Int. J. Autonomous and Adaptive Communications Systems, Vol. 13, No. 1, 2020 11

SDN replaced deployment and real-time QoS provisioning based on network models

Junhua Zhang

College of Science and Technology,
Ningbo University,
Ningbo, Zhejiang, 315212, China
Email: junhuazhangg@21cn.com

Abstract: From industrial communications to real-time internet applications, end-to-end delay guarantee is one of the most important aspects of quality of service (QoS). As a new networking paradigm, software-defined networking (SDN) exposes an opportunity to provide this guarantee. In exploiting SDN, We can replace the legacy network devices to SDN forwarding units (SDN-FUs) gradually. This paper focuses on the SDN replaced deployment problem, striving for providing real-time QoS provisioning in a network meanwhile. Utilising previous works originated from network calculus, we first propose the candidate nodes selection methods for upgrading legacy nodes to SDN-FUs and then present a routing algorithm located on SDN controller to direct the running of SDN-FUs. Via a simulation about a small network, we show that the delay of a flow passing through a network can reduce significantly with the usage of SDN.

Keywords: quality of service; QoS; software-defined networking; SDN; network models; network calculus; real-time.

Reference to this paper should be made as follows: Zhang, J. (2020) ‘SDN replaced deployment and real-time QoS provisioning based on network models’, *Int. J. Autonomous and Adaptive Communications Systems*, Vol. 13, No. 1, pp.11–22.

Biographical notes: Junhua Zhang received his PhD in Computer Science and Technology from the Nanjing University of Aeronautics and Astronautics in 2011. He is currently an Associate Professor in the College of Science and Technology of the Ningbo University. His research interests include internet of things, cyber physical system and formal methods.

1 Introduction

Conventional networks utilise special algorithms implemented on dedicated devices (hardware components) to manage routing paths and control the data flow. More complex routing devices can process data packets more meticulously. For example, a cisco router allows the users to mark out the priorities of different flows and allows more efficient flow control by prioritisation. Nowadays, networks are faced with increasing demand for scalability, security, reliability, network speed and even some special quality of service (QoS), such as real time character. Current network devices lack enough resource and flexibility to deal with these issues.

A possible solution to these problems is the usage of an innovative technology, called software defined networking (SDN) (Hu et al., 2014; Shu et al., 2016). SDN is a new networking paradigm that separates the control and data planes on the independent devices. A SDN usually comprises of two main hardware components: SDN Controller (SDN-C) and SDN forwarding unit (SDN-FU). One or a few SDN-Cs constitutes the control plane and SDN-FUs set up the data plane. Complex data handling algorithms are implemented on the control plane and its results can send to data plane to conduct data plane's running. Since data handling algorithms are located on controllers, they can be changed conveniently, without a re-programming or re-tasking on routers or switches.

In the current Internet environment, the mainstream network device is still legacy network equipment. We cannot discard current network devices and instead them with SDN-FUs at all, due to the budget constraints. In the meaning time, current SDN technology is new technology, there are some immature and incomplete. If we adopt them in large scale at once, there may be out of control about network running. So a practical solution is to let SDN component coexist with current Internet equipment (Jia et al., 2016). We call such a network as hybrid SDN (Vissicchio et al., 2014; Levin et al., 2013). According to Xu et al. (2017), we can upgrade a legacy network into hybrid SDN using two strategies. There are replaced deployment and incremental deployment. In replaced deployment, we just use SDN-FUs to replace legacy network device, which is a normal way. In incremental deployment, a set of additional SDN-FUs (and SDN-Cs) are deployed. In latter case, we can use few additional SDN-FUs to advance network performance, but the large part of network devices are still legacy equipment. The final network is always a hybrid SDN. In this paper, we focus on the former case. We want the legacy network to update gradually from hybrid SDN into a pure SDN at last. That is to say, we focus on the question of replaced deployment of SDN.

From industrial communications to real-time internet applications, there are always strict QoS requirements. Although QoS can be described by parameters such as bandwidth, delay, jitter and packet loss ratio, end-to-end delay is a common basic index, especially in wired communication (Ongaro et al., 2015; Guck et al., 2017; Zoppi et al., 2018). In this paper, when we upgrade the network using replaced deployment strategy based on SDN technology, we focus on the optimisation of end-to-end delay.

