

Introduction

With ever-growing volume of data being transferred over the Internet, the need for reliable monitoring becomes more urgent. Monitoring devices should be able to provide accurate information such as traffic patterns, statistics and various anomalies. This technical report describes an implementation of a network flow monitoring system using a dedicated hardware platform cooperating with the host PC. By exploiting the hardware-software codesign principle, we implemented time-critical functions in hardware and the rest in software. This way, the NetFlow probe offers good performance at low cost.

NetFlow as a protocol for flow monitoring, first implemented in Cisco routers, is the most widely used measurement solution in a form of NetFlow v5. Statistics on IP traffic flows provide information about who communicates with whom, how long, how often, using what protocol and service and also how much data was transferred.

Following part of this report describes the implementation of NetFlow v5 probe firmware itself.

System Architecture

Hardware of NetFlow probe consists of host computer and NetFPGA network card. The measurement process is completely implemented on the NetFPGA whereas control, configuration and collecting process are implemented as a user software running in the host computer.

The main parameters of the probe are as follows:

- Measurement of four 1 Gbps interfaces at the line rate
- Memory for up to 60000 concurrent flows
- Indexing of flow records using hash and 8 additional parallel lookups
- Export of flowrecords using NetFlow v5 export protocol

Application Firmware

The firmware part of NetFlow probe is composed of several units chained in a pipeline (see Fig. 1). The pipeline is instantiated as a user data path module into NetFPGA firmware. Each unit in the pipeline has its specific task which usually consists of processing of arriving data and adding the result in the output data. The interconnection of units is provided by netfpga interconnection protocol, i.e., data bus, control bus, write and ready signals.

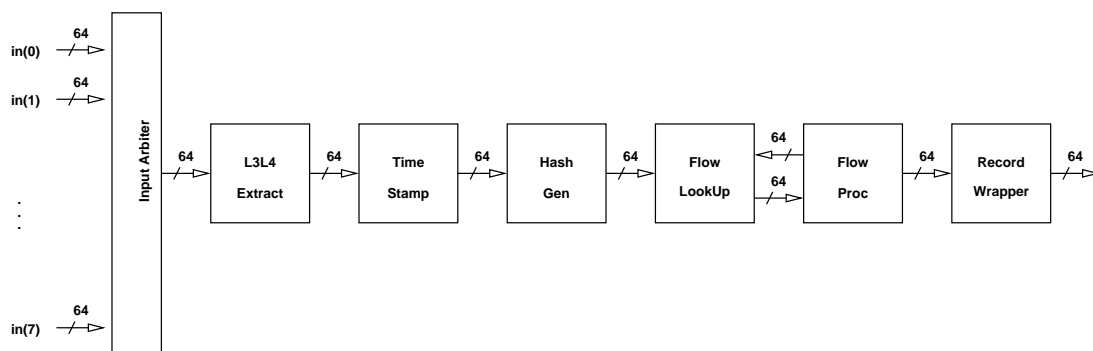


Figure 1: Processing pipeline of NetFlow firmware

NetFlow probe on NetFPGA

NetFlow probe on NetFPGA

Verze #1.00, 2008-12-12

L3L4 Header Parser

Parser extracts significant fields of the packet header for monitoring which forms a so called Packet Record (later referred to as *PR*). Rest of the packet payload is discarded. The input and output of unit is displayed on Fig. 2. The unit identifies the type of packet encapsulated in the frame and extract information to create the Packet Record. Supported packets are:

- TCP/IPv4
- UDP/IPv4
- ICMP

Other packets are discarded as not relevant to NetFlow measurement. The unit consists of two independent processes, one is to parse the packet header and the second one is to create correct output stream.

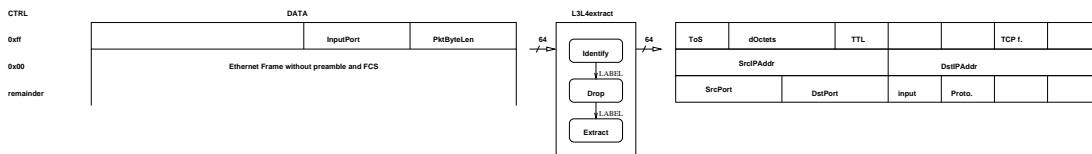


Figure 2: Extracting L3 and L4 information from the packet

Table 1: Address space of L3l4extract module.

