Enabling Emergency Flow Prioritization in SDN Networks

Jerico Moeyersons*, Behrooz Farkiani†, Bahador Bakhshi†, Seyyed Ali Mirhassani‡, Tim Wauters*, Bruno Volckaert* and Filip De Turck*
*IDLab, Department of Information Technology
Ghent University - imec, Ghent, Belgium
Email: jerico.moeyersons@ugent.be
†Computer Engineering Department and ‡Mathematics and Computer Science Department
Amirkabir University of Technology, Tehran, Iran

***Abstract*—Emergency services must be able to transfer data with high priority over different networks. With 5G, slicing concepts at mobile network connections are introduced, allowing operators to divide portions of their network for specific use cases. In addition, Software-Defined Networking (SDN) principles allow to assign different Quality-of-Service (QoS) levels to different network slices.**

This paper proposes an SDN-based solution, executable both offline and online, that guarantees the required bandwidth for the emergency flows and maximizes the best-effort flows over the remaining bandwidth based on their priority. The offline model allows to optimize the problem for a batch of flow requests, but is computationally expensive, especially the variant where flows can be split up over parallel paths. For practical, dynamic situations, an online approach is proposed that periodically recalculates the optimal solution for all requested flows, while using shortest path routing and a greedy heuristic for bandwidth allocation for the intermediate flows.

Afterwards, the offline approaches are evaluated through simulations while the online approach is validated through physical experiments with SDN switches, both in a scenario with 500 best-effort and 50 emergency flows. The results show that the offline algorithm is able to guarantee the resource allocation for the emergency flows while optimizing the best-effort flows with a sub-second execution time. As a proof-of-concept, a physical setup with Zodiac switches effectively validates the feasibility of the online approach in a realistic setup.

***Index Terms*—Network Management, SDN, ILP, LP, Dijkstra, Emergency flows**

I. Introduction

With the expected release of 5G by the end of 2019 [1], slicing concepts at network level will be introduced [2], [3], to allow network operators to provide portions of their networks for specific use cases such as Internet of Things (IoT), streaming videos and smart energy grids.

Network slicing is a virtual networking mechanism that is part of the same family as Software-defined Networking (SDN) and Network Functions Virtualization (NFV), two closely related network virtualization technologies that are moving networks towards automation through software. SDN is an important technology to implement dynamic and flexible network management by separating the data plane from the control plane in networks [4]. Every SDN switch (further called switch) within an SDN network acts like a simple packet

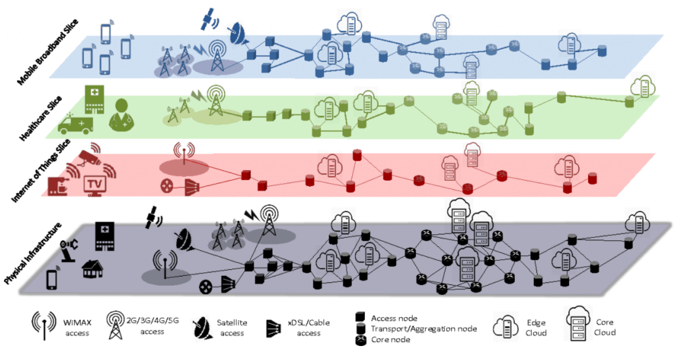

Fig. 1: 5G network slices running on a common underlying multi-vendor and multi-access network. Each slice is independently managed and addresses a particular use case [3].

forwarding device that is controlled by a logically centralized software program, the SDN controller. NFV on the other hand separates network functions from the underlying proprietary hardware appliances [5]. The network functions running on dedicated hardware are thus transferred to software-based applications running in datacenters, network nodes, end-user premises etc. Complementary to SDN and NFV, network slicing allows the creation of multiple virtual networks atop of a shared physical infrastructure whereby each virtual network has its own specific features depending on the use case. An example of different possible network slices together is illustrated in Fig. 1.

The network slicing concept introduces the possibility to enable new features such as more fine-grained Quality-of-Service (QoS). In this paper we therefore propose an offline and online SDN-based solution to guarantee and optimize flows based on their priority by creating bandwidth meters who are responsible for limiting the maximum allowed bandwidth per flow.

Different emergency events are taken as use cases. During such an event, it is required to prioritize the network traffic that is coming from and going to the emergency services in the presence of large civilian crowds in order to coordinate the relief and response. Every emergency event is different, and needs different types of network traffic as some situations require high bandwidth for streaming high resolution video feeds and other situations only require a low amount of bandwidth to enable communication or streaming low resolution video feeds. The bandwidth for the emergency traffic should be guaranteed while the network traffic coming from (non-prior) users should be optimized over the remaining available bandwidth. The proposed solution in this paper guarantees the bandwidth of emergency network traffic by generating SDN high-priority flows while other non-priority traffic will receive best-effort resources based on their priority within the network. To evaluate the solution, a topology is simulated whereby the routing of different best-effort flows and emergency flows is optimized. Afterwards, a tree topology is built with Zodiac switches [6] and used for the practical evaluation of the algorithm on a smaller and real topology, controlled by the Ryu SDN controller software [7].

