International Journal of Innovative Technologies
WWW.IJITECH.ORG
ISSN 2321-8665
Vol.01,Issue.02,
September-2013,
Pages:105-111

Improving Network Administration in the Areas of Path, Quality of Service and Security Management with Software Defined Networks

CHINTHAGUNTA MUKUNDHA[1], DR. MOHAMMAD VAMMANLI[2], DR. MOHAMMAD LALTWSOL[3]
[1]Research Scholar, Dept of CSE, Techno Global University, Shillong.
[2]Professor, Dept of CSE, Techno Global University, Shillong.
[3]Associate Professor, Dept of CSE, Techno Global University, Shillong.

Abstract: Many real-time services, like voice and video conversations, have been adapted for transportation over Ethernet and Internet Protocol networks. The growth of these services resulted in over complicated and inflexible configurations at the control plane of routing and switching hardware, and high demands are set on the availability of the network infrastructure. The Software Defined Network paradigm is designed to abstract the available network resources and control these by an intelligent and centralized authority with the aim to provide centralized path management, Quality of Service and Security to the network. On these networks, failures must be detected and restored within millisecond order to not disturb provided network services. Currently, the Open Flow protocol enables the SDN paradigm for Ethernet IP networks. We will attempt to solve the problems like failover times on Ethernet IP networks with the application of active link monitoring and advanced capabilities of the Open Flow protocol. To enable protection scheme, a routing algorithm is required that provides link-based protection. We designed a protection algorithm that guarantees protection, minimizes path cost on discovers protection paths and primary path for intermediate switches on the primary path with the main purpose to optimize network traffic, minimize failover times, and reduce the need for crankback routing.

Keywords: Real-time Services, Path Management, Protection Algorithm, Open Flow, Quality of Service.

I. INTRODUCTION

Since the adaptation of Ethernet and the Internet Protocol (IP) as network standards, many real-time services, like voice and video conversations, are adapted to be transported over IP networks. The low cost character and the ease of packet routing made operators to converge to IP networks. In order to transport and manage the enormous amounts of data in an efficient, robust and flexible manner, multiple networking protocols have been implemented in switching and routing network solutions. Generally, the switching and routing solutions can be split into a data and control plane. The data plane performs per-packet forwarding based on look-up tables located in the memory or buffer of the switch, where the control plane is used to define rules based on network policies to create the look-up tables. Due to the high demands on network performance and growing configuration complexity, the control plane has become over complicated, inflexible and difficult to manage. To solve this problem a new networking paradigm was needed, which was compatible with the widely used Ethernet switching and IP routing techniques. The solution was found in virtualization techniques used in server applications, where an abstraction layer is positioned above the server hardware to allow multiple virtual machines to share the available resources of the server. Software Defined Networking (SDN) adopted this paradigm and introduced an abstraction layer in networking. By abstracting the Network resources, the data and control plane are separated.

The data plane is located at the switch hardware, where the optimized forwarding hardware is preserved and the control of the network is centralized into an intelligent authority with the aim to improve flexibility and manageability. A centralized authority provides the intelligence to instruct network switches to route and control the traffic through the network infrastructure. Optimal paths through the network can be provided by the central authority in advance or on demand. The current implementation of the SDN networking paradigm is found in the Open Flow protocol [1] developed by Stanford University in 2008 and is currently under development with the Open Networking Foundation [2]. Open Flow has attracted some big vendors in the networking community and became the most popular realization of the SDN networking paradigm. In this paper the main objective is focused on switching Ethernet networks transporting Internet Protocol [3] data packets. The main purpose for Ethernet switches is to interconnect nodes to a local network, with the ability the exchange data packets. From the Open Systems Interconnection [4] reference model, Ethernet switches are located at the bottom two layers, the physical and data link layer. Ethernet is developed as a broadcasting mechanism to sense the broadcast channel and to transmit when the channel is not occupied.

On the physical level connectivity can, among others, be provided by 1000BASE-T twisted pair copper cables for local data distribution at 1 Gbps with a maximal range of 100 meters. To increase transmission range, optical transmission

Copyright @ 2013 IJIT. All rights reserved.

lines can be applied in the form of 1000BASE-ZX single-mode fibers. Connectivity is not limited to physical transmission lines. In Wi-Fi networks, Ethernet is also adopted as standard, so we can conclude that Ethernet can be utilized on a large variety of transmission mediums, which makes it the most popular protocol in current data networks.

