Journal of
*Sensor and
Actuator Networks*

Article

SDN-Based Routing for Backhauling in Ultra-Dense Networks

Dania Marabissi *, Romano Fantacci and Linda Simoncini

Department Information Engineering, University of Florence, 50139 Firenze FI, Italy; romano.fantacci@unifi.it (R.F.); linda.simoncini@stud.unifi.it (L.S.)

* Correspondence: dania.marabissi@unifi.it

Received: 7 February 2019; Accepted: 18 April 2019; Published: 23 April 2019

Abstract: Ultra-Dense Network (UDN) deployment is considered a key element to achieve the requested capacity in future fifth-generation (5G) mobile networks. Backhaul networks in UDNs are formed by heterogeneous links with multi-hop connections and must handle massive traffic. Backhauling in future 5G networks may represent the capacity bottleneck. Therefore, there is the need for efficient and flexible routing schemes able to handle the dynamism of the traffic load in capacity-limited networks. Toward this goal, the emerging Software-Defined Network (SDN) paradigm provides an efficient solution, transferring the routing operation from the data plane switches to a central controller, thus achieving more flexibility, efficiency, and faster convergence time in comparison to conventional networks. This paper proposes and investigates an SDN-approach for an efficient routing in a capacity-limited backhaul network that carries data and control traffic of a heterogeneous UDN. The routing algorithm is centralized in the SDN controller and two different types of traffic flow are considered: data and control plane coordination traffic. The goal is to reduce or even to avoid the amount of traffic that the backhaul network is not able to support, distributing in a fair way the eventual lack of bandwidth among different access points. Simulation results show that with the considered approach the performance significantly improves, especially when there is an excess of traffic load in the network. Moreover, thanks to the SDN-based design, the network can reconfigure the traffic routing depending on the changing conditions.

Keywords: 5G; software-defined network; backhaul routing; ultra-dense networks

1. Introduction

Fifth-Generation (5G) wireless networks are expected to support a high variety of services and applications, with different quality-of-service requirements and network functions, in a cost-effective manner [1]. Disruptive changes in mobile networks are needed for supporting such an evolution. In particular, access-point densification (i.e., Ultra-Dense Networks, UDN) is one of the key elements of 5G to achieve the desired increase of the network capacity. The transmitter and the receiver are moved closer together, creating a multi-layer heterogeneous architecture where many low-power small cells, such as microcells, picocells, and femtocells are overlaid on high-power macrocells. Small cells can be deployed anywhere by operators and end users [2]. However, small-cell deployment on large scale is challenging. A certain level of centralization is required to manage inter-cell interference, mobility, load balancing and coordinated transmission schemes [3].

The traffic generated by cells (both data and coordination traffic) must be routed through the backhaul network toward the core network (CN) and other cells, often with stringent requirements in terms of capacity, latency, availability, energy, and cost efficiency. The consequent massive traffic forwarding is a great challenge for future 5G backhaul networks that may represent the capacity bottleneck. Moreover, the 5G backhaul network is complex and must have the ability to dynamically adapt to changes of the radio access network (RAN) in a flexible, efficient, and timely manner. 5G backhaul network is characterized by heterogeneous links: small cells are used in different environments, such as homes, small offices, hotspots, and enterprises, where the available connections to the CN are different (i.e., a mix of wired and wireless links with different characteristics). Finally, in presence of many access points, it is expected that the backhaul network is composed by multi-hop connections, where some small cells act as relays aggregating the incoming traffic and forwarding it. Therefore, there is the need for efficient and flexible routing schemes for capacity-limited backhaul networks, in order to enable a scalable and efficient deployment of UDNs. These schemes must be able to handle the variations of the traffic load at the access points, which fluctuates across space and time. Toward this goal the emerging Software-Defined Network (SDN) paradigm provides an efficient solution. SDN moves the routing decisions from switches to a central controller, thus achieving more flexibility, efficiency, and faster convergence time.

The basic idea of SDN is decoupling data (i.e., forwarding) and control planes (i.e., configuration and management) of the network, and facilitating the interaction between the two using an open interface, such as OpenFlow protocol, which specifies how to program the behavior of the network elements by a remote controller [4]. In that way the main intelligence of the network is centralized in a software-based controller which has an abstract view of the physical network topology. The controller also provides application programming interfaces (APIs) to operators that can directly program customized control algorithms depending on their objectives, thus reducing capital and operational expenses. Software-defined networking could effectively facilitate the management of heterogeneity, complexity, and dynamism of the mobile systems [5]. In particular, an SDN-based UDN will be able to dynamically respond to changes of traffic load and interference levels in the small cells, and will facilitate the co-existence of heterogeneous backhaul links. In addition, such an architecture provides the possibility of scaling the network adding/changing some links.