Network calculus is a system theory for communication networks. It presents a theoretical framework for analysing performance guarantees in computer networks. Specifically, worst-case bounds on delay and buffer requirements in a network can be computed (Le Boudec and Thiran, 2012). Deterministic bounds are for example necessary in real-time network, where packets have to reach their destination before a pre-defined deadline.

Guck et al. (2017) studies the question of the real-time QoS provisioning in SDN-based industrial environments. The paper proposes two network models based on network calculus theory, which can calculate or provide worst-case delay on one physical link. Our paper is an extensible research of Guck et al. (2017). We consider two new backgrounds. First, we extend the data flow range from one physical link to a computer network, where routing should be considered. Second, we research SDN replaced deployment in a hybrid SDN, instead of a pure SDN circumstance. So we face two new questions: How to order and select candidate legacy nodes to replace with SDN FUs? How to design proper algorithms on SDN controller to let data flow pass through a network with minimum delay?

The rest of this paper is organised as follows. Section 2 discusses the related works on SDN deployment and QoS question using SDN. Section 3 reviews the research basis of this paper, which are two network models. Section 4 discusses the replaced deployment problem and proposes an approximate method to deal with this challenge. In Section 5, we present a routing algorithm running on SDN controller to calculate next hop for SDN FU in a hybrid SDN. The experimental illustrations are presented in Section 6. We conclude the paper in Section 7.

2 Related works

Chen et al. (2017) studies the strategy to upgrade legacy switches to SDN switches for enterprise networks. The paper proposes a switch deployment strategy named HybridScore, which can provide the most control capacity over the network given a fixed number of switches to upgrade in hybrid SDN environment. HybridScore exploits the real traffic pattern of the network to determine the location of new SDN switches. Poularakis et al. (2017) studies the SDN upgrades in internet service providers (ISPs). The paper also considers that ISPs will gradually upgrade their networks to SDN over a period. The paper presents a general model which captures different migration costs and network topologies and devises algorithms through leveraging the theory of sub-modular and super-modular functions to optimise the process of SDN upgrades. Huang et al. (2018) further presents a survey of the deployment solutions and optimisation strategies for hybrid SDN networks. The paper systematically reviews solutions to control plane and data plane deployments and compares various optimisation strategies.

Tomovic et al. (2014) presents an original design of SDN/OpenFlow control environment, which provides bandwidth guarantees for priority flows in automated manner. In order to keep the whole network performance optimally, the paper proposes new routing algorithm considering both shortest path and path resource utilisation. This is an important probe for QoS routing algorithms and towards delay guaranteeing. Duan (2014) studies the inter-domain end-to-end QoS question. Service-oriented principle in SDN is used to solve the problem. Every autonomous networking system is abstracted as a network service. The composition of network services offers end-to-end services. The paper models the network service capability using a high-level abstraction model and presents a technique to determine required bandwidth in network services to achieve QoS guarantee. The modelling and analysis are all based on network calculus. Network slicing refers to the use of SDN and network function virtualisation to slice a network into logically isolated sub-networks. Each network slice can be defined with different properties such as latency and reliability guarantees. Kalor et al. (2018) uses network slices to dispose the different requirements in the scenarios of Industry 4.0 and exploits network calculus to assess the end-to-end properties of the network slices. Li et al. (2018) also studies the question of different delay requirements in the context of industry 4.0 and industrial internet of things. The paper proposes adaptive transmission architecture with SDN and edge computing and coarse/fine grained schemes are provided for low/high-deadline data streams.

Comparing to these related works, our work based on network calculus theory can calculate delay bounds more accurately and focus on using shorter delay for transmission of data packet during SDN upgrades. So our approaches can be used suitably for real-time internet and the emerging networking standards-time-sensitive networking (TSN).

3 Research basis: network models

3.1 Theory basics and preliminaries

In this section, we review main conclusions of Guck et al. (2017) and use these as the basis of our research.