This paper contributes to three main topics: *(i)* design of models for guaranteeing bandwidth for emergency flows while optimizing best-effort flows over the remaining network resources, *(ii)* design of a joint online-offline approach to practically implement the model and *(iii)* the validation of the model through both simulation and practical evaluation. The remainder of this paper is organized as follows: Section II presents related work. In Section III, the problem description is given followed by the problem formulation for both splittable and unsplittable flows. Section IV presents the evaluation methods and the corresponding results. Finally, Section V discusses conclusions and future avenues of research.

II. Related Work

There has been a great deal of research on providing QoS in SDN over the past few years [8]. The authors in [9], [10] presented algorithms to provide QoS, but without considering bandwidth guarantees. In [9] authors proposed a QoS solution based on SDN technology. The authors first defined a cost function which assigns a positive value to each link based on the length, bandwidth and the weight of the link. Then, they utilized the Dijkstra algorithm [11] to find multiple paths for each source and destination pair in the network. When a flow arrives, the path with the lowest congestion is selected as the routing path for the flow. Zhang et al. [10] proposed a QoS framework based on the OpenFlow protocol which dynamically calculates a path for each flow. If the flow is a QoS-required flow, an algorithm based on Dijkstra is used to find a path with the minimum delay and cost values.

An approach to allocate bandwidth and satisfy QoS requirements is presented in [12]. The authors categorized flows into QoS and best effort flows and defined a metric, used in path selection, that considers the requested rates. Shaohua et al. [13] categorized cloud applications into three levels based on the sensitivity to delay and bandwidth. The flow-based adaptive routing algorithm is presented in [14] which utilizes Dijkstra and K-shortest path [8] algorithms with the aim of maximizing the utilization of network resources. The authors evaluated the proposed algorithm through simulation. Pinto et al. [15] defined four service classes including best effort and bandwidth guaranteed classes. Each new flow is first assigned to the probing class and its behavior is monitored. After some time, if the network can support its bandwidth along the path it will be reassigned to the bandwidth guaranteed class or otherwise the best effort class. The authors in [16] designed a method to provide bandwidth guarantees by using OpenFlow meters and queues. The authors categorized flows into QoS flows which have minimum guaranteed bandwidth and best effort flows with no requirements. For each QoS flow, first, an admission control process checks whether there is a path that can accommodate the flow rate. After that, by using a meter at the ingress switch, the input rate of the flow is monitored and if it exceeds the defined rate, the packets will be marked. Using three different queues at the egress port of each switch along the path for marked and unmarked QoS and best effort flows, traffic prioritization is made possible. MPLS tunnels are used in [17] to provide end-to-end bandwidth guarantees. Similar to [16] the authors used OpenFlow meters at the ingress switches. For each flow the input rate of the flow is monitored and based on that, a priority value is set in the header of each packet. Then, an MPLS tunnel is used to route the packets toward the egress switch and the priority of each packet specifies its output queue. Lu et al. [18] utilized preplanned network slices to both satisfy QoS requirements and maximize the overall throughput of the network. The authors used the traffic history to create network slices which have fixed configurations during the network lifetime. When a flow arrives, it is assigned to a slice by using the VLAN ID of the slice. The MaxStream framework is proposed in [19] to maximize the number of streaming sessions and bandwidth provisioning. The authors formulated two ILP problems. The first problem maximizes the number of accepted flows by considering the requested rate of the flows. Then, the set of accepted flows is used in the second problem to maximize the total rate of the accepted flows. Since the authors focused on multimedia streams, they ignored best effort flows with no QoS requirements.

The most important studies are summarized in Table I. In this paper, we utilize both online and offline approaches to provide bandwidth guarantees for emergency flows and maximize the total rate of best effort flows. The offline approach optimizes all existing emergency and best-effort flows while the online approach routes and allocates new incoming flows sub-optimally in between offline batches. The offline approach is defined as two models, a Linear Programming (LP) and an Integer Linear Programming (ILP) model and the online approach is based on Dijkstra to route new incoming flows along with a greedy heuristic for bandwidth allocation. We also use OpenFlow meters to implement our method and evaluate it using Zodiac switches [20]. Only drop meter policies are used in this paper because the current OpenFlow versions [21] do not support other policies such as 2-color-marking [22] and 3-color-marking [23].