Now a day's networks form the backbone for social communication and entertainment services. Providing these services over an inflexible current state switching network is leading to non-optimal network utilization and over-dimensioned networks, but also the requested service is not guaranteed. A better solution is found in the SDN philosophy, where the network topology is configured based on requests from network services and QoS solutions [5] can be implemented. Services request connectivity to a network and if the request can be fulfilled, paths through the topology are provided to the service for the requested amount of time. To embrace this philosophy, switch configurations must be performed centralized and an abstraction interface is needed to translate the current network state to the network services [2]. In Fig.1 the SDN concept is presented.

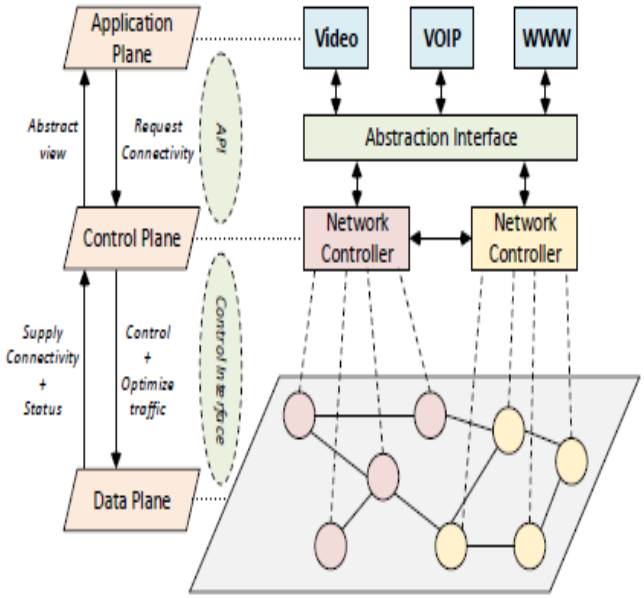

Fig.1. SDN concept of abstracting network view.

The SDN concept speaks of three planes like Data Plane, Control Plane and Application Plane.

- **Data Plane:** The Data Plane is built up from Network Elements and provides connectivity. Network Elements consist of Ethernet switches, routers and firewalls, with the difference that the control logic does not make forwarding decisions autonomously on a local level. Configuration of the Network Elements is provided via the control interface with the Control Plane. To optimize network configuration, status updates from the elements are sent to a Network Controller.
- **Control Plane:** Network Controllers configure the Network Elements with forwarding rules based on the requested performance from the applications and the network security policy. The controllers contain the forwarding logic, normally located at switches, but can be enhanced with additional routing logic. Combined with actual status information from the Data Plane, the Control Plane can compute optimized forwarding configurations. To the application layer, an abstract view from the network is generated and shared via a general Application Programming Interface (API). This abstract view does not contain details on individual links between elements, but enough information for the applications to request and maintain connectivity.
- **Application Plane:** Applications request connectivity between two end-nodes, based on delay, throughput and availability descriptors received in the abstract view from the Control Plane. The advantage over current state networks is the dynamic allocation of requests, as non-existing connectivity does not need processing at local switch level. Also applications can adapt service qualities based on received statistics. For example reduce the bandwidth for video streaming applications on high network utilization.

II. THEORETICAL BASIS AND LITERATURE REVIEW

Since the introduction and the acceptance of Open Flow as enabler of the SDN network paradigm, much research has been performed and published. Most of the research is performed to address and solve specific problem areas in software defined networking and, in some cases, networking in general. The main problem and research areas identified are scalability of control traffic in high performance data networks, the resiliency against network infrastructure failures and enabling security solutions on network level. The identified layered structure of performed research on SDN and Open Flow solutions is given below. For our graphical framework we define that controllers perform the computations and tasks to control traffic and additional layers can be added for administrative and synchronizing purposes.

