Monitor Traffic Flow through In-band Network Telemetry (INT)

Chih-Yuan Sun (cs2368)
Yu-Hsuan Chen (yc2423)

Objective

In this project, we want to let the packet receiver can get the information about how the packet is forwarded in the network. There is several information about the packet that the packet receiver may want to know.

1. **Forwarding path -** the id of the switch that is involved in the packet transmission.
2. **Length of queue -** when the packet is enqueued, the number of packet in the queue
3. **Time stayed in queue -** the time, in microseconds, that the packet spends in the queue
4. **Ingress Timestamp -** the time, in microseconds, that the packet first appears in the ingress pipeline.
5. **Enqueued Timestamp –** the time, in microsecond, that the packet is first enqueued.

User can select the information from those choices. The switch in our network will insert the information into the packet and transmit it to the receiver.

Header Design

Byte 0	Byte 1	Byte 2	Byte 3
Ethernet			
IPv4			
TCP			
INT Identifier			
INT Type	Reserved	INT Length	Reserved
ins_cnt	max_hop_cnt	total_hop_cnt	Instruction map
INT INFO			
Payload			

The header structure is shown in the figure, which can be separated to 6 parts:

1. **Ethernet, IPv4, TCP:** We don't change their structure. But during the packet transmission, the value of *IP.totalLen* should be update.
2. **<4 bytes> INT Identifier:** It is used to indicate whether the packet receiver wants INT information. If it is set to 1, the INT information will be inserted during the packet transmission.
3. **<4 bytes> shim_header:** It includes *INT Type, INT Length* and 2 reserved bytes.
   i. **<1 byte> INT Type:** The INT header type. The INT header type can be destination based,

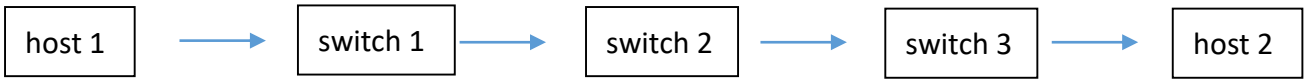
which means that the sender already has INT Information that needs to be transmitted to the receiver. Therefore, switches don't need to insert INT info into the packet. The INT header type can also be hop-to-hop based, which means that switches have to insert the INT data into the packet. In our project, we assume that the INT header type is hop-to-hop based.

ii. **INT Length:** The size of *shim_header*, *INT header* and *INT INFO.*

4. **< 4 bytes> INT_header:** It includes *ins_cnt, max_hop_cnt, total_hop_cnt* and *instruction map.*
   i. **<1 byte> Ins_cnt:** The instruction count, which is the number of kinds of information that the receiver may want to know.
   ii. **<1 byte> max_hop_cnt:** The maximum hop count, which is the maximum number of switch that is allowed to insert data into the packet.
   iii. **<1 byte> total_hop_cnt:** The number of switch that has already inserted INT data into the packet.
   iv. **<1 byte> instruction map:** Each bit in the field indicate whether the receiver wants a certain kind of information
      1. **<1 bit> e:** The most significant bit. If *max_hop_cnt* == *total_hop_cnt, e* will equal to 1, and then the INT data insertion is not allowed anymore.
      2. **<1 bit> ins_0:** If the receiver wants to know the forwarding path. Ins_0 will be set to 1, and the switch id will be recorded.
      3. **<1 bit> ins_1:** Time stayed in queue
      4. **<1 bit> ins_2:** Enqueued timestamp and the length of the queue
      5. **<1 bit> ins_3:** Ingress timestamp
      6. **<3 bits> reserved bits**
5. **INT INFO:** According to the instruction, switches will insert different INT information in this field.
6. **Payload**

Scenario

We assume that t sender h1 want to transmit a packet to receiver h2, and the network topology is



The sender h1 generate a packet, and the part of header is shown below:

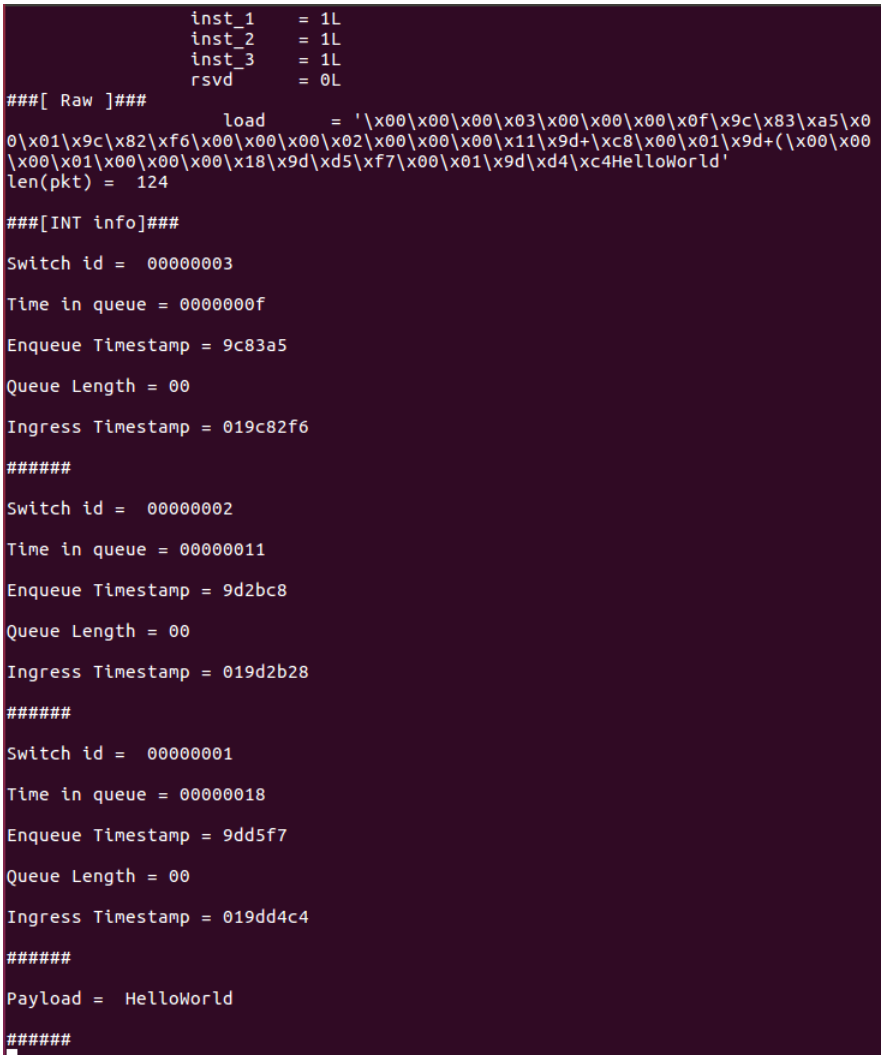
INT Identifier = 1			
1	00000000	2	00000000
2	5	0	01100000
Payload = "HelloWorld"			

When receiver h2 get the packet, the header will become

INT Identifier = 1			
1	00000000	8	00000000
2	5	3	01100000
(sw3) id = 3			
(sw3) t in q = 0x7			
(sw2) id = 2			
(sw2) t in q = 0xd			
(sw1) id = 1			
(sw1) t in q = 0x11			
Payload = “HelloWorld”			

Testing Result

The parameter setting and the network topology is the same as what we used in #scenario# part, but the instruction map is 01111000. The following figure is the data that the receiver gets.



Future Work

Our original target is, when there are multiple routing paths, according to the INT information, the switch can decide the routing path which has the lowest latency. However, we don't have enough time to implement it. We will try to implement it in the future.

How to Run the Program?

1. Run *sudo ./p4app run mini.p4app*
2. Run *sudo ./p4app exec m h1 /tmp/send.py 10.0.2.101 "HelloWorld" 1 1 1 1*
   The input vaules '1 1 1 1' are the setting of instruction map. User can change any of the bit to 0 to disable certain information.
   User also can just run *sudo ./p4app exec m h1 /tmp/send.py 10.0.2.101 "HelloWorld"* if they don't need any INT information
3. The INT information will be displayed on the screen.