III. Problem Description and Formulation

In this section, the problem described in Section I is first analyzed in detail. Afterwards the offline approach con-

TABLE I: Summary of Related Research

Reference	Offline/Online	Objective	Path Selection	Evaluation
[12]	Online	Satisfy the QoS requirements of the QoS flows	Greedy	Geni Testbed [24]
[15]	Online	Admission control and traffic management	Sink tree	Mininet [25]
[16]	Online	Satisfy the minimum bandwidth requirements of QoS flows	Widest shortest path	Open vSwitch [26] and physical switches
[17]	Online	Guarantee bandwidth of QoS flows	SAMCRA [27] and Dijkstra	Mininet and physical switches
[19]	Online	Maximize the number of streaming sessions and bandwidth provisioning	Two ILP problems	Mininet
This Paper	Online & Offline	Maximize the total rate of the best effort flows and guarantee bandwidth of high priority flows	Dijkstra (online), ILP problem (offline)	Physical switches

sisting of two formulations is presented, able to guarantee emergency flows while the best-effort flows are optimized over the remaining bandwidth in the network. Finally, the online approach is described which is able to run in practical environments. It combines the offline approach with a sub-optimal solution to handle new incoming flows in between the calculations of the offline approach.

A. Problem Description

Within an SDN network, there are different OpenFlow-enabled switches connected to one or more SDN controllers and a set of best-effort flows responsible for the correct routing of the network traffic. Each flow is described using a tuple `<source, destination, class>` whereby the class describes the traffic class of the flow. Each traffic class has a priority value and enforces the lower and upper bound bandwidth rates of the flows assigned to it. At a certain moment, emergency flows are requested for prioritizing network traffic coming from and going to different emergency services. These emergency flows need to be satisfied by guaranteeing the requested bandwidth while the remaining network bandwidth should be allocated to the best-effort flows. The optimization of the best-effort flows depends on the priority of the traffic classes where a higher priority requires a larger share of the available network bandwidth.

This paper answers to the question about how to maximize the total input rate of the best-effort flows in the network while the requested rates of the emergency flows are satisfied and the bandwidth capacity constraints of the network are respected. We assume that the requested rate for the emergency flows is not higher than the total available rate in the network. To solve this problem, two offline models are defined and evaluated.

In the first model, defined by means of an ILP formulation, we assume that flows cannot be split up and each flow needs to be assigned to a single path from source to destination. In the second model, defined by means of an LP formulation, we assume that flows can be split up, allowing traffic to be separated over different links, optimizing the bandwidth resource allocation [28]. The packet reordering effect that can occur when using flow splitting can be mitigated using hash-bashed splitting and packet tagging [29].

TABLE II: Notations Summary

Variables	
$y^i_{u,v}$	Equals 1 if the traffic for flow i passes through link (u, v)
R^i	The rate assigned to best effort flow i
Parameters	
$F \equiv M \cup B$	Set of all flows
M	Set of all emergency flows
B	Set of all best effort flows
$G = (V, E)$	The graph of the network. V is the set of nodes and E is the set of physical links. All links are bidirectional with different capacity in each direction
$Z^i_{u,v}$	The rate of flow i on link (u, v)
$Cap^{(u,v)}$	The bandwidth of the link between u and v, in the direction from u to v
W_i	The weight assigned to flow i based on the traffic class it belongs to
τ_i	The requested rate for emergency flow i
$minRate^i$ $maxRate^i$	The lower bound and upper bound for the rate of flow i based on the traffic class it belongs to.
$Source\,(i)$ $Destination\,(i)$	The source and the destination of flow i

The formulations of these two offline models are provided in Section III-B and Section III-C. Notations used in the formulations are summarized in Table II. Some described constraints contain a multiplication of a continuous and a binary variable and because this cannot be directly solved by state-of-the-art solvers, they need to be linearized first. These formulations will optimize the best-effort flows over the remaining bandwidth that is not used by emergency flows. In case the offline models are not able to run in real-time, the online approach manages new incoming flows in between offline batches, providing the shortest (but possible sub-optimal) path with a greedy-based solution to allocate bandwidth to these new flows.

B. The ILP formulation

$$\max \sum_{i\in B|\{u=Source(i),(u,v)\in E\}} W_i \times Z^i_{u,v} \tag{1}$$

Subject to:

$$\sum_{(u,v)\in E} y^i_{u,v} - \sum_{(v,u)\in E} y^i_{v,u} = \begin{cases} 1 & u = Source\,(i) \\ 0 & \text{otherwise} \\ \text{-1} & u = Destination\,(i) \end{cases} \quad \forall u \in V,\ i \in F \tag{2}$$

$$\sum_{i\in B} y^i_{u,v} \times R^i + \sum_{i\in M} y^i_{u,v} \times \tau_i \leqslant Cap^{(u,v)} \tag{3}$$

$$\sum_{(u,v)\in E} y^i_{u,v} \leqslant 1 \ \ \forall u \in V,\ i \in F \tag{4}$$

$$\sum_{(v,u)\in E} y^i_{v,u} \leqslant 1 \ \ \forall u \in V,\ i \in F \tag{5}$$

$$R^i \in \left[minRate^i,\ maxRate^i\right] \tag{6}$$

$$y^i_{u,v} \in \{0,1\} \tag{7}$$

$$Cap\,(u,v) \geqslant \tau_i \ \ \forall i \in M \tag{8}$$

The objective of the ILP formulation is to maximize the sum of traffic rates of best-effort flows multiplied by their assigned weights, illustrated in (1), subjected to constraints [(2) - (5)]. (2) is the flow conservation constraint which guarantees a path from source to destination. (3) enforces the capacity limit of each physical link and (4) and (5) are used to prevent loops as much as possible. (6) and (7) specify the bounds for the assigned rate and whether or not traffic is passing through link `(u, v)`. Finally in (8), the assumption is made that the network is at least able to handle all requested emergency flows.

