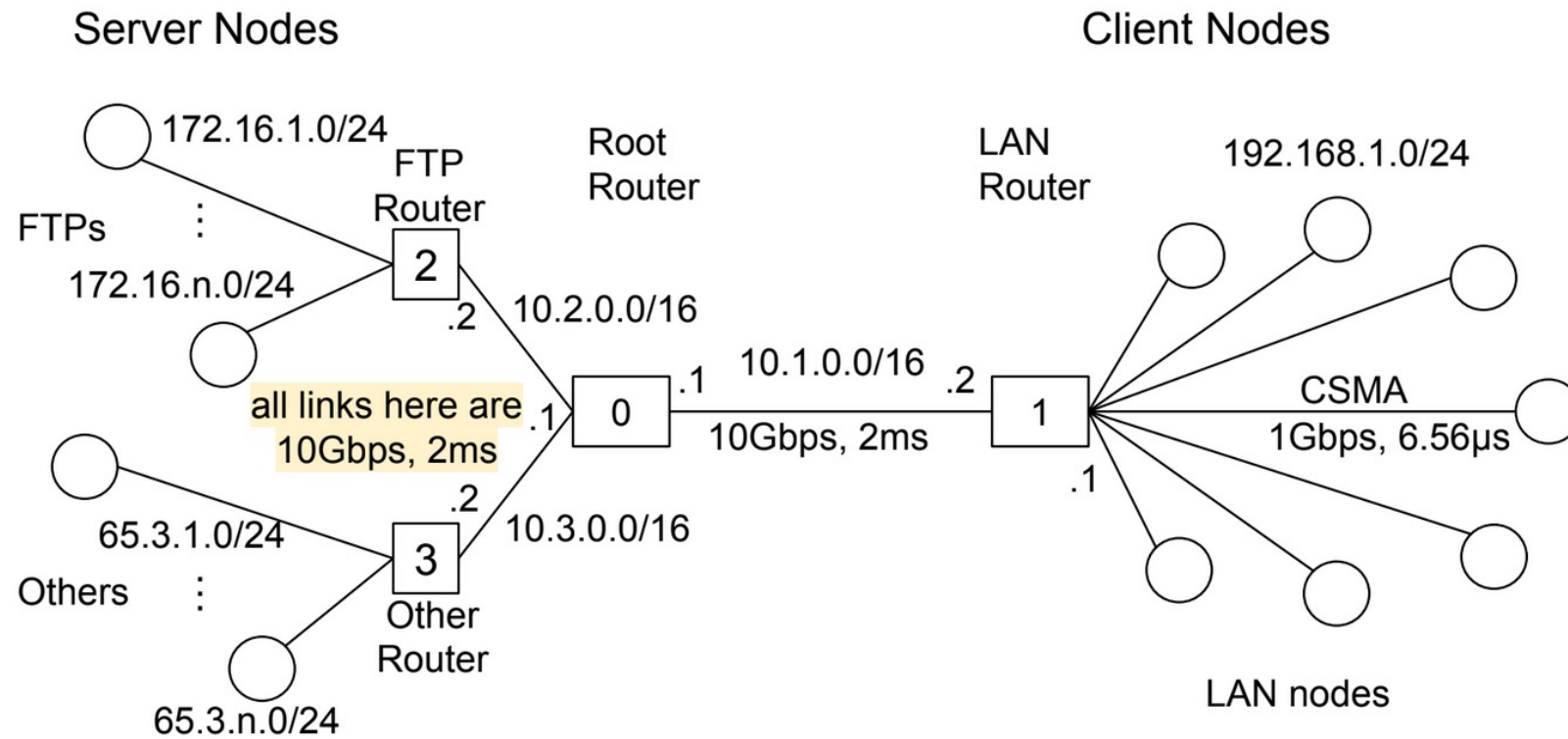
Fq-CoDel

- FqCoDelQueueDisc subclasses QueueDisc by implementing following virtual methods:
 - DoEnqueue/DoDequeue
 - CheckConfig/InitializeParams
- FqCoDelFlow subclasses QueueDiscClass by adding methods:
 - Get/Set/IncreaseDeficit
 - Get/SetStatus
 - Get/SetIndex

ABB CoDel

- **Modeled after Fq-CoDel** by subclassing QueueDisc and QueueDiscClass
- Implementing all methods on the left
- ABBCoDelQueueDisc added methods:
 - Optimize
 - GetFlowInfo (using the callbacks below)
 - Callbacks: flowIdCb & flowWeightCb
- ABBCoDelBin added methods:
 - Get/SetBandwidthThreshold
 - Get/Set/IncreaseWeight

Preliminary result



Simulated network topology



Preliminary result

- Configuration:
 - Five **FTP-type of flows** sent to five LAN nodes through a LAN router
 - Results to be compared with LAN router runs Fifo, FqCoDel, and ABBCoDel
 - Simulation time: 100s and 1000s
 - ABBCoDel flow **reclassification interval**: 1s



Preliminary result

Bandwidth Allocation (100s)

Flow	FIFO	FqCoDel	ABBCoDel
1	198.93	142.71	119.06
2	94.47	147.72	167.48
3	151.32	143.32	113.83
4	89.81	144.38	147.61
5	169.02	140.76	160.17
JFI	0.9164	0.9997	0.9773
MMR	0.4515	0.9529	0.6797

For performance comparison, LAN router network device is set to use FIFO, Fq-CoDel, and ABBCoDel respectively.

All other network devices are set to use **pfifo-fast**.



Preliminary result

Bandwidth Allocation (1000s)

Flow	FIFO	FqCoDel	ABBCoDel
1	126.91	145.14	133.02
2	114.76	142.86	149.08
3	163.63	142.28	146.71
4	137.47	144.88	134.23
5	162.61	144.47	149.43
JFI	0.9815	0.9999	0.9974
MMR	0.7013	0.9803	0.8902



Experience and challenges of ns-3 implementation

- ns-3
 - A **very nice platform** for writing code, but is also a huge “beast” to tame
 - Went through **several iterations** to get it right
 - **Flow monitor**: a big saver for getting performance metrics for my network simulation
 - The **logging system** is great in helping me with the debugging
- Questions for the experts and wishes
 - ns-3 is **slow**: wish it could be **faster**
 - **Too much log**: can we ask for **the logs from only the last few seconds** before crashing?
 - **Default qdisc** has been changed from **pfifo-fast** to **FqCoDel**:
 - This changes everything for us to do qdisc research: Is it the right move?



Future work

- Tiering support
 - It is the common practice that an ISP provides several service tiers to its subscribers in exchange for different charges
- Better approximation
 - By adjusting bin weights?



Questions?

