Evolving Switch Architecture toward Accommodating In-Network Intelligence

Shuangwu Chen, Xiang Chen, Zhen Yao, Jian Yang, Yangyang Li, and Feng Wu

Abstract

Motivated by the breakthroughs of AI in both theory and applications, we are perceiving a great potential for network innovations from a new dimension of intelligence. However, as the engine of networks, switches are designed as "dumb" network elements with the sole purpose of forwarding network packets, and thus a barrier against the entrance of AI as an intrinsic part of the network is unconsciously erected. This article proposes an evolved switch architecture aimed at breaking this barrier and accommodating in-network intelligence. We enhance the current switch architecture by embedding an intelligence plane, which is externalized as an intelligent computation pluggable module of an evolved switch. The module employs an integrated solution of "X86 CPU+GPU+DPDK," which provides a high-performance and high-throughput open platform for hosting in-network intelligence. We further conceive a flexible processing framework for an intelligent traffic measurement, recognition, and traffic regime. We also carry out extensive experiments to demonstrate the capability of the evolved switch by deploying a prototype in a campus network, with two promising application scenarios: in-network application identification and in-network anomaly detection.

Introduction

As the most important information infrastructure, the Internet has been undergoing a great leap forward in development in the past few decades, and it becomes increasingly busy with billions of websites, active users, and connected devices. The huge amount of data generated by these devices are leading to the zettabyte era where the global IP traffic will grow to 4.8 ZB per year by 2022 [1]. This explosive trend in the Internet inflicts unprecedented challenges of scale, complexity, dynamics and cost on the current "human-in-the-loop" network operation and management. Due to the reliance on humans to intervene, it is prone to induce misoperations, slow response to network events, and lots of heavy manual work [2]. Against these leap-over changes in the network landscape, "human-on-the-loop" network operation, rather than human-in-the-loop, is highly expected to bolster the robustness of the cyber network and to achieve prompt response to network events and dynamics.

Recently, artificial intelligence (AI) has made a breakthrough in both theory and applications. Due to its intrinsic nature of handling complex problems, AI provides a new opportunity to explore the network innovation of human-in-the-loop, aimed at self-motivated and proactive network operation. There are rich achievements in intelligent network operations, for example intelligent resource allocation [3], traffic predication [4], route planning [5], quality of experience (QoE) provisioning [6], and fault diagnosis [7], as well as intelligent anomaly detection. Although these advances improve the network performance, most of them lay emphasis on the network system modeling and intelligent algorithms concerning training, inferring, or decisions, rarely discussing a fundamental problem of whether current network infrastructure is capable of bolstering the running of these algorithms.

In light of the basic end-to-end principle, the Internet was originally conceived as a dumb network, where switches, the intermediary nodes of the network, were designed as dumb pipes only to forward network packets. Naturally, this design unconsciously induces a barrier against in-network intelligence, and the intelligence is expelled into the end system at the network periphery. Such a network landscape spawns cloud and edge-based intelligence [9]. With the aid of cloud and edge computing facilities, AI algorithms can be deployed by collecting, analyzing, and inferring network data concerning users, applications, content, or network status from the intermediary switches. Any policy generated by the AI algorithms is mapped into network configuration actions for automating its operations. This promising centralized intelligence exhibits a clumsy and tardy nature in response to network events and dynamics due to such a massive amount of data distributed across the network being collected, analyzed, and stored. In contrast, in-network intelligence that dwells in the intermediary nodes of a network provides an agile way to enforce online cognition of network events from network traffic and local execution of the network policy, thus achieving a prompt reaction to network events and dynamics. However, current network infrastructure lacks sufficient capabilities of supporting the deployment of in-network AI. In this section, we introduce two kinds of representative architectures in future networks.

As the engine of networks, switches are designed as "dumb" network elements with the sole purpose of forwarding network packets, thus a barrier against the entrance of AI as an intrinsic part of the network is unconsciously erected. This article proposes an evolved switch architecture aiming for breaking this barrier and accommodating in-network intelligence.

Shuangwu Chen, Xiang Chen, Zhen Yao, JianYang and Feng Wu are with the University of Science and Technology of China; Yangyang Li is with China Academy of Electronics and Information Technology

Digital Object Identifier: 10.1109/MCOM.001.1800923

The evolved switch may be applicable in various network contexts to achieve ubiquitous in-network intelligence. For instance, it can be deployed as an edge computing infrastructure in wireless cellular networks. Meanwhile, it can also be installed to measure and analyze the east-to-west network traffic to infer the network events in the data center network.