This paper proposes and investigates an SDN-approach for an efficient routing in a capacity-limited backhaul network, which carries data and control traffic of a heterogeneous UDN. The routing algorithm runs on the SDN controller, and two different types of traffic flow are considered: data and control plane coordination traffic. The goal is to avoid or reduce the amount of traffic generated by the small cells, which the backhaul network is not able to support, distributing in a fair way the eventual lack of bandwidth among different access points. Simulation results show that with the considered approach the performance significantly improves, and the network can reconfigure the traffic routing depending on the changing conditions.

2. Related Literature

SDN has been widely investigated for wired networks, where its benefits have been proved both in terms of performance and management. As a consequence, there is an increasing interest in evaluating if also mobile networks can take advantages from SDN [5,6]. Benefits and challenges of the software-defined-networking in wireless networks are analyzed in several papers that present high-level architectures [7–9]. In particular, there is a great interest in SDN-based backhauling due to the dominant role that these networks will have in future 5G systems, especially in presence of UDNs and new paradigms such as Cloud-RAN, caching, fog/edge computing [10–12]. In this area, recent papers focus on specific services provided by an SDN-based backhaul network, such as mobility management [13–15], multi-tenant network slicing [16,17], caching [18,19] and cross-layer coordination [20].

For what concerns routing schemes, the literature mainly refers to wired and fixed networks. In particular, the shortest-path routing protocols are the most widely used and investigated. Main research efforts have been focused on bandwidth-constrained solutions, where a new flow can be admitted only if there is a route that can support its bandwidth demand. Recently, also bandwidth-delay constrained problems have been considered. Using SDN paradigm, paths between source and destination can be defined with more flexibility, thus trying to overcome limits of the classical routing, and mainly aiming at minimizing the maximum link use of the network. An exhaustive survey of quality-of-service (QoS) routing strategies for SDNs is provided in [21], while a comparison between routing on conventional and software-defined wired networks is presented in [22] in terms of convergence time. Moreover, different SDN-based routing strategies have been recently investigated for wired networks. For example, in [23] a hybrid SDN is presented, combining the flexibility of SDN with the robustness of Open Shortest Path First (OSPF) algorithm. A multipath routing protocol that integrates also load balancing and admission control is proposed in [24], while the network use is maximized in [25] solving a bandwidth-delay constrained problem.

By contrast, few papers are available on SDN-based routing in mobile network backhauling. In this area, the attention is mainly devoted to wireless backhauling at *mmWave* frequencies, which is a promising solution for small cells. The focus is on multi-hop transmissions, whose goal is to increase system throughput jointly optimizing resource allocation and capacity-aware routing [26–30]. Routing in a wireless SDN (i.e., not a backhaul network) is investigated in [31], where a modified shortest-widest path algorithm is proposed. Conversely, actual backhaul networks for UDNs made of heterogeneous links and multi-hop connections are considered in [32,33]. A distributed anycast routing protocol is proposed in [32] for a conventional network. Differently, [33] considers an SDN-approach that mainly focuses on jointly solving the problems of routing in the backhaul network and user equipment (UE) cell association, following a cross-layer approach.

Paper Contribution

From above we can summarize that routing policies, specifically designed for UDNs backhauling, have not been yet widely investigated. The literature mainly refers to routing in wired and fixed networks, where the common goal is to minimize the maximum link use, thus avoiding that some links are particularly congested while others are underused. However, sufficient capacity to admit a new flow is always assumed.

Differently, this paper focuses on UDNs taking into account that: (i) UDNs backhaul network may not be able to support all the traffic demand generated by cells, (ii) UDNs backhaul network is characterized by an irregular topology, changing in time and space, made by heterogeneous links and multi-hop connections.

Hence, the paper contribution can be summarized as

- a new optimization problem is defined for a capacity-limited backhauling in UDNs. In particular, the problem considers two types of traffic (data and control) according to the SDN paradigm, and aims at guaranteeing control-flow capacity demands and minimizing the amount of *unsatisfied data-flow demands*. This is the amount of data-flow capacity requested by cells that cannot be supported by the backhaul network. Moreover, the aim is fairly distributing the eventual lack of bandwidth (i.e., *unsatisfied requests*) among cells.
- a new algorithm is proposed to solve the problem. It differentiates the network behavior for control and data traffic, to satisfy their different constraints. Moreover, it takes into account not only the capacity of each link, but also its congestion. In particular, we propose a two-step algorithm based on a modified Dijkstra algorithm. It is designed for an SDN implementation that requires a global knowledge of the network and is performed centrally at the SDN controller.

The paper is organized as follows. First we introduce the main concepts of SDN in Section 3, then the considered scenario is described in Section 4. Section 5 presents the problem formulation whose solution is proposed in Section 6. Finally, numerical results are presented in Section 7 and conclusions are drawn in Section 8.

3. Software-Defined Backhaul Network

An SDN-based backhaul network is composed by a set of forwarding elements as a traditional network. However, while in traditional networks the control is distributed in every node (i.e., routing protocols are executed in every node), in SDNs devices are simply forward nodes without any control function.

