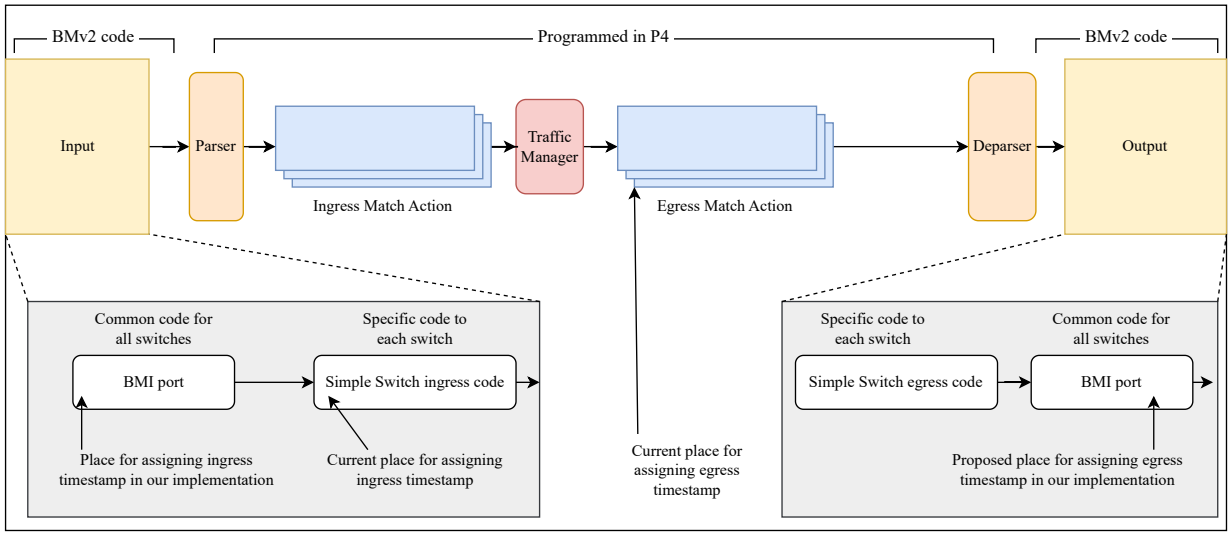
Supporting PTP-1588 in BMv2: A Proposed Ingress and Egress Timestamping Scheme

Atabak Nojavan, François-Raymond Boyer, Bill Pontikakis, Yvon Savaria, Polytechnique Montréal

Motivation: We wanted to test our modular P4 design for the eCPRI 5G fronthaul protocol, which uses PTP synchronization. PTP requires accurate timestamping to ensure precise timing and synchronization across network elements. However, the BMv2 switch does not natively support precise timestamps required for protocols like PTP-1588. This limitation necessitates modification to the BMv2 switch for accurate ingress and egress timestamping needed for precise timing protocols. This proposal will be useful for the industry, enabling them to test their designs rigorously before implementing them in hardware.

Proposed Solution: To address the shortcomings of the current BMv2 implementation, we integrated timestamping directly into the behavioral model port to capture ingress timestamps at the earliest possible point in the packet processing pipeline. We used our enhanced port capability in Simple Switch. The switch's architecture further complicates matters as only packet data are recorded at the BMI port, the initial packet handling point, with timestamps added later within specific switch logic, leading to delayed and inconsistent measurements. To facilitate this, the BMI port code was converted from C to C++ and restructured using input structs for function handlers, making it easier to add other metadata to input packets. This restructuring eventually expanded to all the parent classes of device managers and switches. For the egress timestamps, we added two metadata fields accessible from P4 to specify the timestamp format (currently a single format is supported) and the offset in the packet where the egress port code must inject that timestamp. In addition, we ensured total backward compatibility of the converted and new sections of the software switch by creating a set of new functions capable of handling extended metadata alongside the existing code.



Validation and Ongoing Work: After passing all existing, unmodified, tests, to show good backward compatibility, we developed new test codes to verify our new functionalities. As none of the unit tests in current BMv2 or P4C use the BMI port for testing it, but instead inject packets afterwards, we created a libpcap mock to do actual port code tests. Rigorous testing with our new codes uncovered and addressed some bugs and race conditions in the current software switch (not caused by our new code). Using the new ingress timestamp with enhanced precision, we reduced the inconsistency of the ingress timestamp from a range of 100-1000 milliseconds to 0-10 microseconds. Improvements were also observed in the final offset value calculated using PTP-1588. Currently, we are obtaining results from our egress timestamp implementation, which will be finished and ready for the P4 Workshop.