- **Level-0: Switch Layer** - The lowest layer identified in the Open Flow structure is the switch layer, with the main purpose to deliver data plane functionality. Data plane functions are performed at the Switch / Open vSwitch sublayer, where the two additional sub-layers, being the Additional Switch Layer and Local Open Flow Controller, add additional functionality to perform minor control plane tasks. Additional layers are added to enhance control capabilities at the switch layer in the network;
- **Level-0.5: Virtualization Layer** - On top of the switch layer, the Virtualization Layer can be placed with the main function to divide and share the switch resources over multiple Open Flow controllers. It enables multiple virtual network topologies on top of a single physical infrastructure. Resources of physical switches are virtualized by this layer and presented to the Control Layer as multiple virtual switches;
- **Level-1: Control Layer** - The functionality of the control layer is to perform the tasks of the SDN control plane in a defined area of the network topology for a

number of switches. Decisions made at this layer influence only a part of the network and are locally optimal. In regular Open Flow configurations, only a single Area Open Flow Controller is present. Solutions have been proposed to enhance the control layer with additional area Open Flow layers to extend functionality, such as synchronization of Flow Rules with other controllers;

- **Level-2: Global Layer -** The top layer of the Open Flow layered structure in the framework has the functionality to control network topology at global level, where forwarding and routing decisions influence the whole topology. A Global Open Flow Controller can thus compute globally optimal routes through the network, as it controls all switches. The structure of the global layer is similar to the control layer, so an Open Flow controller and an additional layer.

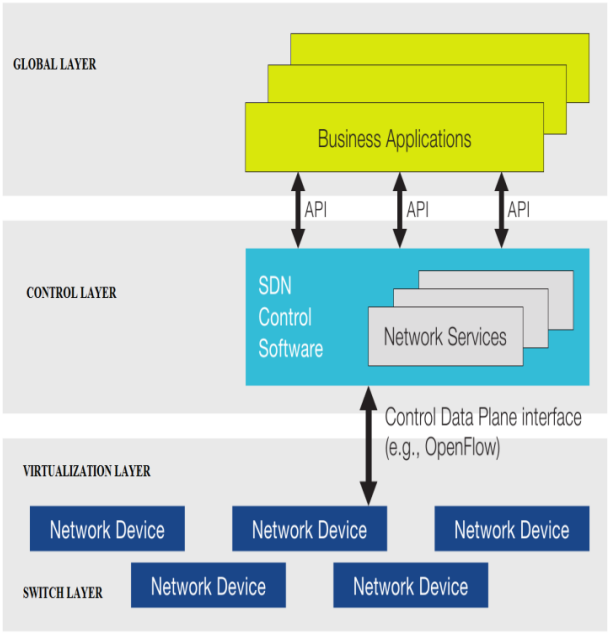

Fig.2. SDN Architecture.

The above fig2 gives overall structure of SDN networks. To create a robust and resilient network, the network topology must include redundant paths [6]. For a resilient control layer, the network state must be synchronized and identical between master and slave controllers, to support seamless overtaking without the need for network initialization and discovery processes. Additional modules and synchronization schemes must meet these requirements without compromising the performance and adding unwanted latencies. In [7] Fonseca et al. showed how the master-slave capabilities of the Open Flow protocol can be utilized. This indicates that a Primary Controller (master) has control over all switches and on the master failure, a Backup Controller (slave) can take over control of the switches. The developers of the replication component [7] came up with a solution, indicated in this review as CPR, which integrates a replication component into a general Open Flow controller. Research performed in [8] by Sharma et al. has the aim to deploy the SDN philosophy to carrier grade networks. These networks have high demands, looking to reliability and availability, which can be translated into network requirements, such as fast switchover times and minimal packet loss on network failures. Carrier grade networks must recover from a Network failure within 50 ms, without impacting provided services. These are high demands, looking to the process-flow of the Open Flow protocol and its current limitations.

To guarantee performance of the network, two resilience mechanisms can be applied to network, being protection and recovery mechanisms. Protection mechanisms configure pre-programmed and reserved redundant paths at the forwarding plane in case of failures. On failure detection, no additional signaling is required with the control logic and the reserved path is directly available. This makes protection a proactive strategy to increase resiliency of networks. The recovery mechanism can either be a proactive or reactive strategy, where the proactive recovery strategy is similar to the protection mechanism, but requires no reserved resources. The reactive recovery strategy constructs paths on demand and dynamically based on current topology status. Network security is applied to control networks access, provide separation between users and protect the network against malicious and unwanted intruders. It remains a hot topic under SDN researchers, because a basic security level is expected from a new network technology, as well as the fact that network security applications can easily be applied to the network control logic. Much research has been performed and the results of [10, 11, 12, 13] are used to determine security properties within SDN. All researchers follow roughly the same procedure to apply security to the network. The procedure consists on three steps where a short description of the process is given, as well as a reference to the performed research.