The main pillar of SDN is decoupling control and data planes: the first one is represented by the SDN controller that aggregates and performs control decisions, while the second is represented by forwarding nodes and links connecting these nodes, and simply becomes packet forwarding. In particular, the main elements of an SDN network are:

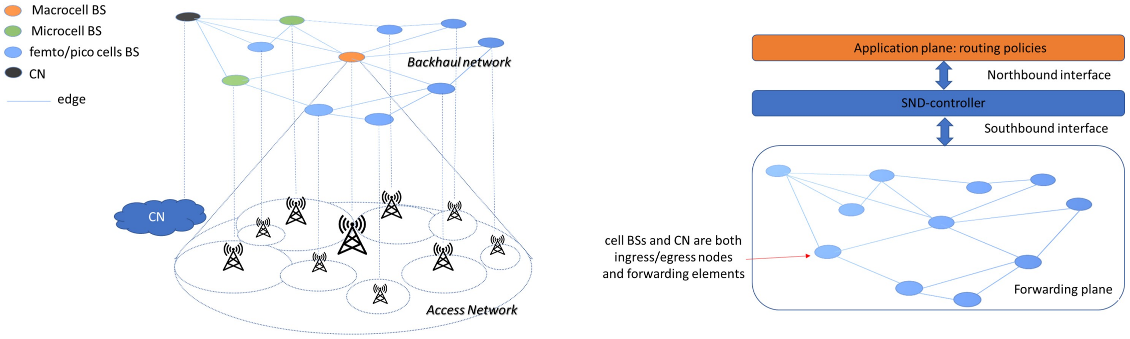
- *forwarding devices* (FD)—physical devices that accomplish instructions that are provided by the SDN controller throughout the southbound interface, which is installed by the SDN controller on devices depending on the used protocol;
- *southbound interface*—the interface that defines the communication protocol and the instructions set that are given by the SDN controller to FDs;
- *SDN controller*—the central controller that represents the network intelligence that instructs the FDs;
- *northbound interface*—the interface of the SDN controller toward the applications. Usually, a northbound interface abstracts the low-level instructions set used by southbound interface to program FDs;
- *application plane*—the set of applications that define the policies that are then translated in instructions to program the forwarding elements. Examples of applications are routing, monitoring, and load balancing.

In an SDN the control logic is moved on an external entity and is programmable via software. The SDN controller supervises the behavior of the forwarding elements through an open programming interface such as OpenFlow. The controller can translate the application requirements down to the data plane, and to provide an abstracted view of the physical network to applications. More in detail, the SDN controller defines forwarding rules that are pushed down to the FDs, which simply match the rules with the incoming data flows. Simplifying the functionality of FDs, these can be implemented on general-purpose hardware and controlled with standard protocols. This leads to a significant flexibility increase respect to traditional networks where each router must perform control functions and implements proprietary interfaces that cannot be reconfigured via software. Moreover, if significant changes frequently occur (like in case of UDNs), traditional networks present a significant signaling increase and longer convergence time of routing algorithms. Differently, centralized control operations require less communications (i.e., signaling) and shorter delays. SDN controller must have an updated global view of the network connectivity and throughput. When a change occurs in the network, the SDN knowledge is updated.

4. System Model

We consider here a heterogeneous network where micro-, pico-, and femto-cells (generally referred as small cells or access points in what follows) are densely and randomly deployed in the macrocell area. The total number of cells is C (i.e., small cells plus macrocell). Each cell base station (BS) provides access to UEs connected with it, hence, the cell BS has a certain amount of traffic generated by the associated-UEs that must be routed in the backhaul network. Due to differences in cell coverage areas the amount of traffic generated by different cells can significantly vary. Following the SDN principle, we consider two types of traffic flow: *data* and *control*. Indeed, cell coordination can be split in control plane coordination, which allows optimization of network parameters (e.g., interference management, load balancing, mobility management), and data plane coordination, which allows data exchange and resource scheduling. The two flows are routed following different principles, and control traffic has higher priority.

Each cell BS has a backhaul connection with the CN and/or with one/more other cell BSs by means of heterogeneous links with different communication capacity. In particular, macro- and micro-cell placement usually follows operator's network planning; hence, macro- and micro-cells have high-capacity links toward the CN and among them. Conversely, pico and femto-cells can be placed anywhere by operators and end users following a random deployment. Some small-cell BSs can have direct backhaul links toward the CN while others use multi-hop connections toward the CN, exploiting other cells as forwarding elements. The UDN scenario is represented in Figure 1a. The UDN backhaul network can be seen from an SDN perspective as a network where the CN and the cell BSs are both ingress/egress nodes and forwarding elements. Indeed, BSs generate and receive traffic, and at the same time forward traffic of other neighbor BSs. The UDN backhaul from an SDN perspective is represented in Figure 1b. Decisions on paths for data forwarding are taken by the SDN controller that is placed in the CN. The SDN controller implements routing policies that are indicated by the application plane, which in this case requires different treatments for control and data traffic.



