\chapter{Conclusion and Future Work}

In this chapter the conclusions derived from the implementation and evaluation of the In-Network Event Processing framework are presented. The future work that can be undertaken to build up on the implementation and the conceivable improvements are additionally discussed.

\section{Future Work}

In the current implementation, event processing actions do not take advantage of the megaflow cache implementation of the Open vSwitch. Instead each event has to be looked up the OpenFlow processing pipeline and event actions applied for accurate results. To achieve the same, the megaflow cache eviction rate is increased which results in poor performance of the Open vSwitch bridge. This is not ideal because this affects all the systems bridged using Open vSwitch and not just the implemented event processing pipeline. To avoid this problem a sophisticated re-validator thread may be developed to evict only event based rules from the cache and allow other rules to remain cached. \newline

The implemented event processing within Open vSwitch results in a significant increase in processing cycles per packet. This is because for each packet, the application layer is accessed, event attributes are extracted and de-serialized for further processing. This adds significant cycles per packet. A future work may address this drawback to make the event extraction process much leaner than what it is currently.\newline

Furthermore, the current implementation focuses on the UDP transport protocol. A future work may extend the support to other protocols.

\section{Conclusion}

Network virtualization and Software defined networking offer boundless possibilities for provisioning chained network functions on demand with the aid of software-based solutions and programmable network control planes. As part of research conducted in the thesis, an exercise in programming the network control plane with application context and enabling the data plane to process application logic is presented within the context of a complex event processing ecosystem. To achieve the goals of the research the following contributions have been made:

\begin{itemize}

\item An event processing framework is implemented within the highly adopted Open vSwitch.

\item The vSwitch is enabled to perform logical and stateful operations based on user logic configured as event rules.

\item A framework to remotely offload event rules onto the network control plane via http is implemented using the RYU controller.

\item A thorough evaluation of the implementation against several parameters is detailed and discussed.

\end{itemize}

The results of the evaluation shows that the benefits of detecting and redirecting events at the network and playing the role of a in-network event broker are compelling. Evaluation of this model shows a reduction in point-to-point latency between the producers and consumers of events and significant reduction in network traffic and processing for single staged processing systems by avoiding the utilization and context switch to a broker application. These results when extrapolated to multi-staged processing systems can potentially avoid multiple context switches and thereby improve the latency significantly and reduce burden on the network. However the results also show in an increased number of processing cycles per packet; which when viewed along with added benefits can be considered as a modest price to pay. When higher level logical and stateful operations are performed on event attributes, the benefits are less apparent in the current implementation as a result of disabled caching. Although a reduction in network traffic, prevention of context switch to a broker and consequent avoidance of broker processing are observed, the point-to-point latency increases because of reliance on the OpenFlow processing pipeline instead of the cache. In addition to the impact on the event processing pipeline, this also adversely impacts the generic performance of the Open vSwitch. A possible solution to this problem is presented in section 6.1. Overall the thesis elaborates on the potential of offloading aspects of event processing onto the underlying network. Although stateful event operations are implemented, the benefits are not apparent because of the current caching limitations. Nonetheless, whilst performing the role of event broker the benefits become more apparent. This provides network operators with promising avenues to explore models of complementing existing complex event processing ecosystems with highly tuned application-aware custom network solutions.