Guck et al. (2017) presents two network models in order for real-time QoS provisioning in SDN-based industrial environments. Through the exploitation of these models on SDN controller, it is guaranteed accurate and fast control of worst-case delay of a physical link. Both network models are based on network calculus. The first model, the multi-hop model (MHM), assigns a rate and a buffer budget to each queue in a physical link of the network, which can compute worst-case delays for any path in the network. The second model, which is the threshold-based model (TBM), fixes a maximum delay for each queue in a physical link of the network, through automatically adapting the allocation of rate and buffer capacity based on the type of traffic. TBM has the potential to outperform MHM in accepting more flows and hence increasing network utilisation.

In this paper, we adopt a queue link network topology same as Guck et al. (2017). We use P to denote the physical link graph and E to refer to the set of edges of the graph. A physical link $(u, v) \in P_E$, where u and v are the nodes of an edge in the graph. We assume a non-preemptive strict priority scheduler with $Q_{u,v}$ queues at the physical link (u, v). Edges in the queue link network are denoted by (u, v, p), where (u, v) is the corresponding physical link and $p \in \{1, \ldots, Q_{u,v}\}$ is the priority of the corresponding queue at the physical link. Queue $p = 1$ is the highest priority queue and queue $p = Q_{u,v}$ being the lowest priority. $R_{u,v}$ denotes the capacity of the physical link (u, v). $T[u, v, p]$ represents the worst-case delay of the queue link edge (u, v, p).

3.2 Multi-hop model

In MHM, there are three assumes:

1 The packet size of a flow cannot be greater than the maximum packet size $L^{\max}$ in the network.

2 Each queue link edge is allocated a data rate $A_R[u, v, p]$.

3 The maximum allowed burst at a queue is fixed and denoted as $M_B[u, v, p]$.

Under these assumes, we can make the following conclusion:

$$T[u, v, p] \leq \frac{\sum_{j=1}^{p} M_B[u, v, j] + 2 \cdot L^{\max}}{R_{u,v} - \sum_{j=1}^{p-1} A_R[u, v, j]}$$

We use $T^{MHM}[u, v, p]$ to denote the upper bound of the worst-case delay $T[u, v, p]$. So,

$$T^{MHM}[u, v, p] = \frac{\sum_{j=1}^{p} M_B[u, v, j] + 2 \cdot L^{\max}}{R_{u,v} - \sum_{j=1}^{p-1} A_R[u, v, j]} \tag{1}$$

3.3 Threshold-based model

In TBM, the worst-case delay of each queue is simply fixed by defining a threshold $T^{TBM}[u, v, p]$. That is to say,

$$T[u, v, p] \le T^{TBM}[u, v, p] \tag{2}$$

We use $U_B[u, v, p]$ denoting the sum of the bursts of the flows routed through the queue link edge. $U_R[u, v, p]$ represents the sum of the rates of the flows routed through the queue link edge. $l_{u,v,p}^{\max}$ describe the maximum packet size of the aggregate flow traversing the queue link edge. When we let best effort traffic flow through the lowest priority queue and use $L^{\max}$ to approximately denote its maximum packet size. We can make the following conclusion:

$$T[u, v, p] \le \frac{\sum_{j=1}^{p} U_B[u, v, j] + L^{\max} + l_{u,v,p}^{\max}}{R_{u,v} - \sum_{j=1}^{p-1} U_R[u, v, j]} \tag{3}$$

4 Design of replaced SDN-FU deployment

There are two kinds of SDN deployment strategies, which are replaced deployment and incremental deployment. Our paper discusses the replaced deployment problem. In this strategy, we order the legacy nodes to be updated by some rules and update the top nodes into SDN-FUs considering the factors of limited budget and system dependability. These SDN-FUs will be controlled by at least one SDN-C. Since the SDN-C will not participate into data forwarding, this work ignores the deployment of SDN-C.

In this paper, we assume that each legacy node is a router running the legacy routing protocol, e.g., OSPF. We also assume that SDN-FUs are hybrid devices (e.g., OpenFlow hybrid switches) that run traditional network protocol stack while also being controlled by the SDN-C. So SDN-FUs can interact with non-SDN routers and moreover, routing decisions taken by the SDN-C are preferred over that by distributed OSPF protocols.

Generally, the data rate of each queue link edge $A_R[u, v, p]$ is not fixed. The value of $A_R[u, v, p]$ is affected by the capacity, traffic load (and priority), which can be available by OSPF-TE (Katz et al., 2003). In this paper, we consider long-term traffic estimation for the data rate and set each queue link edge a default data rate. In this background, we can use MHM to calculate the delay of a queue link edge starting from a legacy network node. While a legacy node updates to a SDN FU, we control the node working according to TBM and we calculate the delay of a queue link edge sourcing from a SDN FU by TBM.