(**a**) UDN scenario: access and backhaul networks.

(**b**) SDN-based backhaul network.

Figure 1. Considered scenario.

From the figure we can see that the backhaul network can be modeled with an undirected graph G where vertexes are cell BSs plus the CN and edges are the backhaul links. In particular, the set of vertexes $\mathcal{V}$ has cardinality $V = C + 1$, and the set of edges $\mathcal{E}$ has cardinality E. We define a capacity matrix $\mathbf{C}$ whose element $c_{e_{v,w}}$ is the capacity of the edge connecting vertexes v and w, $e_{v,w}$. If the capacity is zero it means that the edge does not exist. Vertex v has a demand of data and control traffic to route in the backhaul network. We define two traffic matrices $\mathbf{D}^D$ and $\mathbf{D}^C$ that represent the data- and control-traffic demands from the sources to the destinations. In particular, the element $d^X_{v,w}$, with $X = D$ or C (D for data traffic and C for control traffic), represents the amount of data/control traffic from ingress node v to egress node w. Control traffic generated by cells is directed toward the CN, hence assuming that the CN is the first vertex, we have that $d^C_{v,w} = 0 \; \forall v, w \neq 1$, while the control traffic generated by the CN can be directed to any cell.

5. Problem Formulation

As stated before, dense and random deployment of the network infrastructure imposes new challenges in the backhauling, which becomes the capacity bottleneck and may not be able to support all the traffic demanded by cells. As consequence, we formulate a new routing problem that is particularly suited for this scenario.

We suppose that the admission of an access point to the network is constrained by the control traffic, i.e., a small cell can be admitted only if there is at least one path on the backhaul network that supports its control-traffic bandwidth demand. By contrast, the data traffic demand can be only partially satisfied (i.e., it is not constrained). The main concept is that control traffic is mandatory for an access point to function, hence, it cannot be limited. Conversely, if the backhaul network is not able to support all the data traffic demand, the SDN controller instructs the small-cell BS to take actions at RAN level to reduce its data traffic such as call blocking, UE's throughput reduction and load balancing policies [34]. In general, in a UDN, UEs are under the radio coverage of multiple cells, hence, they can be forced to connect to different cells, thus operating load balancing directly at the ingress nodes. As a consequence, we want to jointly solve:

- a bandwidth-constrained problem for the control traffic;
- the minimization of the *total amount of unsatisfied traffic demand (UTD)* of the cells, with a fair distribution of unsatisfied demands.

The routing algorithm allows the finding of the path from a source node to a destination for both control and data traffic. In particular, being $\mathcal{L}_{v,w}$ the set of all the possible paths from ingress node v to egress node w, whose generic element is $L_{v,w}$, the routing algorithm allows selection of one of these paths for the control $\bar{L}^{C}_{v,w}$ and for the data traffic $\bar{L}^{D}_{v,w}$ (The two flows have different constraints, hence, can have different routes even if they have the same ingress and egress nodes.). Each path $L^{C(D)}_{v,w}$ is composed by a set of edges $L^{C(D)}_{v,w} = \{e_{v,i}, e_{i,j}, ..., e_{k,w}\}$. As a consequence, we can define two three-dimensional matrices $\mathbf{P}^C$ and $\mathbf{P}^D$ with dimension $VXVXE$ whose element $p^{C(D)}_{v,w,e_{i,j}} = 1$ if the path from ingress node v to egress node w passes through link $e_{i,j}$, zero otherwise.

The total demand of traffic through edge $e_{i,j} \in \mathcal{E}$ is

$$Q_{e_{i,j}} = \sum_{(v,w)\in\mathcal{V}^2} \left[d^C_{v,w} p^C_{v,w,e_{i,j}} + d^D_{v,w} p^D_{v,w,e_{i,j}} \right] \tag{1}$$

where $\mathcal{V}^2$ is the set of all the possible vertex-pairs, (v,w), whose dimension is V X V. If the edge capacity, $c_{e_{i,j}}$, is lower than $Q_{e_{i,j}}$, the link $e_{i,j}$ is not able to support all the traffic demand.

Let us define two matrices of allocation $\mathbf{A}^C$ and $\mathbf{A}^D$, whose element $a^{C(D)}_{v,w}$ represents the amount of capacity that has been allocated to the flow from node v to node w for the control and data traffic, respectively.