- **Classification:** Data flows through the network must be classified in order to determine malicious behavior and network attacks. Without classification it is impossible to protect the network and take according actions. The main source for traffic classification is found in traffic statistics [11];
- **Action:** Once a traffic flow is marked as malicious, the control layer must modify Flow Tables to protect the network and prevent propagation of the malicious traffic through the network. For each threat, different actions are needed, so the control layer must be flexible for quick adaption of new protection schemes [10];
- **Check:** The last process in the security process is the checking of computed flow rules with the applied security policy from the network manager. Flow rules may (unintentionally) disrupt the security policy and therefore an extra control process is needed. Preventing network security violations by checking Flow Rules before installation on the switches, completes the overall security process [12, 13].

To protect a network, a backup path must be computed by the control logic of the switches and provided to the data plane. When a failure occurs, the main purpose of the path protection schemes is to recover paths as soon as possible A protection scheme does not guarantee a backup path, assuming it exists in the network after failure, as multiple link or switch failures can invoke the protection plan. Re-active pushing of backup paths utilizes the current state of the network and is therefore more flexible. From Sharma et al. [8] and [9] we can conclude that both a Protection and restoration scheme can work in Open Flow environments, but recovery of paths with reactive pushing is significantly slower.

III. PATH MANAGEMENT

We found that more attention is needed for increasing the robustness of SDN. Robustness issues are identified at the transport layer, where protection and recovery schemes are present to protect data paths through the network. In [9] path-based protection is applied, where [14] applied segment-based protection. Before going into more detail on specific protection algorithms, the functional description will be translated to an algorithmic problem. Consider a bidirectional and connected network N(N,L) with N switches and L links. Each link l(Si → Sj) between a switch Si and Sj is characterized by a link weight w(Si → Sj), where the weight is non-negative (wi > 0) and we assume the weight is equal in both directions w(Si → Sj) = w(Sj→ Si). For SDN and OpenFlow networks, this weight is a singular and measurable link parameter, such as round trip time, packet loss or link utilization. The link weight is assumed additive, which means that some parameters must be transformed before summations can be applied. Path P(SA → SB) or PS→SB contains k switches and k − 1 links to travel between switch SA and SB, where the path cost C(PSA!SB) equals the sum of weights as given in [60].

$$C(P_{SA \to SB}) = \sum_{j=1}^{k-1} w(Sj \to Sj+1) \quad (1)$$

$P_{SA \to SB}$ is the set of all paths between switch SA and SBin N, where $P^*_{SA \to SB}$ is the shortest path for which $C(P_{SA \to SB})$ is minimized. We define S^*i as the set of (protected) switches transversed in $P^*_{SA \to SB}$, where i≠ B and the protection path from S^*i to S_B as $PP(S^*_i)$.Furthermore, we define S^*f and $l^*(Si \to Sf)$ as respectively the switch and link in $P^*_{SA \to SB}$ in failure, where f = i + 1. Given a single link or switch failure, the link-based protection problem is to discover path $P^*_{SA \to SB}$ in network N and for the set of switches S^*i compute and guarantee a protection paths $PP(S^*i)$ towards the destination node SB, assuming S^*f or $l^*(Si \to Sf)$ in failure, such that crankback routing or overall path costs are minimized.

A. Protection Algorithm for Link-Based Protection Paths

We will propose a scheme to solve the link-based protection problem, where we choose K=2. Protection scheme is suggested to provide survivability for the protected switches S^*i against a single link or switch failure. Our proposal consists of two phases:

- Phase A - Disjoint path discovery - Computation of shortest path and disjoint path pair are discovered . If feasible paths are discovered, protection can be guaranteed and protection paths can be discovered in phase B;
- Phase B - Protection path discovery - For each protected switch in the shortest path a protection path towards the destination must be discovered. To discover the requested protection paths, an extension is made on the modified Dijkstra algorithm. The aim for the extended Dijkstra algorithm is to minimize path cost or the application of c rank back routing.