In this paper, we order legacy nodes according to their effects on delay. For simplicity, we assume that every physical link occupies with same number of queue link edges and note as h. Also, we assume that every queue link edge in a physical link has same $M_B[u, v, p]$ and $A_R[u, v, p]$.

From formula (1), the higher the priority p, the longer the delay under MHM. Since p = h is used to denote the flow of best effort transferring, we can use the delay of queue link edge p = h – 1 to represent the delay of the corresponding physical link. In (1), letting p = h – 1, we can obtain:

$$ST^{MHM}[u, v] = \frac{\sum_{j=1}^{h-1} M_B[u, v, j] + 2 \cdot L^{\max}}{R_{u,v} - \sum_{j=1}^{h-2} A_R[u, v, j]} \tag{4}$$

where $ST^{MHM}[u, v]$ represents the worst case delay of the physical link (u, v) sourcing from a legacy node.

Generally, starting from node u, there are several physical links. The total influence of delay for a node u can be calculated as the sum of delay of all these physical links. So we can have:

$$TT^{MHM}[u] = \sum_{j=1}^{M} ST^{MHM}[u, v_j] \tag{5}$$

where M is the number of nodes adjacent to node u in a network.

When we obtain the total influence of delay for a node u using formula (4) and (5), we can queue the nodes using $TT^{MHM}[u]$ with descending order and update some of the nodes to SDN FUs according to budget permission and consideration of system dependability.

5 Design of routing in HSDN

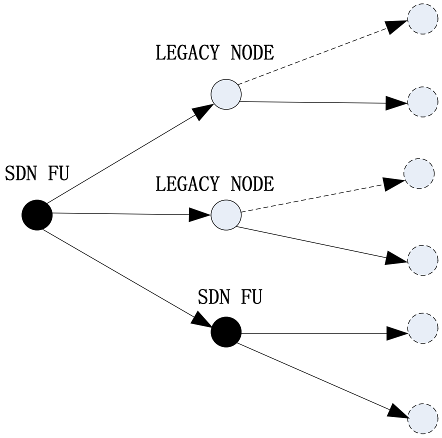
For a legacy router, it selects next hop for a flow according to a fixed routing algorithms such as OSPF. For a SDN-FU, we can design an individualised routing algorithm running on the corresponding SDN-C. SDN-C can send the running result to SDN-FU and help SDN-FU to determine next hop for a flow. In this paper, we focus on the transferring delay and aim to forward flows to destinations with minimum delay. We propose an algorithm for SDN-C to achieve this goal in this section.

When we replace some legacy network nodes with SDN-FUs, there are SDN-FUs and legacy routers co-existing in a network. Figure 1 is a part of network starting from a SDN-FU.

In Figure 1, starting from a node, there are several adjacent nodes. For a non-SDN-FU, its routing obeys OSPF protocol. From the point of view by this non-SDN-FU, all other nodes also obey OSPF protocol (including SDN-FU) and the algorithm of computing shortest path, such as Dijkstra algorithm, can be utilised. So, a non-SDN-FU can determine its next hop by computing shortest path from it to the destination node.

For a SDN-FU, all its adjacent nodes are the candidate next hop. SDN-FU will select the adjacent node through which minimum delay can be gotten from this SDN-FU to the destination node.

Figure 1 Part of network starting from a SDN-FU



For the calculation of the delay of a physical link, the basic methods are different for source nodes with legacy node and SDN-FU. The worst case delay for legacy node obeys MHM and we can use formula (4) to denote the delay of a physical link (u, v) as Section 4. The worst case delay for a SDN-FU follows TBM. In the TBM, for a queue link edge, its worst case delay is fixed. Let's select the values in a moment of its running and notes its U_B and U_R as U_{B_0} and U_{R_0}. From formulas (2) and (3), we can evaluate the worst case delay of a queue link edge sourcing from SDN-FU as following:

$$T^{TBM}[u, v, p] = \frac{\sum_{j=1}^{p} U_{B_0}[u, v, j] + L^{\max} + l_{u,v,p}^{\max}}{R_{u,v} - \sum_{j=1}^{p-1} U_{R_0}[u, v, j]} \tag{6}$$