We can formulate the optimization problem as

$$\min_{\mathcal{L}_{v,w}\forall(v,w)\in\mathcal{V}^2} \left\{ \sum_{(v,w)\in\mathcal{V}^2} \frac{d^D_{v,w} - a^D_{v,w}}{d^D_{v,w}} \right\} \tag{2}$$

$$\text{s.t.:} \quad d^C_{v,w} = a^C_{v,w} \quad \forall(v,w)\in\mathcal{V}^2 \tag{3}$$

$$d^D_{v,w} \leq a^D_{v,w} \quad \forall(v,w)\in\mathcal{V}^2 \tag{4}$$

where the first constraint imposes that the control-traffic capacity demand is always satisfied, while the second imposes that the amount of capacity assigned to a data flow does not exceed the demand. Moreover, *fairness* is introduced by normalizing the UTD for the data-traffic demand of each flow, $d^D_{v,w}$. The aim is to have almost the same percentage of UTDs for different flows.

The optimal solution of the proposed problem is not feasible with an affordable complexity, hence, we divide the problem in two sub-problems: (i) finding the solution for the bandwidth-constrained problem of the control traffic, and (ii) finding the paths for the data flows that minimize the UTDs.

Benchmark Approaches

Before to explain the proposed approach, we shortly describe the methods considered to be benchmarks. Being the shortest-path routing protocols the most widely used, we have considered two of these algorithms. In particular, we consider the Dijkstra algorithm [35] to find the path with the minimum cost in two different cases:

- the cost of each edge is equal to 1, i.e., the routing algorithm looks for the path with the *minimum number of hops* (MNH);
- the cost of each edge is the inverse of the edge capacity, i.e., the routing algorithm takes into account both the number of hops and the capacity of each link thus preferring links with higher capacity (HC).

For each vertex of the graph, the Dijkstra algorithm [35] generates a shortest-path tree (A tree is a graph with one and only one path between every two nodes) with the given vertex as root, i.e., the node that originates the path. Two sets are maintained, one contains vertices included in the shortest-path tree and the other contains those not included. At every step the algorithm finds a vertex not yet included in the shortest-path tree that has minimum cost from the source. At the beginning only the root (i.e., the selected vertex of the graph) is included in the shortest-path tree.

We suppose that first the algorithm (MNH or HC) is carried out for the control traffic with a bandwidth constraint: the selected path is the one that has minimum cost and has sufficient bandwidth to accommodate the control flow. Then the algorithm is performed again for the data traffic without any constraint.

6. Proposed Approach

To solve the considered problem, the routing algorithm must take into account both the capacity of each edge and the edge's congestion. Toward this goal we have modified the Dijkstra algorithm for an SDN implementation that requires a global knowledge of the network and is performed centrally at the SDN controller. Being the goal the minimization of UTDs with a fair distribution we refer to the algorithm as *fair unsatisfied demands minimization* (FUDM).

6.1. Control-Flow Routing

For the control traffic the bandwidth-constrained Dijkstra algorithm is applied using as edges' cost the inverse of the *residual capacity*, i.e., the edge's capacity minus the traffic demand already allocated on the edge. We define a *residual capacity* matrix $\mathbf{R}$, whose generic element is $r_{e_{i,j}}$, and whose dimension is VxV. This matrix is initialized at the edges' capacity value, i.e., $\mathbf{R}(0) = \mathbf{C}$.

The SDN controller performs the path search successively for all control flows that must be transmitted in the network. In particular, for each flow, the algorithm finds the path that has the minimum cost and can support the flow's bandwidth demand. Then the residual capacity is updated by removing from each edge that belongs to the selected path, the capacity demand of the allocated control flow. More in detail, at the s-th iteration the s-th control flow is selected. The selected flow is characterized by the source-cell v, the destination-cell w, and the traffic demand $d^C_{v,w}$. The Dijkstra algorithm is used to determine the path from cell v to cell w, with the minimum cost and sufficient capacity, i.e., $\bar{L}^C_{v,w} = \{e_{v,i}, e_{i,j}, ..., e_{k,w}\}$. Hence, the residual capacity of edges belonging to the path $\bar{L}^C_{v,w}$ is updated as: $r_{e_{i,j}}(s) = r_{e_{i,j}}(s-1) - d^C_{v,w}$ with $e_{i,j} \in \bar{L}^C_{v,w}$. Moreover, the capacity allocated to the control flow is $a^C_{v,w} = d^C_{v,w}$.

6.2. Data-Flow Routing

Data traffic path selection is always based on a modified Dijkstra algorithm. However, the algorithm is not constrained and introduces a proportional fairness concept. At the beginning of this phase, the residual capacity matrix, $\mathbf{R}$, has the value achieved at the end of control-flow allocation.

The SDN controller lists data flows in descending order, and then performs the allocation starting from the data flow with the highest capacity demand. At the s-th iteration the Dijkstra algorithm is used to determine the path for the s-th data flow that has v as ingress node, w as egress node and the traffic demand $d^D_{v,w}$.

As before, first the algorithm looks for the path that has minimum cost and sufficient capacity to accommodate the data-flow demand. If this path, $\bar{L}^D_{v,w} = \{e_{v,i}, e_{i,j}, ..., e_{k,w}\}$, exists, the residual capacity of all edges belonging to $\bar{L}^D_{v,w}$ is updated as $r_{e_{i,j}}(s) = r_{e_{i,j}}(s-1) - d^D_{v,w}$.