Algorithm1 PHASE A:Discover Shortest Path and Disjoint path Discovery

STEP 1 - DISCOVER SHORTEST PATH

1. EXECUTE Modified Dijkstra's algorithm $(\mathcal{N}, S_A, S_B) \Longrightarrow Q_1$
2. IF Q_1 exists:
3. Go to Step 2
4. ELSE:
5. Stop algorithm

STEP 2 - DISCOVER DISJOINT PATH PAIR

1. EXECUTE Bhandari's algorithm $(\mathcal{N}, S_A, S_B, Q_1, s) \Longrightarrow Q_2, P_1, P_2$
2. IF P_2 exists:
3. SET $PP(S_A) = P_2$
4. ELSE:
5. Stop algorithm
6. IF $L_{Q_1} > 1$:
7. Go to Phase B
8. ELSE:
9. RETURN $\{Q_1, PP(S_A)\}$

B. Modified Dijkstra Algorithm for Path Management

Three modifications are required in order to use Dijkstra's algorithm in the disjoint path algorithm. First, the algorithm be adapted such that negative links costs can be applied. Second, the algorithm must directly output the shortest path, without the need for path construction and at last the running time must be reduced. All modifications are assimilated in algorithm, where C is the cost array, P is the discovered path, CP is the cost for path P, P* is the shortest path between SA and SB, V is the set of visited nodes and Q is a priority queue. The first modification includes the application of a priority queue to store intermediate results. More on the benefits of priority will be discussed later on this section. To enable negative link weights, the set of visited switches V is introduced. After a switch is extracted from the priority queue, it is relaxed and added to V. This mechanism prevents multiple iterations of the algorithm with the same switch and negative link weights will not result in continuous updates of the cost array. The priority queue is filled with a set containing the path discovered so far P, its corresponding cost CP and the selected switch Sj . During the relaxation process,

the cost array C(Sn), path cost CP(Sn) and path P are updated, if the relaxation condition is met.

STEP 1 - INITIALIZATION

1. a. SET $\mathcal{C}(S_x) = \infty$, $S_x \in U$
2. b. SET $C_P(S_A) = 0$,$\mathcal{C}(S_A) = 0$
3. c. SET $P = S_A$
4. d. ADD S_A to V
5. e. INSERT $\{C_P(S_A), S_A, P\}$ in Q

STEP 2 - SWITCH SELECTION

1. WHILE Q is not empty:
2. EXTRACT-MIN $\{C_P(S_j), S_j, P\}$ such that $C_P(S_j)$ $min_{S_x \in Q}\mathcal{C}(S_x)$
3. IF S_j is S_B :
4. Stop algorithm and return $\{P^*, \mathcal{C}\}$
5. IF S_j not in V:
6. a. ADD S_j to V
7. b. Go to Step 3

STEP 3 - RELAXATION

1. FOR all neighbors S_n of S_j in $\mathcal{N}$:
2. IF $C_P(S_j) + w(l_{j \to n}) < \mathcal{C}(S_n)$:
3. a. SET $\mathcal{C}(S_n) = \mathcal{C}(S_j) + w(l_{j \to n})$
4. b. SET $C_P(S_n) = C_P(S_j) + w(l_{j \to n})$
5. c. SET $P = P + S_n$
6. d. INSERT $\{C_P(S_n), S_n, P\}$ in Q
7. e. Go to Step 2

IV. QoS MANAGEMT

Quality of Service, or QoS, covers several mechanisms that were designed to support flows that require some performance guarantees[5]. Steps to end-to-end QoS support over the Internet are as follows:

- Define the service class a packet should receive at each switch.
- Allocate to each class a certain amount of resources.
- Sort the incoming packets to their respective classes.
- Control the amount of traffic admitted for each class.
- Apply the four steps above to each and every switch, or at least all bottleneck routers.

QoS routing is a routing scheme that takes into consideration the available bandwidth and other relevant information about each link, and based on that information selects paths that satisfy the quality of service requirements of a traffic flow. The objectives are of QoS are:

- **Dynamic Determination of Feasible Paths:** That is, to find a feasible path for the flow in question that can accommodate or at least has a good chance of accommodating the QoS requirements of the flow.
- **Optimization of Resource Usage:** QoS-based routing can be used to help balancing the load of the network by efficient utilization of resources, and thus improving the total throughput of the network.
- **Graceful Performance Degradation:** In overload situations QoS routing should be able to provide better throughput in the network than best effort routing or any state-insensitive routing scheme, and more graceful performance degradation.
- **QoS Routing With Resource Reservation:** Resource reservation and QoS routing are independent mechanisms but complement each other well. QoS routing can find feasible paths for flows that need QoS guarantees but cannot ensure that the path will remain feasible for the duration of the flow. Resource reservation protocols can be used to allocate the required resources along the selected path.