The second constraint contains a multiplication of a continuous and a binary variable as in $\sum\limits_{i\in B} y^i_{u,v} \times R^i$. The constraint can be linearized as follows:

$$Z^i_{u,v} \leqslant Cap^{(u,v)} \times y^i_{u,v} \tag{9}$$

$$Z^i_{u,v} \leqslant R^i \tag{10}$$

$$R^i + Cap\,(u,v) \times y^i_{u,v} - Z^i_{u,v} \leqslant Cap^{(u,v)} \tag{11}$$

$$Z^i_{u,v} \in \left[0,\ \ maxRate^i\right] \tag{12}$$

C. The LP formulation

$$\max \sum_{i\in B} W_i \times R^i \tag{13}$$

Subject to:

$$\sum_{(u,v)\in E} y^i_{u,v} - \sum_{(v,u)\in E} y^i_{v,u} = \begin{cases} -R^i & u = Source\,(i) \\ 0 & \text{otherwise} \\ -R^i & u = Destination\,(i) \end{cases} \quad \forall u \in V,\ i \in B \tag{14}$$

$$\sum_{(u,v)\in E} y^i_{u,v} - \sum_{(v,u)\in E} y^i_{v,u} = \begin{cases} \tau & u = Source\,(i) \\ 0 & \text{otherwise} \\ -\tau_i & u = Destination\,(i) \end{cases} \quad \forall u \in V,\ i \in M \tag{15}$$

$$\sum_{i\in F} y^i_{u,v} \leqslant Cap^{(u,v)} \tag{16}$$

$$R^i \in \left[minRate^i,\ maxRate^i\right] \tag{17}$$

$$y^i_{u,v} \in \mathbb{R}_{\geqslant 0} \tag{18}$$

$$Cap\,(u,v) \geqslant \tau_i \ \ \forall i \in M \tag{19}$$

The LP formulation uses the principles of flow splitting to solve the described problem. The objective is again to maximize the sum of the traffic rates of the best-effort flows multiplied by their assigned weights, illustrated in (13), subjected to constraints [(2) - (4)]. (14) and (15) are the flow conservation constraints for the best effort and emergency flows respectively. (16) enforces the bandwidth capacity limits of physical links. The LP formulation is solvable in polynomial time [30], [31].

D. Online approach

As shown later in Section IV, the LP model is much faster than the ILP model and could be used in a near-real-time scenario. However, the use of the slower ILP model is obliged if the network does not support flow splitting, which is for example the case when it is built with SDN switches from Northbound Networks [6]. In this case, an alternate online approach is necessary for handling new incoming flows in between calculations of the ILP model.

The offline batches are responsible for guaranteeing the emergency flows and optimizing the remaining best-effort flows. In case a new flow arrives during the calculation of the offline batches, the shortest path between the source and destination is determined by using Dijkstra's algorithm [32]. New incoming emergency flows obtain their requested bandwidth while newly arriving best-effort flows are assigned to the average best-effort traffic class. In case there is no bandwidth available for the new flow, the greedy heuristic calculates the bandwidth and decreases the other best-effort flows bandwidth based on their priority, as illustrated in Algorithm 1. The complete online approach, following a greedy approach, is described in Algorithm 2.

Algorithm 1 Greedy heuristic

```
$\tau \leftarrow$ requested bandwidth
$C \leftarrow$ traffic classes sorted by priority (low to high)
for $i$ in $count(C) - 1$ do
    $a \leftarrow \tau/2$
    $\tau \leftarrow \tau/2$
    $band[i] \leftarrow a$
end for
$band[count(C) - 1] \leftarrow \tau$
for each traffic_class in $C$ do
    $i \leftarrow index$
    $s \leftarrow$ number of meters in traffic_class
    for each $meter$ in traffic_class do
        $meter \leftarrow meter - band[i]/s$
    end for
end for
```

Algorithm 2 Online approach

```
$R \leftarrow$ average best-effort traffic rate
$B \leftarrow$ best-effort flows
while batch is running do
    $X \leftarrow$ new incoming flow
    if $X$ is emergency then
        $\tau \leftarrow$ requested bandwidth by $X$
    else
        $\tau \leftarrow R$
    end if
    if $\tau$ is not available then
        apply_greedy_heuristic()
    end if
    apply_flows()
end while
run_batch()
```

IV. Implementation, Simulation and Evaluation

In this section, the solution described in Section III-A is implemented based on the two offline formulations summarized in Section III-B and III-C and evaluated in a simulation environment. Afterwards, the online approach described in Section III-D is implemented and evaluated on a practical environment consisting of Zodiac SDN switches [6].