However, if no path in $\mathcal{L}_{v,w}$ supports the traffic demand of the s-th data flow, the route selection is based on the minimum amount of UTD. Toward this goal, we define the set of data flows allocated on the edge $e_{i,j}$ at the $(s-1)$-th step as $\mathcal{E}_{i,j}(s-1)$. That is, the data flow from ingress node v to egress node w, $(v,w) \in \mathcal{E}_{i,j}(s-1)$ if $e_{i,j} \in \bar{L}^D_{v,w}$. Hence, at the s-th iteration for each possible path from node v to node w belonging to the set $\mathcal{L}_{v,w}$, the UTD is calculated, and the route that guarantees the minimum

UTD is selected. The UTD for the considered flow ($UTD^D_{v,w}$) in the path $L_{v,w}$ is derived taking into account a proportionally fair distribution of the UTD among all the data flows allocated on an edge, i.e.,

$$UTD^D_{v,w} = \max_{e_{i,j}\in L_{v,w}} \left\{ \frac{d^D_{v,w}}{\sum_{(q,r)\in\mathcal{E}_{i,j}(s-1)} d^D_{q,r}} \left[\sum_{(q,r)\in\mathcal{E}_{i,j}(s-1)} d^D_{q,r} - r_{e_{i,j}}(0) \right] \right\} \tag{5}$$

More in detail, $\left[\sum_{(q,r)\in\mathcal{E}_{i,j}(s-1)} d^D_{q,r} - r_{e_{i,j}}(0)\right]$ is the amount of requested data traffic that cannot be satisfied by the edge $e_{i,j}$, which is the sum of all the allocated demands minus the capacity of the edge. The total amount of data traffic that cannot be satisfied by the edge $e_{i,j}$, is fairly distributed among all the paths allocated on that edge. In particular, the weighted UTD of the flow (v,w) on edge $e_{i,j}$ is proportional to its original demand, $d^D_{v,w}$. Finally, the $UTD^D_{v,w}$ is given by the edge $e_{i,j} \in L_{v,w}$ that introduces the maximum amount of weighted UTD. Indeed, the traffic that can pass through the path $L_{v,w}$ depends on the edge that has the minimum capacity (i.e., the bottleneck).

As a consequence, the selected path is

$$\bar{L}^D_{v,w} = \arg\min_{\mathcal{L}_{v,w}} UTD^D_{v,w} \tag{6}$$

and the capacity allocated to the data flow is $a^D_{v,w} = d^D_{v,w} - UTD^D_{v,w}$.

The procedure is repeated for all the data flows. Figure 2 reports the flow diagram of the proposed algorithm.