This situation inspires us to empower AI as an intrinsic part of network infrastructure so that the network operation and management can be automated. To achieve this, we advocate evolving the switch architecture from a "dumb" network element to an intelligent network agent having the capability of network cognition. We embed an intelligence plane into the switch while inheriting the original data plane and control plane. This intelligence plane, which is embodied as a pluggable intelligence card, collaborates with the data plane and control plane to form a local closed loop of "sensing-cognizing-acting," so the proposed switch is able to understand and react automatically to potential network events and dynamics. Notably, this idea is very different from the typical in-network computations [9], which focus on performing operations on the received data in the intermediate nodes, including data compressing, transcoding, aggregating, and so on, to improve the communication efficiency. The contributions of our work are three-fold:

- We propose an evolved switch architecture for hosting in-network intelligence, while conforming to legacy devices, systems, and protocols. In the implementation, we conceive a high-performance open platform to achieve high-throughput traffic processing based on commercial off-the-shelf "X86 CPU+GPU+DPDK."
- We develop a flexible processing framework for accommodating diverse demands. Equipped with various algorithms, it can be applicable to intelligent traffic sensing, cognizing, and regulating.
- We conduct two promising applications, application identification and anomaly detection, to demonstrate the potential of the evolved switch architecture. The results show its high feasibility and applicability for supporting in-network intelligence.

The remainder of this article is organized as follows. We present the structure of the evolved switch system architecture and its key concepts in the following section. Then we review the supporting software framework in detail. Following that, we present the open hardware platform for supporting congestion in traffic. Finally, we conclude the article.

Switch System Architecture for In-Network Intelligence

In this section, we give the basic design principles and then present a two-tier system architecture for supporting in-network intelligence.

Basic Design Principle

While evolving the switch architecture, we adhere to the following five particular principles.

Inheritance of Existing Functions: The evolved architecture should not alter the fundamental functions of a switch like data forwarding and controlling logic, which is beneficial for keeping interoperability with legacy devices, systems, and protocols.

Independence from Data Forwarding: Embedding AI into switches should not sacrifice the performance of data forwarding. The nontrivial AI workload running on the switch may occupy a significant amount of computing and memory resources. This principle guarantees high performance in the data forwarding of the switches.

High-Performance for High-Throughput Traffic Cognition: The traffic perception of the dynamic and uncertain network environment is the foundation of building in-network intelligence for autonomous decisions. This principle ensures the pre-requisite condition for recognizing massive traffic concerning diverse applications and services.

Openness for Accommodating Diverse Demands: Network intelligence is expected to be developed diversely for handling network issues in different aspects and at different levels. Hence, this principle guarantees an open and flexible platform that facilitates the innovation of in-network intelligence and accelerates its deployment.

Cooperativeness among Switches: Network-system-level knowledge is a precondition to enforce network-system-level intelligence. This principle enables intelligent switches to share their knowledge acquired locally and to form global network knowledge to achieve high-level intelligence.

It is challenging to design a novel switch architecture to fulfill all the requirements for compatibility, capability, and scalability concurrently.

System Architecture

Following the design principles, we conceive a two-tier comprehensive system architecture as shown in Fig. 1, which provides both the local and global network intelligence. We establish an intelligence plane in the evolved switches for hosting in-network intelligence, while the original data and control planes of the switches are kept in order to conform to a variety of legacy systems. The intelligence plane performs three roles within the network: sensor, cognition apparatus, and regulator. Specifically, the sensor refers to extraction of the critical features from the network traffic passing through the switches. The cognition apparatus is able to understand latent events occurring in the network. Equipped with various AI-based cognition algorithms, it can be applicable to carry out fault diagnosis, anomaly detection, traffic identification, and so on. The regulator means orchestrating various control policies to manipulate traffic flows. The aforementioned three functions constitute a closed loop of a "sensing-cognizing-acting" process, which contributes to managing the local network in an autonomous manner, that is, achieving local in-network intelligence.

The evolved switch may be applicable in various network contexts to achieve ubiquitous in-network intelligence. For instance, it can be deployed as an edge computing infrastructure in wireless cellular networks. Meanwhile, it can also be installed to measure and analyze the east-to-west network traffic to infer the network events in the data center network. In addition, in the scenario of enterprise networks, it can enforce intelligent traffic identification for internal visibility, thus understanding the network status, monitoring and and protecting internal assets.