Name	Access	Description
NFLOW_L3L4EXTRACT_TOTAL_PACKETS_REG	R	Total number of seen packets
NFLOW_L3L4EXTRACT_ACCEPTED_PACKETS_REG	R	Packets that were accepted for NetFlow measurement

Timestamp Unit

Timestamp unit inserts current timestamp value from the timestamp counter into the packet record. The timestamp represents *SysUpTime* – time from the start of the NetFlow monitoring. The timestamp is held in milliseconds. The timestamp counter is 32 bit and overflows after 49 days and 17 hours. The timestamp is inserted in the packet record, see Fig. 3. The speed of Timestamp counter is adjusted by the increment value. The increment value represents the number of clock cycles in one millisecond, i.e., the length of one millisecond. This value can be set by software in order to speed up or slow down the counter which allows to synchronize time domain of the firmware with the time domain of local host.

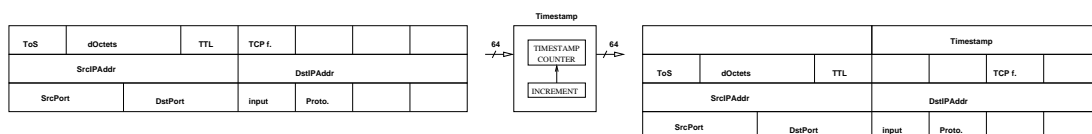


Figure 3: Timestamp unit

NetFlow probe on NetFPGA

NetFlow probe on NetFPGA

Verze #1.00, 2008-12-12

Table 2: Address space of Timestamp module.

Name	Access	Description
NFLOW_TIMESTAMP_INCREMENT_REG	RW	The length of millisecond in number of clock cycles
NFLOW_TIMESTAMP_TIMESTAMP_REG	R	Current value of the timestamp, i.e., SysUp-Time
NFLOW_TIMESTAMP_FRACTIMESTAMP_REG	R	Fraction of current timestamp in number of clock cycles

Hash Generator

Hash Generator computes a 64-bit hash value (CRC-64) and inserts it into the packet record, Fig 4. The hash value is computed only from following fields:

- SrcIPAddr
- DstIPAddr
- SrcPort
- DstPort
- Input – Input Interface
- Protocol

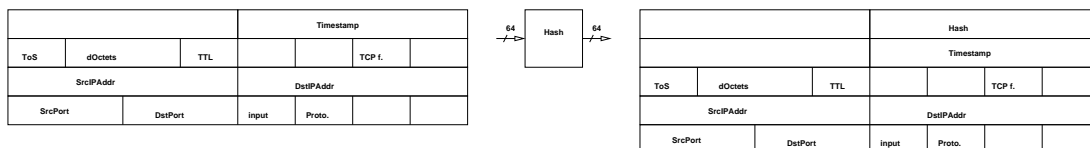


Figure 4: Hash Unit

The probability that two different flows share the same hash result (supposing the uniform distribution of hash) is:

$$p_{collision}(n) = 1 - \frac{m!}{m^n(m-n)!} \approx 1 - e^{-\frac{n^2}{2 \times m}} \quad (1)$$

where in our case $c = 2^{48}$ (only 48 bits of CRC-64) which is the number of possible CRC-64 values and $n = 4000$ is number of active flows. The probability of collision is very low, $p_{collision} = 10^{-13}$.

Table 3: Address space of HashGen module.

Name	Access	Description
NFLOW_HASHGEN_INITSEED0_REG	RW	Low 32 bits of hash init seed.
NFLOW_HASHGEN_INITSEED1_REG	RW	High 32 bits of hash init seed.

NetFlow probe on NetFPGA

NetFlow probe on NetFPGA

Verze #1.00, 2008-12-12

FlowLookUp Unit

FlowLookUp splits the hash value into two parts. First part is used to address a line which contains 8 hash values of 8 different flow records (8 fingerprints). These fingerprints are compared to the second part of the splitted hash. If there is a match with one of the hash values in the line then the flow record is already in the flow memory. Its address is acquired as the join of first part of the hash and the rank of the matched fingerprint. If there is no match and number of flow records in the set is lower than 8 then the free space is used to enter new fingerprint. If there is no match and no space then an arbitrary flowrecord in the set is expired and replaced with new one.