A. Simulation Environment

The proposed offline models are first validated using simulations. The evaluated topology, as shown in Fig. 2, consists of 16 ingress/egress points of traffic and 32 switches. Switches 16-30 are backbone switches and the backbone network has the same topology as the Internet2 network [33]. This topology is used to simulate a provider network catering to about 2050 flows. Switches 31-38 are mobile base stations and the ingress/egress points attached to them represent mobile users. Switches 39-46 are DSL switches and the ingress/egress points attached to them represent DSL users. The bandwidth of each the backbone network link is 40 Gbps bidirectional.

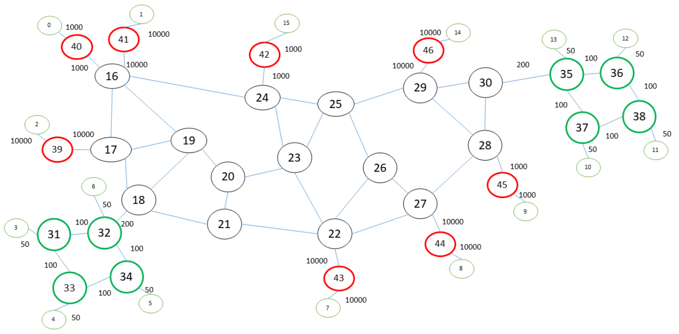

Fig. 2: The simulation topology based on the Internet2 network.

TABLE III: Specification of the network scenario

Source/Destination of All Flows	0 - 15
Backbone Network (40 Gbps)	16 - 30
Source/Destination Emergency Flows	{0, 2, 3, 5, 7, 8, 11, 13, 14}
DSL Network	39 - 46
Mobile Network	31 - 38

TABLE IV: Comparison of the LP and ILP model simulation results

	LP Model	ILP Model
Solving Time-Before(ms)	484	18408
Solving Time-After(ms)	484	15210

The specifications of the network scenario are summarized in Table III. IBM ILOG CPLEX [34] v12.7 is used to implement the models and the simulations are executed on a server with 2 Xeon E5-2690 v4 CPUs operating at 2.6GHz with 16GB of memory.

We defined 3 traffic classes with ranges $[0, \ 25000]$, $[0, \ 10000]$ and $[0, \ 5000]$ Kbps with priorities of 100, 50 and 10, respectively for best effort flows. Moreover, the requested rate of each emergency flow was randomly chosen from set $\{25000, 10000, 5000\}$ Kbps because of the variation in types of emergency network traffic. Each best effort flow was randomly assigned to a class. Each evaluation result is the average of 30 simulation runs.

B. Simulation Evaluation - Results

The performance of the two models is compared in Fig. 3. By increasing the number of best effort flows, the solving time increases in both models. However, the increase rate of the ILP model is exponentially higher than for the LP model. For 2000 best effort flows along with 50 emergency flows, the ILP model solves the problem in almost two minutes. It is worthwhile to mention that the solving time of the ILP model can further be decreased by up to one order of magnitude when using acceleration methods such as the novel algorithm based on the Benders decomposition method as described in [35].

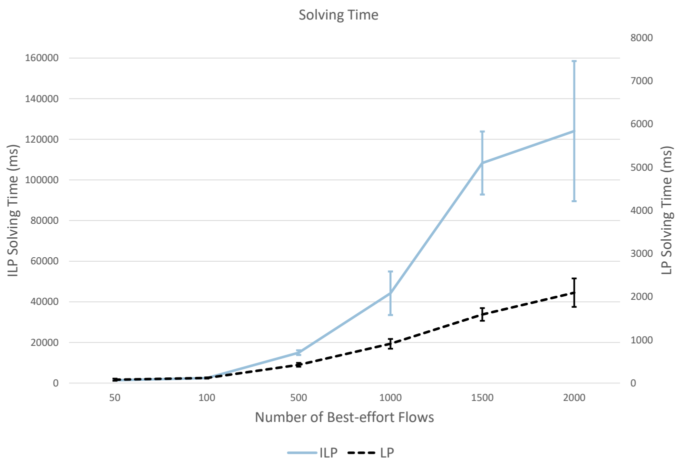

Fig. 3: The solving time of the ILP and LP models. Standard deviations are shown in the form of error bars

To investigate the operational details of the models, we first generated 500 best effort flows and solved both the ILP and LP models. After that, we added 50 emergency flows and solved the problems again. Both models reported the same optimal values before and after adding the emergency flows which means that the same result is achieved by both models and the LP model solved the problem 30 times faster than the ILP model. After adding the emergency flows, the models decreased the rate of best effort flows to allocate the requested bandwidth of the emergency flows which resulted in a lower optimal values. A summary of the results is shown in Table IV.