For the purpose of achieving high-level network intelligence, a management plane is proposed to coordinate these switches. Since a specific network event may have its own unique

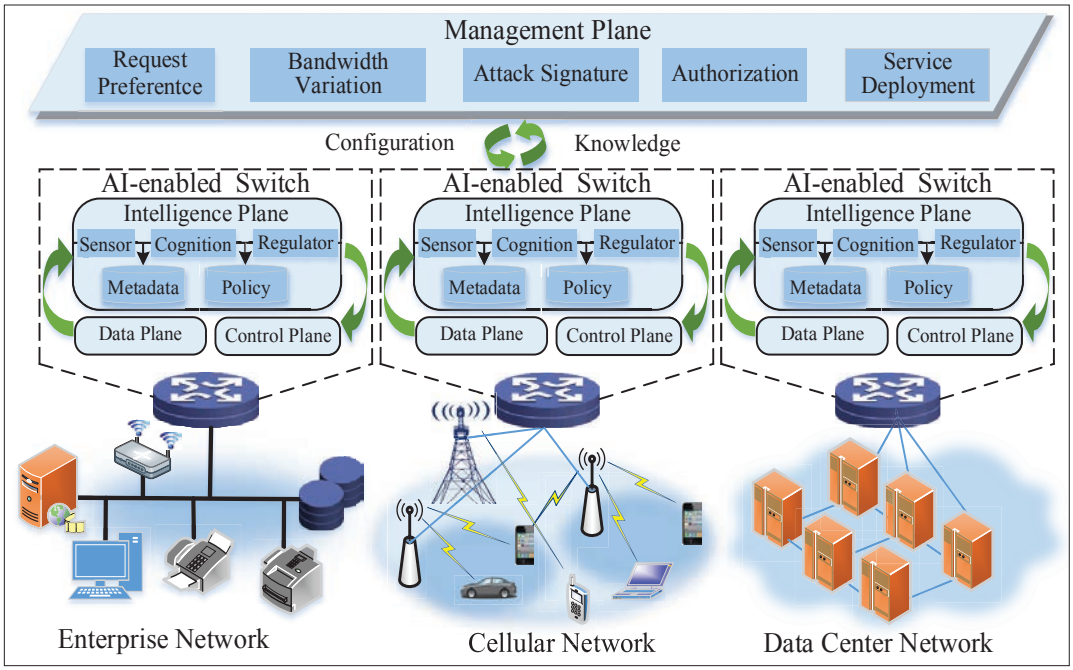

Figure 1. Two-tier in-network intelligence architecture with AI-enabled switches.

Different from the typical cloud-based network intelligence solution, only the cognition results rather than the raw data are gathered. Hence, the amount of data is trivial. It would considerably reduce the bandwidth consumption and avoid a data disclosure. This gathering-learning-configuring cycle at a high level could fulfill the self-management of the network system.

traffic pattern, such as request preference, attack signature, or bandwidth variation, sharing this knowledge is beneficial for improving network performance at the network level. For example, the locally learned features or knowledge of a detected attack can be shared and reused by other remote intelligent switches. The management plane gathers and learns this knowledge from different parts of the network, and optimizes the network configurations from a global perspective.

Different from the typical cloud-based network intelligence solution, only the cognition results, rather than the raw data, are gathered. Hence, the amount of data is trivial. It would considerably reduce the bandwidth consumption and avoid data disclosure. This gathering-learning-configuring cycle at a high level could fulfill the self-management of the network system.

Flexible Software Framework for Supporting Customizable Network Intelligence

The software framework for the intelligence plane depicted in Fig. 2 is composed of three basic components: traffic sensor, traffic cognition apparatus, and traffic regulator.

Traffic Sensor

As shown in Fig. 2, the traffic sensor consists of a *packet capture* module and a *feature extraction* module. In the packet capture module, a packet filter is employed to screen out the pertinent captured data by matching filter rules in terms of protocols, IP addresses, ports, and so on. Packet processing is supposed to be accelerated using multithreading, especially for a high-speed network. It may be that multiple threads are manipulating the same flow in parallel. However, for stateful protocols such as TCP, packets belonging to the same flow must be processed in sequence. Therefore, a *flow reassembly* module is designed to aggregate and align packets according to their protocols and sequence numbers.

As claimed in [10], the network traffic metadata could provide the underlying information required for representing and profiling user application activity. This would considerably reduce the requirement of data transmission, storage, and processing. Motivated by this fact, the feature extraction module in a traffic sensor is designed to extract the critical feature information concerning structural, statistical, or even hidden features. The structural features contain the fields of version, IP address, port number, and type of service in the protocol header of the network flow, while the statistical features are the measurement results of each flow including mean and variance of packet size, inter-packet duration, and so on. The hidden features represent a combination of latent attributes that describe the network communication, applications, and content. These features may be extracted for representation learning using a convolutional neural network (CNN) [11].