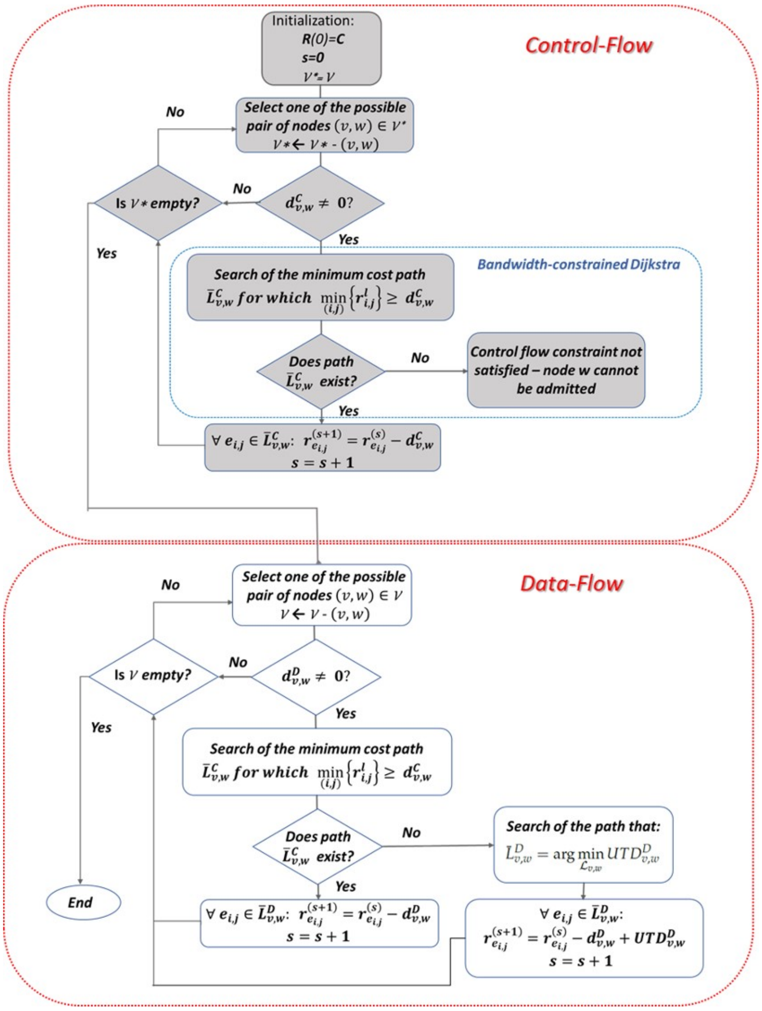

Control-Flow
Initialization:
R(0)=C
s=0
V*= V
Select one of the possible pair of nodes (v,w) ∈ V*
V*← V* - (v,w)
No
Is V* empty?
No
$d^C_{v,w} \neq 0$?
Yes
Yes
Bandwidth-constrained Dijkstra
Search of the minimum cost path $\bar{L}^C_{v,w}$ for which $\min_{(i,j)}\{r^l_{i,j}\} \geq d^C_{v,w}$
Does path $\bar{L}^C_{v,w}$ exist?
No
Control flow constraint not satisfied – node w cannot be admitted
Yes
$\forall\, e_{i,j} \in \bar{L}^C_{v,w}$: $r^{(s+1)}_{e_{i,j}} = r^{(s)}_{e_{i,j}} - d^C_{v,w}$
$s = s+1$
Data-Flow
Select one of the possible pair of nodes (v,w) ∈ V
V← V - (v,w)
No
Is V empty?
No
$d^D_{v,w} \neq 0$?
Yes
Yes
Search of the minimum cost path $\bar{L}^C_{v,w}$ for which $\min_{(i,j)}\{r^l_{i,j}\} \geq d^C_{v,w}$
Does path $\bar{L}^C_{v,w}$ exist?
No
Search of the path that: $\bar{L}^D_{v,w} = \arg\min_{\mathcal{L}_{v,w}} UTD^D_{v,w}$
Yes
End
$\forall\, e_{i,j} \in \bar{L}^D_{v,w}$: $r^{(s+1)}_{e_{i,j}} = r^{(s)}_{e_{i,j}} - d^D_{v,w}$
$s = s+1$
$\forall\, e_{i,j} \in \bar{L}^D_{v,w}$:
$r^{(s+1)}_{e_{i,j}} = r^{(s)}_{e_{i,j}} - d^D_{v,w} + UTD^D_{v,w}$
$s = s+1$

Figure 2. Proposed routing algorithm flow diagram.

6.3. SDN-Based Routing—OpenFlow Approach

As already stated the proposed routing algorithm runs at the SDN controller that supervises the FDs behavior by means of OpenFlow protocol. When a change occurs in the network, the SDN controller recomputes traffic-routes following the steps described before, and sends a MODIFY-STATE command to FDs affected by changes. FDs change corresponding entries in the forwarding tables. In particular, three types of messages are supported: *controller-to-switch*, *asynchronous* and *symmetric*.

Controller-to-switch messages are initiated by the SDN controller and used to manage and observe the FDs' state. Hence, these messages are used for feature/status requests and configurations. For example, the SDN controller periodically performs traffic monitoring (READ-STATE message) to collect links' statistics, in order to react in real time. SDN controller sends a flow-status request to FDs, which answer with a flow-status reply that contains *flow-duration* and *byte-count*, thus, to evaluate the link throughput.

Asynchronous messages are initiated by FDs to update the SDN controller of events and changes (e.g., a new flow, status change, etc.).

Symmetric messages are initiated by either the FD or the controller and sent without solicitation, such as HELLO or ERROR messages. For example, if a link goes down the FD sends an ERROR message to the controller that selects new paths that do not involve the failed link.

Figure 3 reports the message-flow for updating the network knowledge and forwarding tables in two cases: (a) SDN controller initiated (status update), (b) FD-initiated (link failure).

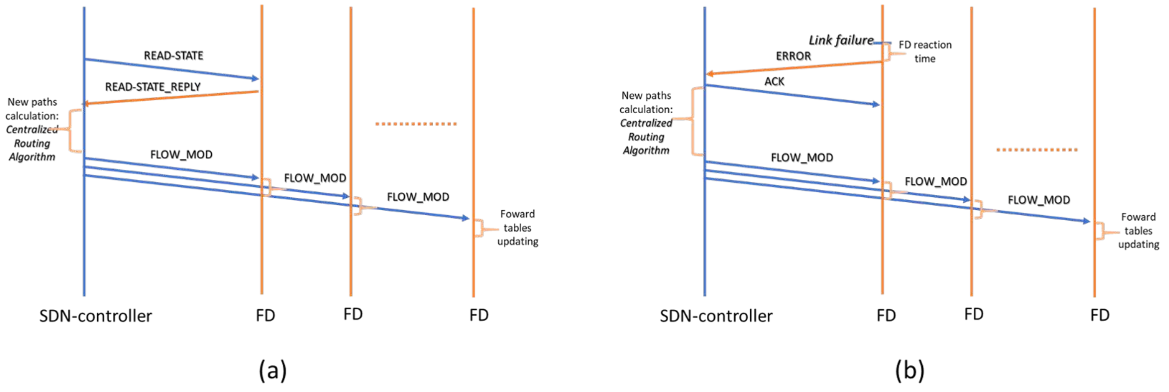


Figure 3. Messages-flow for network updates: (**a**) SDN controller initiated (status update), (**b**) FD-initiated (link failure).