C. Prototype Implementation

To implement the proposed solution, a network consisting of Zodiac SDN switches and a Ryu [7] SDN controller with 4 Intel Core i5-7440HQ running at 2.80GHz and 16 GB of memory is used. As discussed in Section III-D, the Zodiac SDN switches do currently not support flow splitting and thus the use of the slower ILP model is obliged in this environment. A database using the schema presented in Fig. 4 is used to store the current active flows and the information about the topology. The topology used for the evaluation is visualized in Fig. 5 and built with 1 Zodiac GX switch (sw1), 8 Zodiac FX switches (sw2 - sw9) and 10 raspberry pi's (d1 - d10). The Zodiac GX has an uplink of 1 gbps while the Zodiax FX switches have an uplink of 100 mbps. The used traffic classes are illustrated in Table V and the requested bandwidth rates based on destination are summarized in Table VI. OpenFlow v1.3 meters were used to specify the upper bound and lower bound rates of each traffic class and the ILP model is implemented with IBM ILOG CPLEX v12.7. Note that with the provided meters in this prototype, the ILP will rather assign the meter with 0 kbps bandwidth to flows with a lower priority in case there is a shortage. When more meters per traffic class are allocated, a downgrade is possible, but this is currently not implemented.

Assume $\{\ m_1, m_2, \ldots, m_n\}$ are n defined meter rates and $m_i \leqslant m_{i+1}\ \forall i$. The weight of best effort flow i is calculated

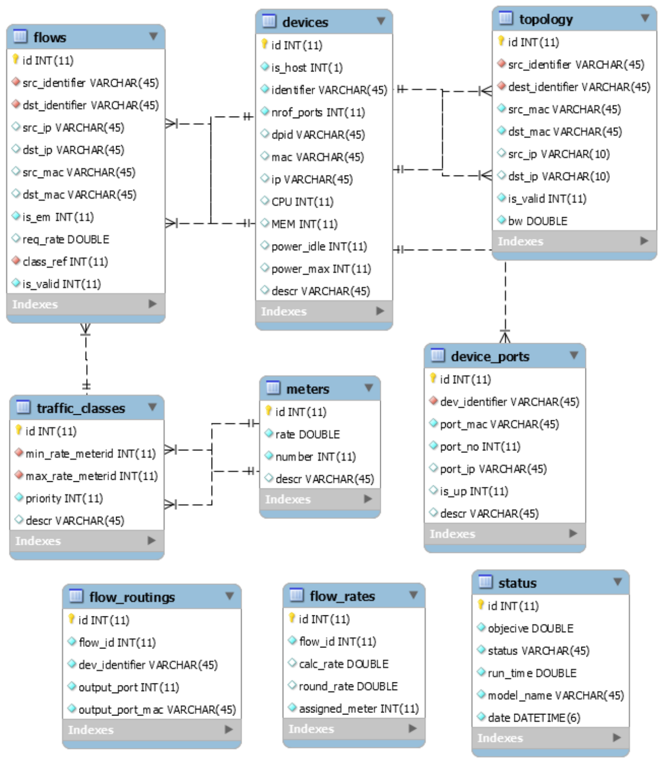

Fig. 4: Topology of the database used by the prototype implementation. The tables devices, device_ports, meters, topology and traffic_classes are filled based on the practical environment. The tables flows, flow_rates and flow_routings contain the optimized best-effort and emergency flows after solving the ILP formulation.

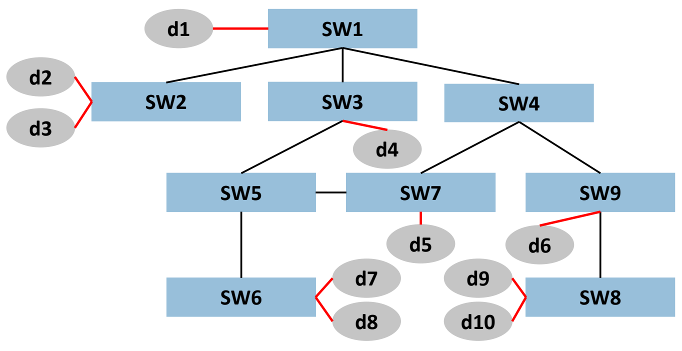

Fig. 5: Topology of the prototype environment. Sw1 is a Zodiac GX switch, sw2 - sw9 are Zodiac FX switches and d1 - d10 are raspberry pi's.

TABLE V: Traffic classes (all in kbps)

Id	Name	Minimum Rate	Maximum Rate
1	High Priority	0	25000
2	Normal Priority	0	10000
3	Low Priority	0	5000

by $\left\lceil P_i \times \frac{m_n}{m_1+1} \right\rceil$ in which P_i is the priority of the class that

TABLE VI: Requested rates per flow based on the destination. Rates between 0 and 4999 kbps are part of traffic class 3, rates between 5000 and 9999 kpbs are part of traffic class 2 and rates higher dan 10000kpbs are part of traffic class 1.