Traffic Cognition Apparatus

The *traffic cognition apparatus* gets deep insight into the metadata in order to understand the behavior of network entities and network applications as well as network services. Fundamental AI algorithms for traffic analysis can be dynamically deployed for traffic visualization, fault analysis, attack detection, and application identification. Specifically, the statistical metrics from the traffic sensor, that is, bandwidth consumption, traffic composition, latency, packet loss, and so on, can be used to carry out real-time visibility of the network dynamics. Training with historical data, a deep neural network (DNN) is able to learn the normal behavior of the network entities, and deviations from the normal behavior can be detected as abnormal or unknown events. Since the char-

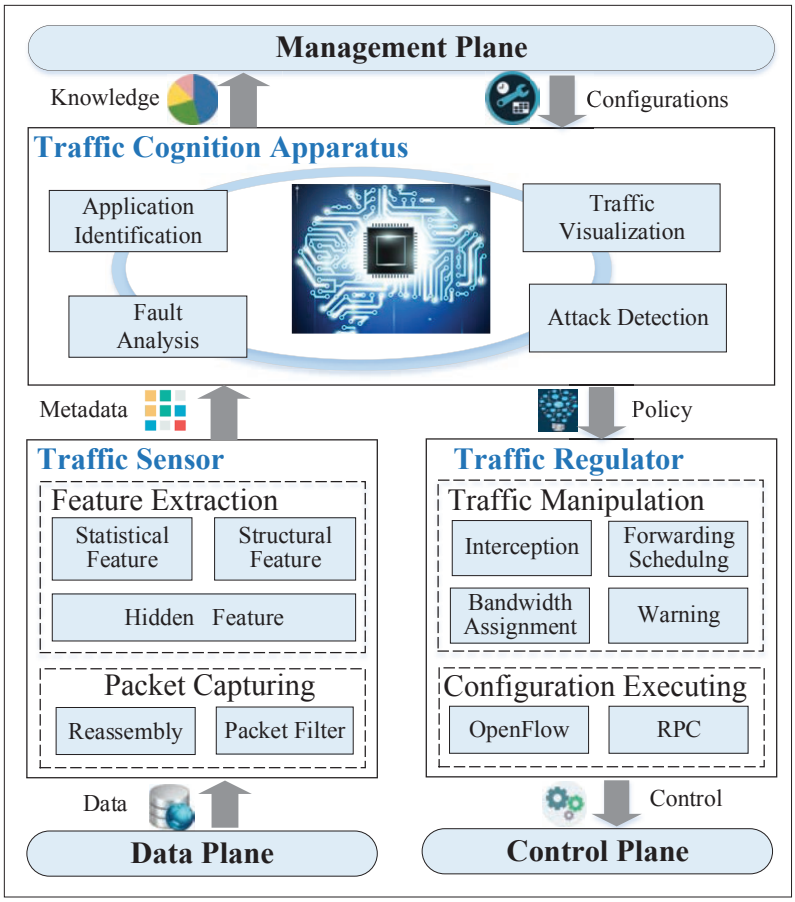

Figure 2. Software framework for supporting in-network AI.

acteristic metadata is able to portray the internal properties of network events, we implement a set of machine learning and deep learning methods to verify their feasibility for both in-network application identification and anomaly detection, which are further discussed in the experiments. The traffic cognition apparatus allows a centralized network management system to access the cognition results and customize their own cognition algorithms on the switches.

Traffic Regulator

Exploiting the cognition results, the traffic regulator is able to orchestrate and enforce control policies to manipulate the traffic. A set of basic operations includes flow interception, bandwidth assignment, forwarding scheduling, and network event warning. The flow interceptor may drop illegal packets, which provides a prompt way to defend against network attacks, malware, or other dangerous entities. With the aid of the local main control card, the forwarding scheduling module is able to dynamically adjust the forwarding port of the forwarding information base (FIB). It also allows support for assigning bandwidth and priority to certain applications to satisfy their own specific quality of service (QoS) requirements. An example is that real-time video streaming is more time sensitive than, say, file download. Thus, video transmission would be assigned more bandwidth or higher priority to prevent playback interruptions, while the file download could be delayed without significantly degrading its QoS performance. These configurations are fulfilled by the control plane of the switch via the Remote Process Call (RPC) protocol. The traffic regulator is also open to allowing the network operator to flexibly deploy their customized control policies.

Open Hardware Platform for Supporting Traffic Cognition

A single switch card may contain dozens of ports having a high capacity of 10/40/100 Gb/s. Hence, in order to conduct real-time traffic capture and analysis, it is essential to develop an open high-performance platform for the evolved switch. Although relying on dedicated chips (application-specific integrated circuit [ASIC], network processor, etc.), the traditional switch has gained success in implementing high-speed packet processing, this hardware design may be not appropriate for hosting in-network intelligence. Since a dedicated chip couples the hardware and software development, it lacks sufficient capability and flexibility for supporting the deployment of various computation-intensive AI algorithms. Despite the fast advances in general-purpose hardware components like X86 multicore CPU and the modern graphics processing unit (GPU), processing traffic online at such a high rate with general-purpose hardware is still nontrivial.