A. RSVP

The protocol most often suggested in papers concerning QoS routing and resource reservation is RSVP. It is receiver oriented, which means that the receiver of the data flow is responsible for initiation of resource reservation. When the source node initiates a flow, it sends a PATH message to the destination node identifying the characteristics of the flow for which resources are requested. Intermediate nodes forward the PATH message according to routing protocol in question. After receiving the PATH message, the destination node sends back a RESV message to do the actual reservation. Intermediate nodes decide separately whether they can accommodate the request. If any of them rejects the reservation, an error message is sent to the receiver. If the reservation is successful, necessary bandwidth and buffer space is allocated. After the connection and reservation is established the source periodically sends PATH messages to establish or update the path state, and the receiver periodically sends RESV message to establish or update the reservation state. Without update messages the reservation times out. This is called Soft-State. When RSVP is used together with QoS routing, the PATH messages are routed using QoS routing. The RESV messages and the actual reservation on resources is not affected by the routing protocol. Due to the more dynamic nature of QoS path Selection criteria, better routes can emerge more easily than in shortest path routing. That is, available bandwidth or other metrics can change rapidly, so that the current selected path is no longer the one with the best capabilities to accommodate the flow. Such sudden changes rarely happen in shortest path routing, since network topology usually stays more or less the same.

B. Routing Problems

Single Metric Routing Problems: In the simplest case the QoS requirements the problem is either an optimization problem,or a constraint problem. The metrics are divided into path-constrained and link constrained metrics. Concave metrics are link-constrained, because the metric for a path depends on the bottleneck link's value. Additive and

multiplicative metrics are Path constrained, because the metric for a path depends on all the values along the path. The four single metric problems are as follows:

- Link-optimization routing
- Link-constrained routing
- Path-optimization routing
- Path-constrained routing

V. SECURITY MANAGEMENT

Network security is applied to control networks access, provide separation between users and protect the network against malicious and unwanted intruders. It remains a hot topic under SDN researchers, because a basic security level is expected from a new network technology, as well as the fact that network security applications can easily be applied to the network control logic. There are two levels of security are defined. The first level invokes logical connections between end hosts inside the network. Protocols like Secure Socket Layer (SSL) or packet encrypting techniques must ensure connection security. Within SDN, this level of security plays an important role, as the control connection between switches and the centralized controller must be ensured. The Open Flow protocol [1] provides a mechanism to secure this connection, but is not required as start condition. It is up to the controller to secure the connection with Open Flow switches and a number of controller implementations have not implemented link security mechanisms [16]. When no link security is applied, a malicious node can impersonate the controller and take over control of the switches. Network traffic can be re-routed for analysis and information extraction. Applying link security between the control and data layer is thus the first prerequisite which must be fulfilled to ensure integrity on the network.

The second level of security is centered to protect switches, servers and end hosts in the network. Numerous examples are present to indicate the threats to the network as a whole. Malicious software can intrude the network, infect hosts and gather information, but also flooding attacks can disable network servers or overload OpenFlow switches and controllers. Security mechanisms must be implemented on the network to detect malicious traffic and take necessary actions to block and reroute this traffic. In current networking state, network security is applied at higher networking layers. Routers and firewalls perform security tasks at layer 3, where end hosts and servers host security applications at layer 7. With SDN, there is a central authority which routes traffic through the network and enables the possibility to apply security policies to all layers in networking. Much research has been performed and the results of [10, 11, 12, 13] are used to determine security properties within SDN. The procedure consists on three steps where a short description of the process is given, as well as a reference to the performed research.

- **Classification:** Data flows through the network must be classified in order to determine malicious behavior and network attacks. Without classification it is impossible to protect the network and take according actions. The main source for traffic classification is found in traffic statistics [11].
- **Action:** Once a traffic flow is marked as malicious, the control layer must modify Flow Tables to protect the network and prevent propagation of the malicious traffic through the network. For each threat, different actions are needed, so the control layer must be flexible for quick adaption of new protection schemes [10].
- **Check:** The last process in the security process is the checking of computed flow rules with the applied security policy from the network manager. Flow rules may disrupt the security policy and therefore an extra control process is needed. Preventing network security violations by checking Flow Rules before installation on the switches, completes the overall security process[12, 13].