Similarly as in MHM, we also use the worst case delay of queue link edge p = h – 1 as the represent value of its physical link. In (6), letting p = h – 1, we can obtain:

$$ST^{TBM}[u, v] = \frac{\sum_{j=1}^{h-1} U_{B_0}[u, v, j] + L^{\max} + l_{u,v,h-1}^{\max}}{R_{u,v} - \sum_{j=1}^{h-2} U_{R_0}[u, v, j]} \tag{7}$$

where $ST^{TBM}[u, v]$ can be used to evaluate the worst case delay of the physical link (u, v) sourcing from SDN-FU.

Based on above discussion, we propose a routing algorithm to determine the next hop for a SDN-FU. The algorithm is running on a SDN-C connected with the SDN-FU.

In Algorithm 1, we can implement the shortest path algorithm (Dijkstra algorithm) using the running time of $O(V^2)$. We denote the maximum queue number of a physical link with h just as before. Also we note the number of nodes and edges of our network as N and V. Then our algorithm's running time is $O(N^2 \cdot \max\{V^2 + h, N\})$.

Algorithm 1 Calculate next hops for SDN-FUs controlled by a SDN-C aiming at minimum delay

```
Input: The set of SDN-FUs $S$ controlled by a SDN-C; The physical link graph $P_E$.
Output: An array $ArrayNode$ storing next hop for every SDN-FU of set $S$.
Main procedure:
    Initialise $ArrayNode$; Initialise integer i = 0;
    For each node $s0 \in S$
        {$t_0$ = Next hop return from $NextHopandDelay(s_0)$;
        $ArrayNode[i].sourceNode = s_0$; $ArrayNode[i].NextHop = t_0$;
        i++;}
Function:
    Node&DelayofNode $NextHopandDelay(s_0)$
        {switch (type of node $s0$) {
        case (Legacy Node):
            {$weight(u, v) = 1$;
            $t_0 = NextHopInShortestPath(s_0)$;
            $Delay(s_0) = ST^{MHM}[s_0, t_0]$ + delay return from $NextHopandDelay(t_0)$;
            break; }
        case (SDN FU):
            {$T = \{t| (s_0, t) \in P_E\}$; $DelayOfNodes_0 = \infty$;
                For each node $t \in T$
                    {$InterDelay = T^{IBM}[s_0, t]$ + delay return from $NextHopandDelay(t)$;
                    if ($DelayOfNodes_0 > InterDelay$) {$DelayOfNodes_0 = InterDelay$; $t_0 = t$;}
                    }
                break;}}
            return $t_0$ and $DelayOf Nodes$.
        }
```

6 Evaluation

In this section, we evaluate our deployment strategy and routing algorithms by our fine designing and analysing.

6.1 Simulation setting

In our simulation, we use the method and tool from The Network Topology from the Monash University. We select a typical topology, denoted by Figure 2, contains ten routers and 14 links.

In this simulation, for simplicity, resources and parameters are allocated among queues identically for each link. Each legacy router in the network will select the successor on its own shortest path to the destination and each SDN-FU will select the successor according to Algorithm 1 of Section 5. In our simulation, we let a flow pass through a series of physical links with fixed priority of p = h – 1. We assume h = 5, so p = 4. The concrete values of parameters are listed on Table 1, basically cited from (Ongaro et al. (2015), with $L^{\max}$ =1542B.

Figure 2 Our simulation topology

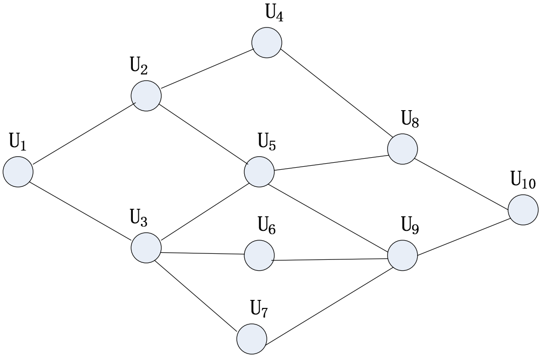


Table 1 The values of related parameters

$M_B[u, v, \cdot]$	$A_R[u, v, 1]$	$A_R[u, v, 2]$	$A_R[u, v, 3]$	$A_R[u, v, 4]$	$R_{u,v}$	$T^{TBM}[u, v, 4]$
60 kB	51.2 MB/s	24.622 MB/s	8.349 MB/s	3.953 MB/s	1 Gbps	0.2254 ms

6.2 Simulation process and result

To verify the effectiveness of our deployment strategy, we compare our node selection method with random scheme. In the latter, nodes are randomly selected to upgrade to SDN FUs in the network. As the result is apparent, we ignore the simulation and discussion about this question.