By inheriting the original modules of a switch including switch card, main control card, switch fabric, and backplane, we conceive an additional new intelligence card for online packet analysis and network decisions. The control information is exchanged via the backplane, which is isolated from the data traffic exchanged through the fabric. This isolation avoids the degradation of forwarding performance due to additional AI workloads. The collaboration of these modules is described as follows. The incoming network packets are forwarded to the out ports via switch fabric according to FIB, while, aided by the main control card, these packets of interested flows are duplicated and forwarded to the intelligence card for further analysis. The decision generated by AI on the intelligence card is mapped into the actions that are executed by the local main control card to optimize the network operations.

Data Highway for Network Traffic Analysis

In order to enforce online traffic analysis, a prominent step is to construct a data highway between the network interface cards (NICs) and CPU/GPU. The internal NICs are connected to CPU through PCIe, as depicted in Fig. 3. The state-of-the-art PCIe 3.0 has a bandwidth of 128 Gb/s, while the emerging PCIe 4.0 improves capabilities up to 512 Gb/s, which allows building a traffic acquisition platform with higher throughput.

With this hardware platform, we employ the user space packet IO engine, namely the data plane development kit (DPDK), to circumvent the slow in-kernel network stacks and construct a high-speed data channel from NIC to user space. By memory address mapping, it provides zero-copy data access for protocol parsing and packet batching in the CPU domain, which avoids frequent system calls and redundant memory copying between kernel and user space. Accordingly, only a single data copying from NIC to the device memory of GPU is performed.

We exploit the unique architecture of multi-core CPU and multi-queue NIC to conceive a one-core-to-one-queue mapping mechanism, as depicted in Fig. 3, for achieving highly parallel and high-throughput traffic acquisition. Specifically, the ring buffer of NIC is split into multiple queues, and each NIC queue is bound with a dedicated core of CPU. A hash function is invoked to uniformly distribute the incoming packets to the NIC queues, thus balancing the workload among the CPU cores. The packets capturing and processing in the CPU domain are performed concurrently, which significantly improves the throughput of packet processing. Since per-flow analysis requires the packets belonging to the same flow to be processed in sequence, the aforementioned hash key is used to map the packets belonging to the same flow to a unique GPU thread.

High-Throughput Traffic Analysis Using GPU

In the context of network intelligence, the GPU is not only the carrier of the running AI algorithm, but also the engine of high-throughput traffic feature extraction for further inferring network events relying on its considerable number of streaming processors. Transferring a deluge of raw traffic data directly from host memory to GPU incurs substantial PCIe transaction overhead, which further reduces the throughput between them. In order to handle this issue, we trim the raw packets to retain only valuable information that might be used by the subsequent traffic cognition in the GPU domain. On the other hand, data are transferred in batches, and a ping-pong buffering scheme is employed, as shown in Fig. 3, to improve the processing throughput. Specifically, in the device memory, we allocate two separate buffers to parallel the cross-device data transfer and the GPU data processing. While the GPU is performing deep analysis for the packets in one buffer, the CPU copies newly arrived packets to another. A monopolized CPU core is in charge of copying the data batches to GPU device memory through direct memory access (DMA). This design can take full advantage of the GPU performance due to no wait for data copying. In order to satisfy diverse processing functions like packet filtering, reorganizing, and cognizing, we organize the traffic processing in the GPU domain into a pipeline, as characterized in Fig. 3. This pipeline consists of three basic units: packet filter, feature extraction, and traffic cognition. For the sake of hiding the memory access latency, the technique of group prefetching is used to de-couple the access overlapping arising from these processing units, thus enabling the accelerated parallel processing for bolstering the in-network AI algorithm.

Achievable Performance Evaluation

In this platform, we conducted an experiment to evaluate the performance improvement by employing the general-purpose X86 CPU + GPU + DPDK design on traffic processing. A single intelligence card was used in the evaluation, which was equipped with two Intel E52620 v2 CPUs and one TESLA P4 GPU. We measured the achievable throughput and the GPU utilization against different packet sizes, while extracting the statistical features on a 40 GbE link. As shown in Fig. 4, the processing throughput increases with the packet size. The reason behind this is that the smaller the packet size is, the more packets are received and copied, which incurs additional cross-device I/O overhead. Specifically, for the 64 B/packet traffic analysis, a single intelligence card reaches over a throughput of 11 Gb/s with maximum GPU utilization higher than 50 percent. It is capable of tackling all the packets on a 40 GbE link with a packet size of 512 B. The results illustrate that the GPU can significantly boost traffic analysis. Given that the GPU is not fully utilized in Fig. 4, the CPU processing has become the bottleneck of our system. Thus, there is great potential to apply high-end CPU and GPU to process packets in a higher-speed network.