V. CONCLUSION AND FUTURE WORK

This paper can be divided into three problem areas, being i.) Path Management with software defined networks and algorithms ii.)Quality of Service with software defined networks iii.)Security management with software defined networks. To increase the robustness on networks in general, path recovery mechanisms exist. These mechanisms must detect a network failure and restore the path to retain connectivity. The extended Dijkstra algorithm we developed, discovers protection paths, minimizes computational overhead and prevents routing loops. Overall, our protection algorithm solves the link-based protection algorithm problem and is applicable to regular networks. Although we believe SDN provides a solid foundation for future network solutions, especially in case of increasing robustness, there is room for future work on the subjects like study on Failover behavior of Open Flow Fast Failover Group Table, Expand link-based protection algorithm, Reduce BFD transmission interval.

VI. REFERENCES

[1]Open Network Foundation. (2013, apr) Openflow switch specification openflow version 1.3.2 (wire protocol 0x04). [Online]. Available: https://www.opennetworking.org/I mages /stories/downloads/sdn-resources/onf-specifications /openflow/openflow-spec-v1.3.2.pdf.

[2]Sdn architecture overview(version1.0).[Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf.

[3]J. Postel, "Internet Protocol," RFC 791 (INTERNET STANDARD), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1349, 2474,6864[Online]. Available: http://www.ietf.org/rfc/rfc791.txt.

[4]ITU-TX,"Information technology–open systems inter-connection–basic reference model: The basic model," International Organization for Standardization / International Electrotechnical Commission, Tech. Rep. Recommendation 200 (1994)| ISO/IEC 7498-1:1994, 1994.

[5]F.A.Kuipers, "Quality of service routing in the internet. theory, complexity and algorithms," 2004.

[6]F. A. Kuipers, "An overview of algorithms for network survivability," ISRN Communications and Networking, vol. 2012, p. 24, 2012.
[7]P. Fonseca, R. Bennesby, E. Mota, and A. Passito, "A replication component for resilient openflow-based networking," in Network Operations and Management Symposium (NOMS), 2012 IEEE. IEEE, 2012, pp. 933–939.
[8]S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Enabling fast failure recovery in openflow networks," in Design of Reliable Communication Networks (DRCN), 2011 8th International Workshop on the. IEEE, 2011, pp. 164–171.
[9]S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester, "Openflow: meeting carrier-grade recovery requirements," Computer Communications, 2012.
[10]S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. Gu, and M. Tyson, "Fresco: Modular composable security services for software-defined networks," in Proceedings of Network and Distributed Security Symposium, 2013.
[11]R. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attack detection using nox/openflow," in Local Computer Networks (LCN), 2010 IEEE 35th Conference on.IEEE, 2010, pp. 408–415.
[12]S. Son, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Model checking invariant security properties in openflow.".
[13]P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for openflow networks," in Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012, pp. 121–126.
[14]Sgambelluri, A. Giorgetti, F. Cugini, F. Paolucci, and P. Castoldi, "Openflow-based segment protection in ethernet networks," Optical Communications and Networking, IEEE/OSA Journal of, vol. 5, no. 9, pp. 1066–1075, 2013.
[15]P. v. Mieghem, Data Communications Networking. Purdue University Press, 2006.
[16]A. Jirattigalachote, C. Cavdar, P. Monti, L. Wosinska, and A. Tzanakaki, "Dynamic provisioning strategies for energy efficient wdm networks with dedicated path protection," Optical Switching and Networking, vol. 8, no. 3, pp. 201–213, 2011.
[17]R. He and B. Lin, "Dynamic power-aware shared path protection algorithms in wdm mesh networks." Journal of Communications, vol. 8, no. 1, 2013.
[18]P. Van Mieghem, H. De Neve, and F. Kuipers, "Hop-by-hop quality of service routing," Computer Networks, vol. 37, no. 3, pp. 407–423, 2001.
[19]T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol," RFC 4253 (Proposed Standard), Internet Engineering Task Force, Jan. 2006, updated byRFC 6668. [Online].Available: http://www.ietf.org /rfc/rfc4253.txt.
[20]M. Huynh, S. Goose, P. Mohapatra, and R. Liao, "Rrr: Rapid ring recovery submillisecond decentralized recovery for ethernet ring," Computers, IEEE Transactions on, vol. 60, no. 11, pp. 1561–1570, 2011.
[21]M. German, A. Castro, X. Masip-Bruin, M. Yannuzzi, R. Martínez, R. Casellas, and R. Muñoz, "On the challenges of finding two link-disjoint lightpaths of minimum total weight across an optical network," Proceedings of NOC/OC&I, pp. 217–224, 2009.