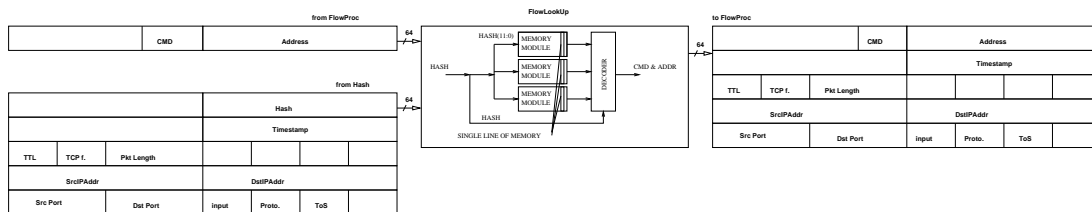


Figure 5: Flow LookUp Unit

When a flow record is about to be expired the FlowLookUp receives command from FlowProc unit to delete the corresponding fingerprint from the hash table. Immediately after this command is executed it is send back to a FlowProc unit.

Table 4: Address space of FlowLookUp module.

Name	Access	Description
NFLOW_FLOWLOOKUP_DEBUG_REG	RW	Debug register with arbitrary information.

Flow Processing Unit

Flow processing unit controls initialization of new flow records, updates of existing flow records, and expiration of inactive flow records. The expiration process runs in parallel with create/update process, it is displayed on Fig. 6. The mutual exclusion is not necessary as it is implicitly quaranteed by following protocol.

1. When the expiration process identifies inactive flow record then it sends a delete command to the FlowLookUp unit.
2. The FlowLookUp unit deletes the index of the flow record and sends the delete command back to the FlowProc unit.
3. The FlowProc unit retrieves and deteles corresponding flow record from its memory and send the expired flow record to its output.

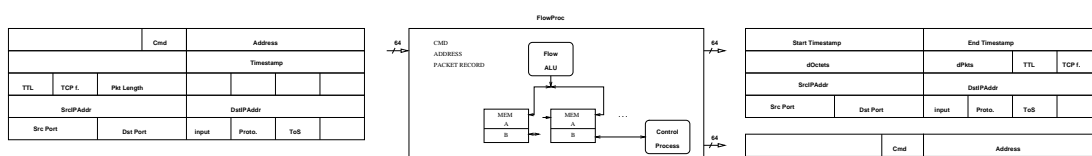


Figure 6: Flow Processing Unit

NetFlow probe on NetFPGA

NetFlow probe on NetFPGA

Verze #1.00, 2008-12-12

The computation of values in flow record depends on the previous values of the flow record and command issued by the FlowLookUp process. Following commands are specified:

- Create (Init) flow record
- Update flow record
- Delete flow record – no operation is performed upon values

The FlowALU implements following operations to support Init and Update commands:

```
fr.StartTimestamp = init ? pr.TimeStamp : fr.StartTimestamp ;
fr.EndTimestamp = pr.TimeStamp;
fr.dOctets = init ? pr.PktByteLen : ( fr.dOctets + pr.PktByteLen ) ;
fr.dPkts = init ? 1 : ( fr.dPkts + 1 ) ;
fr.ttl = pr.ttl;
fr.TCPflags = init ? pr.TCPflags : ( fr.TCPflags | pr.TCPflags ) ;

fr.srcipaddr = pr.srcipaddr;
fr.dstipaddr = pr.dstipaddr;
fr.srcport = pr.srcport;
fr.dstport = pr.dstport;
fr.input = pr.input;
fr.proto = pr.proto;
fr.tos = pr.tos;
```

Table 5: Address space of FlowProc module.

Name	Access	Description
NFLOW_FLOWPROC_CNT_ITEMS_REG	R	Number of flow records in memory.
NFLOW_FLOWPROC_CNT_NEW_REG	R	Number of received commands to create a flow record.
NFLOW_FLOWPROC_CNT_UPDATE_REG	R	Number of received commands to update a flow record.
NFLOW_FLOWPROC_CNT_DELETE_REG	R	Number of received commands to delete a flow record.
NFLOW_FLOWPROC_ACTIVE_TIMEOUT_REG	RW	Active timeout in milliseconds.
NFLOW_FLOWPROC_INACTIVE_TIMEOUT_REG	RW	Inactive timeout in milliseconds.