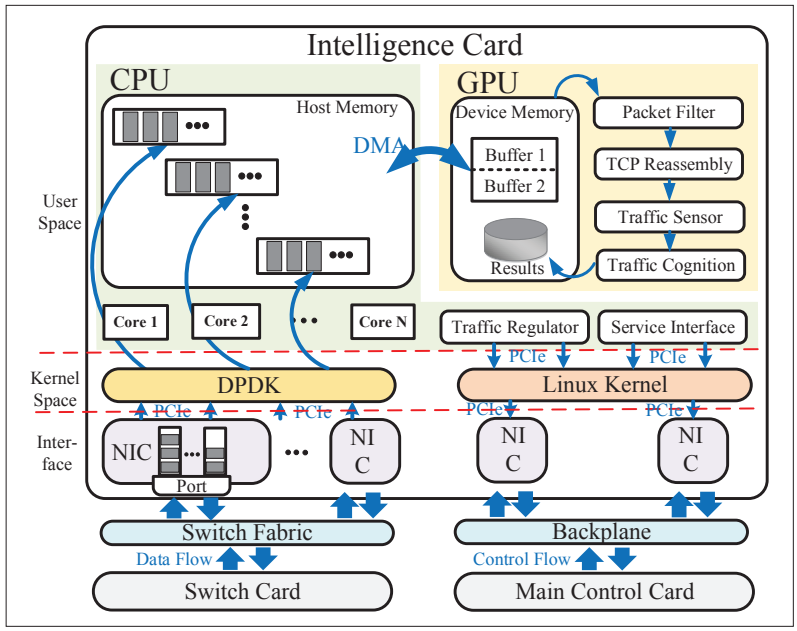

Figure 3. High-performance and high-throughput platform for evolved switches.

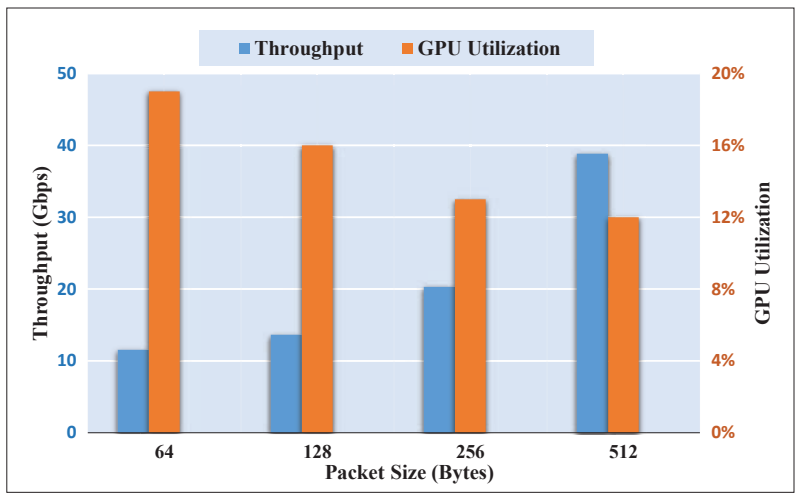

Figure 4. Processing performance for traffic feature extraction.

Application Scenarios for an AI-Enabled Switch

The performance of our AI-enabled switch is verified in two typical scenarios, namely application identification and anomaly detection. The experiment results validate its flexibility and applicability for supporting in-network intelligence.

Scenario	Algorithm	Parameter	Recall	Precision	F1-score
Application identification	XGBoost	tree_method = pgu_hist, max_depth = 6	0.9	0.91	0.9
	ThundersSVM	kernel = rbf, cost parameter c = 10, γ = 0.025	0.92	0.94	0.93
	DNN	6 fully-connected layer + 3 batch-normalize layer	0.94	0.95	0.93
	CNN	2 convolution layer + 2 fully-connected layer	0.97	0.97	0.97
Anomaly detection	XGBoost	tree_method = gpu_exact, max_depth = 15	0.97	0.96	0.97
	ThuderSVM	kernel = rbf, cost parameter c = 10, γ = 0.125	0.93	0.96	0.95
	DNN	8 fully-connected layer + 3 batch-normalize layer	0.95	0.97	0.96
	CNN	4 convolution layer + 2 fully-connected layer	0.98	0.98	0.98

Table 1. Parameters and performance of different algorithms.