Destination	Traffic class
d1	3
d2	3
d3	3
d4	3
d5	3
d6	2
d7	2
d8	2
d9	2
d10	1

the flow belongs to and $\lceil x \rceil$ is the ceiling function.

When a new flow arrives, it is added to the database. When the previous batch of the offline calculations is finished, the controller runs the ILP solver. The implementation reads the database information, solves the ILP problem and stores the results in the database. The output of the ILP is the assignment of each flow to one meter and the routing of flows over the network. To assign a flow to a meter, whether it be best effort flows or emergency flows, the implementation rounds down the calculated rate to the nearest defined meter rate. Based on the simulation results summarized in Section IV-B, the ILP model provides optimal results with a high number of flows but not in real-time. To combat this, we run the offline model consecutively while the online approach is used to route and to apply the corresponding meter to new incoming flows. Best-effort flows will be assigned to the average meter with 10000 kpbs while emergency traffic will be assigned to their requested rate. To decrease the impact on the current best-effort and emergency flows, a greedy heuristic is applied to reassign available bandwidth from other best-effort flows, based on their priority.

D. Prototype Evaluation - Example scenario

The prototype is evaluated to study the behavior of the online approach and the findings are illustrated in Fig. 6. When the batch calculation is running, the online approach will handle the new incoming flows. The flow responsible for the traffic going from d1 to d5, which is part of traffic class 3, is already allocated together with 87 other flows. Next at time t1, a new incoming emergency flow going from d1 to d10 is added to the network, with a requested bandwidth of 25,000 kbps. Because of its priority, the requested bandwidth is allocated and the greedy heuristic reduced the bandwidth from the other best-effort flows. The flows part of traffic class 3 have an average decrease of 284 kbps. Afterwards at time t2, a flow going from d8 to d10 is added, which is part of traffic class 1. As this is a best-effort flow, the average best-

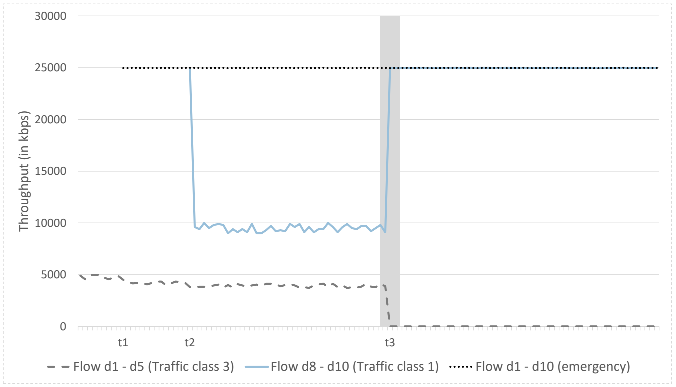

Fig. 6: Throughput of 2 best-effort flows and 1 one emergency flow. At time t1, the emergency flow is added and assigned by the online approach. At time t2, another best-effort flow is added and assigned by the online approach. At time t3, the offline batch has calculated and applied the optimal solution.

TABLE VII: The ILP model results

	Before	After
Solving Time (ms)	20520.234	17489.531

effort meter with a bandwidth of 10,000 kbps is allocated. The greedy heuristic again determines the bandwidth for each best-effort flow without impacting the current emergency flows. Finally, the batch calculations (visualized by the gray vertical line at time t3 in Fig. 6) optimizes the flows of the whole network again.

It is clear that the online approach is guaranteeing the bandwidth of the emergency flows and creates a sub-optimal solution for the new incoming best-effort flows. The sub-optimal solution has 2.76% difference per flow compared to the result of the offline batches in the whole example scenario. In some cases, this difference is 100% because the online approach does not drop any new incoming flows, while the offline batches can decide to drop a flow much faster as explained in Section IV-C. Afterwards, the batch calculations optimizes the best-effort flows over the remaining available bandwidth not used by emergency flows. The solving time of the batch calculations before and after adding 50 emergency flows is illustrated in Table VII and shows that the proposed offline model can solve small-sized networks efficiently.

V. Conclusion

Emergency network traffic needs to have priority over best-effort traffic during emergency situations. With the expected release of 5G, slicing concepts at network level will enable prioritization of the emergency network traffic over mobile connections. In addition, SDN principles allow to assign different QoS levels to different network slices.

In this paper, we therefore propose an SDN-based solution whereby both LP and ILP theoretical mathematical models guarantee the requested rate of emergency flows and maximize the best-effort flows over the remaining available bandwidth. Due to the calculation time of these models, an online approach handles new incoming flows in between these calculations. The shortest path, based on Dijkstra's algorithm, is calculated and a greedy heuristic is applied to obtain bandwidth from the other best-effort flows. When the new incoming flow is an emergency flow, the requested bandwidth will be allocated by any means, a new incoming best effort flow will be allocated with the average bandwidth of all the active best-effort flows.