Record Wrapper

The released flow records are temporarily stored in a Record Wrapper module. As soon as the following conditions are met the stored flow records are reformatted to NetFlow v5 format and wrapped into NetFlow v5 protocols and written into output queue module of NetFPGA platform.

- 15 records arrived in the buffer
- The first record in the buffer is more than 20 ms old

The NetFlow v5 datagram is sent to the output interface specified by software register. There could be more output interfaces specified as the output interface is one-hot encoded.

NetFlow probe on NetFPGA

NetFlow probe on NetFPGA

Verze #1.00, 2008-12-12

Table 6: Address space of RecordWrapper module.

Name	Access	Description
NFLOW_RECORDWRAPPER_SRC_IP_REG	RW	Source IP address of NetFlow probe.
NFLOW_RECORDWRAPPER_DST_IP_REG	RW	Destination IP address of collector.
NFLOW_RECORDWRAPPER_SRCDST_PORT_REG	RW	Source and destination ports.
NFLOW_RECORDWRAPPER_EPOCH_SECONDS_REG	RW	Number of seconds between the epoch and the time when UDP packet was sent.
NFLOW_RECORDWRAPPER_OUTPUT_PORT_REG	RW	One-hot encoded output interface.

Application Software

The application software of NetFlow probe consists primarily of configuration and statistical interface and a simple collector.

The concept is displayed on Fig. 7.

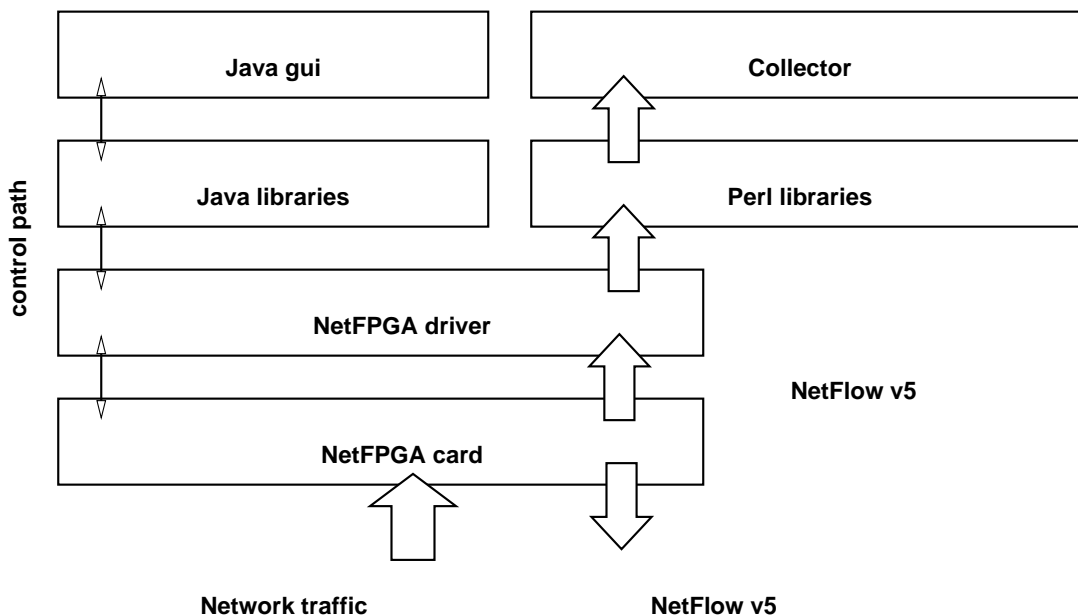


Figure 7: Architecture of Software

Conclusion

The goal of NetFPGA NetFlow probe is to show a potential of network FPGA cards to gather NetFlow data. It allows further improvement for example increasing the memory (using SSRAM or DRAM modules). Next the timestamp could be enhanced to 64 bits or the indexing scheme could be modified.

NetFlow probe on NetFPGA

NetFlow probe on NetFPGA

Verze #1.00, 2008-12-12

References