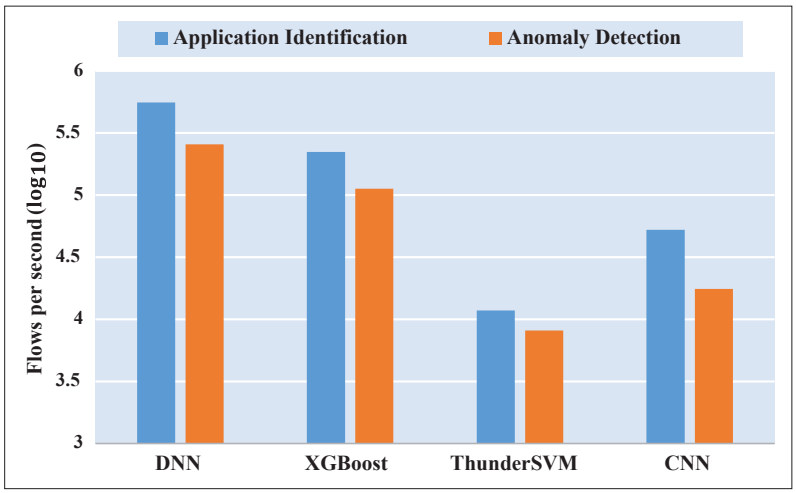

Figure 5. Performance of applying AI-enabled switches to both application identification and anomaly detection.

In-Network Application Identification

Application identification plays a critical role in network management, which contributes to traffic engineering, route planning, network provisioning, and traffic billing. Traditional traffic identification relies on rules matching, which has low treatment efficiency and fails to handle encrypted traffic. Instead, AI-based schemes use observable statistical features or latent embedded features of traffic flows to characterize the distinctive properties of network applications, which seem to be a promising solution. In the experiment, we invoked both traditional machine learning methods (i.e., ThunderSVM [12], XGBoost [13]) and the latest deep learning methods (i.e., DNN and CNN) to implement the in-network application identification on our AI-enabled switch. In particular, 45 statistical features are used as the inputs of the former three algorithms, which are classified into three types: protocol-related features (e.g., protocol type), size-related features (e.g., protocol size), and time-related features (e.g., duration and packet interval). The first 784 bytes of each flow are transformed to a virtual image and taken as the input of a CNN. The learning model is trained offline and inferred online. We captured and labeled the traffic of our laboratory as the training and testing datasets, which were composed of 20 common applications: WeChat, BitTorrent, Skype, online games, and so on. The captured traffic was replayed by TCPReplay for testing, and the identification accuracy is shown in Table 1. Due to the limited capacity of TCPReplay, it was impossible for us to replay the traffic at a rate up to 40 Gb/s. Instead, we took the metadata extracted from the raw data as the input of the four identification algorithms and measured their achievable processing speed, as shown in Fig. 5.

It can be seen that the four algorithms achieve an identification accuracy of higher than 90 percent and can tackle at least 10,000 active flows per second. Although ThunderSVM, XGBoost, and DNN have the same input features, DNN remarkably outperforms the other two schemes in both accuracy and speed. That is because the well-trained neural network is better at characterizing the difference among different applications, and the network structure of DNN is simple with low computational complexity. Also, we can observe that CNN has the highest accuracy of 97 percent, but its processing speed is relatively low. It illustrates that the payloads of the packets are beneficial for identifying different applications, and the convolution operation incurs additional computation overhead. The experimental results exhibit the high performance and high throughput of our hardware design.

In-Network Anomaly Detection

An essential aspect of network security is to detect the attacks traversing the network. AI-based anomaly detection is proven to be a promising way to prevent zero-day attacks and encrypted attacks, which would be a challenge for traditional signature-based methods. We conducted four in-network anomaly detection algorithms, that is, ThunderSVM, XGBoost, DNN, and CNN, on our AI-enabled switch. In the experiment, we used 80 statistical features to characterize the behavior and interaction of a session, which were taken as the input of ThunderSVM, XGBoost, and DNN algorithms. The inputs of CNN were the same as the above experiments. The CICIDS2017 dataset [14], including 10 different types of anomalies and massive benign traffic, were replayed to train and test the algorithms. Table 1 presents their achievable performance in terms of detection accuracy and recall rate. Similar to the above experiments, the metadata were used to verify the processing rate of these algorithms, as shown in Fig. 5.

From Table 1, we can observe that the detection accuracy of the four algorithms reaches higher than 96 percent, while CNN achieves the highest accuracy of 98 percent. It illustrates that the application layer data in the payload are beneficial for anomaly detection. As illustrated in Fig. 5, DNN is able to detect 100,000 active flows per second, which remarkably outperforms the other three algorithms. Meanwhile, compared to the application identification, the increase of input features will slow down the processing speed.