The two offline models are first evaluated by simulations and the results show that both with the ILP and LP mathematical problems can be used whereby the ILP model exhibiting plus-second execution time while the LP model works 30 times faster for 500 best-effort flows and 50 emergency flows. Afterwards, the batch calculations together with the online solution are prototyped and evaluated on an SDN network consisting of Zodiac Switches and Raspberry pi's. The Zodiac switches do not support flow splitting, so the use of the slower ILP model is obliged. The practical evaluation shows that the online problem efficiently handles new incoming flows while guaranteeing the bandwidth for all the emergency flows and providing a sub-optimal temporary solution for the best-effort flows.

Practical evaluation of the faster LP model together with an extended evaluation of the proposed models is envisaged as future work.

REFERENCES

[1] C. McGarry, "The Truth About 5G: What's Coming (and What's Not) in 2019," 2019. [Online]. Available: https://www.tomsguide.com/us/5g-release-date,review-5063.html

[2] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. Leung, "Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[3] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

[4] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 114–119, 2013.

[5] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Network*, vol. 28, no. 6, pp. 18–26, 2014.

[6] P. Zanna, "Zodiac FX," 2019. [Online]. Available: https://northboundnetworks.com/collections/zodiac-fx/products/zodiac-fx

[7] F. Tomonori, "Introduction to ryu sdn framework," *Open Networking Summit*, 2013.

[8] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.

[9] J. Yan, H. Zhang, Q. Shuai, B. Liu, and X. Guo, "HiQoS: An SDN-based multipath QoS solution," *China Communications*, vol. 12, no. 5, pp. 123–133, 2015.

[10] Y. Zhang, Y. Tang, D. Tang, and W. Wang, "QOF: QoS framework based on OpenFlow," in *2015 2nd International Conference on Information Science and Control Engineering*, 2015, pp. 380–387.

[11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[12] A. V. Akella and K. Xiong, "Quality of service (QoS)-guaranteed network resource allocation via software defined networking (SDN)," in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, 2014, pp. 7–13.

[13] S. Cao, M. Tong, Z. Lv, and D. Jiang, "A study on application-towards bandwidth guarantee based on SDN," in *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[14] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, 2016, pp. 303–311.

[15] P. Pinto, R. Cardoso, P. Amaral, and L. Bernardo, "Lightweight admission control and traffic management with SDN," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[16] H. Krishna, N. L. M. v. Adrichem, and F. A. Kuipers, "Providing bandwidth guarantees with OpenFlow," in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, 2016, pp. 1–6.

[17] C. Morin, G. Texier, and C. Phan, "On demand QoS with a SDN traffic engineering management (STEM) module," in *2017 13th International Conference on Network and Service Management (CNSM)*, 2017, pp. 1–6.

[18] Y. Lu, B. Fu, X. Xi, Z. Zhang, and H. Wu, "An SDN-based flow control mechanism for guaranteeing QoS and maximizing throughput," *Wireless Pers Commun*, vol. 97, no. 1, pp. 417–442, 2017.

[19] A. Samani and M. Wang, "MaxStream: SDN-based flow maximization for video streaming with QoS enhancement," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, 2018, pp. 287–290.

[20] Northbound networks. [Online]. Available: https://northboundnetworks.com/

[21] O. S. Specifications, "1.5. 1," *Open Networking Foundation*, vol. 3, 2015.

[22] "Understanding CoS Two-Color Marking - TechLibrary - Juniper Networks," Jun 2019, [Online; accessed 5. Aug. 2019]. [Online]. Available: https://www.juniper.net/documentation/en_US/junos/topics/concept/cos-ex-series-two-color-marking-understanding.html

[23] S. Haddock, "Frame metering in 802.1 q," 2013.

[24] GENI. [Online]. Available: https://www.geni.net/

[25] Mininet overview - mininet. [Online]. Available: http://mininet.org/overview/

[26] Open vSwitch. [Online]. Available: https://www.openvswitch.org/

[27] P. V. Mieghem and F. A. Kuipers, "Concepts of exact QoS routing algorithms," *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, pp. 851–864, 2004.

[28] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Flexible Traffic Splitting in OpenFlow Networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 407–420, 2016.

[29] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[30] L. G. Khachiyan, "Polynomial algorithms in linear programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 1, pp. 53–72, 1980.

[31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[32] E. W. Dijkstra, "A Note on Two Probles in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[33] "Internet2 Network Infrastructure Topology," 2018. [Online]. Available: https://www.internet2.edu/media/medialibrary/2019/04/10/I2-Network-Infrastructure-Topology-All-legendtitle.pdf

[34] "IBM ILOG CPLEX Optimization Studio." [Online]. Available: https://www.ibm.com/be-en/marketplace/ibm-ilog-cplex

[35] B. Farkiani, B. Bakhshi, and S. Ali MirHassani, "Stochastic virtual network embedding via accelerated benders decomposition," vol. 94, pp. 199–213, 2019.