To evaluate the rationality of our routing algorithm and the node selection strategy, we let a flow pass through physical links according to following steps:

1. in a legacy network without SDN-FU
2. select and replace minimum nodes to SDN-FUs and let the flow pass through one SDN-FU to destination
3. select and replace minimum nodes to SDN-FUs and let the flow pass through two SDN-FUs to destination
4. repeat above updating and passing through rhythm, until all nodes have upgraded from legacy routers to SDN-FUs.

Table 2 is the delays the flow passing through the network.

In fact, the value of $T^{TBM}[u, v, 4]$ is fixed manually, we can adjust its value properly. Then we can get the following group of delays as Table 3. In Table 2, $T^{TBM}[u, v, 4]$ is fixed at 0.2254 ms and its value adjusted to 0.2032 ms in Table 3. When the same flow in the network passes through one to four SDN units, the corresponding delays from Table 2 to Table 3 are decreased accordingly. The reason is that $T^{TBM}[u, ,v, 4]$ fixes the worst-case delay of each physical link sourcing from a SDN-FU and then directly affects the whole delay of a flow.

Table 2 The delays of a flow passing through the network under parameters in Table 1 (unit: ms)

None	*One SDN unit*	*Two SDN units*	*Three SDN units*	*Four SDN units*
1.0617	1.0216	0.9816	0.9416	0.9016

Table 3 The delays of a flow passing through the network under $T^{TBM} = [u, v, 4] = 0.2032$ ms (unit: ms)

None	*One SDN unit*	*Two SDN units*	*Three SDN units*	*Four SDN units*
1.0617	0.9994	0.9372	0.8750	0.8128

Based on the parameter values of Table 1, we adjust the value of $R_{u,v}$ from 1 Gbps down to 0.8 Gbps this time and get another group of delays as Table 4. Since $T^{TBM}[u, v, 4]$ has the same fixed value (0.2254 ms), when the flow passes through a physical link sourcing from SDN-FU, the delay of the physical link is same. But when the flow passes through a link sourcing from legacy node, as $ST^{MHM}[u, v]$ will increase according to formula (4), the delay of the physical link is ascent. The delays in Tables 2 and 4 are the whole delays of a flow from U1 to U10 in fig. 2 under different intermediate nodes.

Table 4 The delays of a flow passing through the network under $R_{u,v} = 0.8$ Gbps (unit: ms)

None	*One SDN unit*	*Two SDN units*	*Three SDN units*	*Four SDN units*
1.3583	1.2442	1.1300	1.0158	0.9016

It should be noted that we run these simulations on the personal computer. When a high-performance server runs the simulations, the running time will be reduced significantly.

7 Conclusions

This paper focuses on SDN replaced deployment problem aiming at real-time QoS provisioning. Based on previous research originated from the theory of network calculus, we present the strategy for selecting legacy nodes to upgrade to SDN forward units and the routing algorithms of controlling the running of SDN-FUs. In the process of node selection, we estimate the influence of legacy nodes in delay based on MHM and upgrade nodes occupying high influence within budget and dependability. Also, in the routing algorithm, we are inclined to choose next hops for SDN forward units, which can undergo less transmission delay to destination. In this algorithm, we use TBM to estimate the delay of a physical link sourcing from a SDN forward unit. We illustrate our ideas through simulation. In our simulation, the delays of a flow passing through a small network can reduce significantly with the usage of SDN forward units. We will evaluate our ideas under large scale as our next work and especially when a number of flows across a physical link and the number of flows more than queue number of a physical link.

Acknowledgements

This work is supported by Natural Scientific Foundation of China (Grant No. 61272083, 61262002) and sponsored by K.C. Wong Magna Fund in Ningbo University.