Conclusion

In order to embed AI into network infrastructure for automating the operation of networks, we evolve the switch architecture toward accommodating in-network intelligence. Different from existing switch architecture, we introduce an intelligence plane that could be externalized as intelligent computation pluggable modules in the switches while inheriting the original data plane and control plane. This built-in intelligence design allows us to upgrade current network infrastructure in a smooth and low-cost manner without additional upgrading of supporting facilities. The feasibility and applicability are verified by implementing this design in a commercial switch.

We believe that this proposed evolved switch architecture has many more implications for in-network intelligence than investigated in our work. Many interesting issues remain to be further explored. For instance, we will consider application-aware traffic engineering that can combine the flow information and network state to provide QoS guarantees. AI-based strategies such as re-routing and bandwidth adjustment can be implemented to optimize QoS. Second, multi-agent cooperative decision making can be introduced to enable collaborative network edge caching. Third, combining the intelligent switch with software defined networking will contribute to network-system-level intelligence.

Acknowledgment

This work is supported by the National Key R&D Program of China (no. 2018YFF01012200).

References

[1] Cisco Research, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," tech. rep., Feb. 2019; https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html, accessed Nov. 28, 2019.

[2] C. Fang *et al.*, "Data-Driven Intelligent Future Network: Architecture, Use Cases, and Challenges," *IEEE Commun. Mag.*, vol. 57, no. 7, July 2019, pp. 34–40.

[3] Y. He *et al.*, "Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," *IEEE Commun. Mag.*, vol. 55, no. 12, Dec. 2017, pp. 31–37.

[4] J. Feng *et al.*, "DeepTP: An End-to-End Neural Network for Mobile Cellular Traffic Prediction," *IEEE Network*, vol. 32, no. 6, Nov./Dec. 2018, pp. 108–15.

[5] N. Kato *et al.*, "The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, June 2017, pp. 146–53.

[6] B. Mao *et al.*, "A Novel Non-Supervised Deep-Learning-Based Network Traffic Control Method for Software Defined Wireless Networks," *IEEE Wireless Commun.*, vol. 25, no. 4, Aug. 2018, pp. 74–81.

[7] D. Palacios *et al.*, "Self-Healing Framework for Next-Generation Networks through Dimensionality Reduction," *IEEE Commun. Mag.*, vol. 56, no. 7, July 2018, pp. 170–76.

[8] R. Hofstede *et al.*, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 4, 2014, pp. 2037–64.

[9] H. Zheng *et al.*, "Energy and Latency Analysis for In-Network Computation with Compressive Sensing in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2811–15.

[10] G. Alotibi *et al.*, "Behavioral-Based Feature Abstraction from Network Traffic," *Proc. 10th Int'l. Conf. Cyber Warfare and Security*, 2015, pp. 1–9.

[11] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Commun. Mag.*, vol. 57, no. 5, May 2019, pp. 76–81.

[12] Z. Wen *et al.*, "ThunderSVM: A Fast SVM Library on GPUs and CPUs," *J. Machine Learning Research*, vol. 19, 2018, pp. 1–5.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proc. ACM SIGKDD Int'l. Conf. Knowledge Discovery Data Mining*, 2016, pp. 785–94.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proc. Int'l. Conf. Info. Systems Security and Privacy*, 2018, pp. 108–16.

Biographies

Shuangwu Chen (chensw@ustc.edu.cn) received his B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC). He is currently a postdoctoral researcher at USTC. His research interests include future networks, network security, and stochastic optimization.

Xiang Chen (cx0113@mail.ustc.edu.cn) received his B.S. degree from USTC, where he is currently pursuing a Ph.D.. His research interests include software defined networking, network intelligence, and network security.

Zhen Yao (yaozhen1@mail.ustc.edu.cn) received his B.S. degree from USTC, where he is currently pursuing a Ph.D. His research interests include software defined networking and multimedia communication.

Jian Yang (jianyang@ustc.edu.cn) received his B.S. and Ph.D. degrees from USTC. He is currently a professor at USTC. His research interests include future networks, multimedia communication, and stochastic optimization.

Yangyang Li (liyangyang@cetc.com.cn) received his Ph.D. degree from Beijing University of Posts and Telecommunications. He is now with the National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC). His research interests include mobile Internet, future networks, and edge computing.

Feng Wu [M'99, SM'06, F'13] (fengwu@ustc.edu.cn) received his M.S. and Ph.D. degrees from the Harbin Institute of Technology in 1996 and 1999, respectively. He is currently a professor and the Dean of the School of Information Science and Technology, USTC. His research interests include brain-inspired intelligence, image and video compression, and media analysis and synthesis.

We believe that this proposed evolved switch architecture has many more implications for in-network intelligence than investigated in our work. Many interesting issues remain to be further explored. For instance, we will consider application-aware traffic engineering that can combine the flow information and network state to provide QoS guarantees.