References

Hu, F., Hao, Q. and Bao, K. (2014) 'A survey on software-defined network and openflow: from concept to implementation', *IEEE Communication Surveys and Tutorials*, Vol. 16, No. 4, pp.2181–2206.

Shu, Z., Wan, J., Li, D., Lin, J., Vasilakos, A. and Imran, M. (2016) 'Security in software-defined networking: threats and countermeasures', *Mobile Networks and Applications*, Vol. 21, No. 5, pp.764–776.

Jia, X., Jiang, Y. and Guo, Z. (2016) 'Incremental switch deployment for hybrid software-defined networks', *41st Conference on Local Computer Networks*, IEEE, pp.571–574.

Vissicchio, S., Vanbever, L. and Bonaventure, O. (2014) 'Opportunities and research challenges of hybrid software defined networks', *ACM SIGCOMM Comput. Commun. Rev.*, Vol. 44, No. 2, pp.70–75.

Levin, D., Canini, M., Schmid, S. and Feldmann, A. (2013) 'Incremental SDN deployment in enterprise networks', in *Proc. SIGCOMM*, pp.473–474.

Xu, H., Li, X., Huang, L., Deng, H., Huang, H. and Wang, H. (2017) 'Incremental deployment and throughput maximization routing for a hybrid SD', *IEEE/ACM Trans. Netw.*, Vol. 23, No. 3, pp.1861–1875.

Ongaro, F., Cerqueira, E., Foschini, L., Corradi, A. and Gerla, M. (2015) 'Enhancing the quality level support for real-time multimedia applications in software-defined networks', *International Conference on Computing, Networking and Communications, Communication QoS and System Modeling Symposium*, pp.505–509.

Guck, J., Van Bemten, A. and Kellerer, W. (2017) 'DetServ: network models for real-time QoS provisioning in SDN-based industrial environments', *IEEE Transactions on Network and Service Management*, Vol. 14, No. 4, pp.1003–1017.

Zoppi, S., Van Bemten, A., Murat Gürsu, H., Vilgelm, M., Guck, J. and Kellerer, W. (2018) 'Achieving hybrid wired/wireless industrial networks with WDetServ: reliability-based scheduling for delay guarantees', *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 5, pp.2307–2319.

Le Boudec, J.Y. and Thiran, P. (2012) *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, April, Springer, Heidelberg, Germany.

Chen, M., Wang, W., Chung, I. and Chou, C. (2017) 'Incremental hybrid SDN deployment for enterprise networks', *IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pp.1143–1149.

Poularakis, K., Iosifidis, G., Smaragdakis, G. and Tassiulas, L. (2017) 'One step at a time optimizing SDN upgrades in ISP networks', *IEEE INFOCOM*, pp.1–9.

Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T. and Hu, S. (2018) 'A survey of deployment solutions and optimization strategies for hybrid SDN networks', *IEEE Communications Surveys and Tutorials*, p.1, IEEE Early Access Articles.

Tomovic, S., Prasad, N. and Radusinovic, I. (2014) 'SDN control framework for QoS provisioning', *22nd Telecommunications Forum Telfor (TELFOR)*, pp.111–114.

Duan, Q. (2014) 'Network-as-a-service in software-defined networks for end-to-end QoS provisioning', *23rd Wireless and Optical Communication Conference (WOCC)*, pp.1–5.

Kalor, A.E., Guillaume, R., Nielsen, J.J., Mueller, A. and Popovski, P. (2018) 'Network slicing in industry 4.0 applications abstraction methods and end-to-end analysis', *IEEE Transactions on Industrial Informatics*, Early Access.

Li, X., Li, D., Wan, J., Liu, C. and Imran, M. (2018) 'Adaptive transmission optimization in SDN-based industrial internet of things with edge computing', *IEEE Internet of Things Journal*, Vol. 5, No. 3, pp.1351–1360.

Katz, D., Kompella, K. and Yeung, D. (2003) *Traffic Engineering (TE) Extensions to OSPF Version 2*, Document RFC 2370, Network Working Group, September.

The Network Topology from the Monash University [online] http://www.ecse.monash.edu.au/twiki/bin/view/InFocus/LargePacket-switchingNetworkTopologies (accessed 20 June 2015).