Consequently, an SDN can timely react to the network changes. Indeed, in an SDN-based network the convergence time of routing algorithms is low and quite invariant respect to the network size [22]. Convergence time depends only on the reactivity of FDs to notify a change in the network to the SDN controller. Only the SDN controller must be advised, then it computes new routes and simultaneously instructs all the involved FDs. Conversely, in a traditional network if a change occurs, the FD first updates its routing tables, then sends a message to its neighbors. These on their turn updated their tables and send the message to their neighbors and so on. The message propagates in the network. Then each node updates its routing tables and sends its best paths to the other routers. This messages flooding introduces a significant delay because all FDs must be advised, then updated routing information must be propagated. Obviously, the convergence time and the delay increase with the network dimension.

Moreover, SDN allows use of a centralized algorithm as the one we propose. Indeed, in this case a distributed algorithm would be impracticable, because all routes are calculated/updated together so that to fairly distribute the UTDs. This operation can be done only by a central entity that has a global view of the network. As an example, Figure 4 reports the path-reconfiguration that follows the increase of the data-traffic demand of the node D. In particular, three nodes, (D, E, F), are involved in the network changes. The routes are depicted only for these nodes, and we suppose, in this example, that only data traffic is present directed toward the CN.

Initially, the node D requires 30 Mbps that can be entirely routed toward the CN by means the direct link. The network can satisfy all the traffic demands (i.e., no UTDs). When node D increases its traffic demand up to 50 Mbps, the shortest path (i.e., the direct path toward the CN) would lead an UTD for the node D of 20 Mbps. The proposed algorithm instead looks for the solution that minimizes the total UTD. If the traffic of node D toward the CN is forwarded by node C, the total UTD results to be 10 Mbps (i.e., the edge form node C to node A has 100 Mbps capacity but the total request of nodes C and D is 110 Mbps). Moreover, the proposed algorithm distributes the UTD among all the nodes that route their traffic on the congested link, in this example the 10 Mbps are distributed among the nodes C and D as represented in figure. In particular, the UTD of the two nodes are derived by the (5), which allows distribution of the unsatisfied requests among all the data flows allocated on the congested edge proportionally to their data-traffic demand as previously detailed.

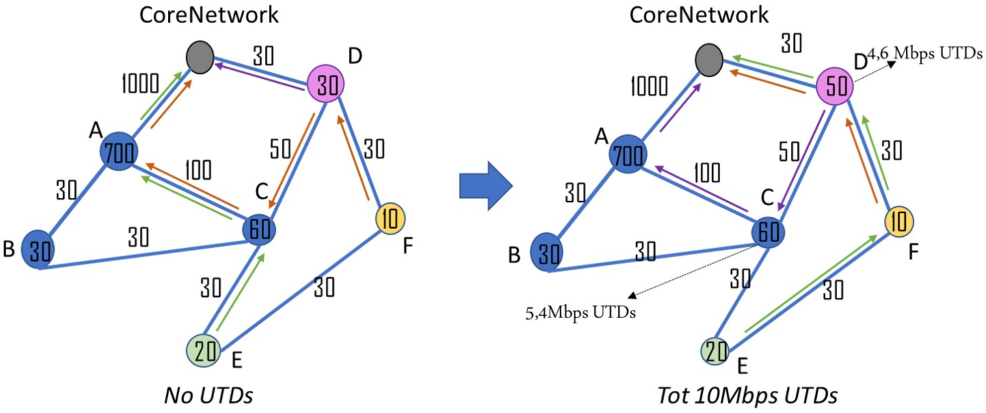


Figure 4. Reconfiguration example using the proposed SDN-routing scheme.

7. Results

This section provides numerical results to validate the effectiveness of the proposed SDN-based routing approach. As stated in Section 5, the aim of our paper is to minimize the total amount of UTDs, which is the requested capacity that the backhaul network is unable to support due to links overloading. As a consequence, we are interested in evaluating the total amount of UTDs achieved with the proposed method.

We define different macro-scenarios where network parameters are random variables with a certain probability distribution as reported in Table 1. We have averaged numerical results over several realizations of a given scenario. In that way results depend on the macro-scenario but not on a specific distribution of cells and traffic-loads.

Macro-scenarios differ in the mean values of the probability distributions. In particular, macro-scenarios are characterized by edges capacity and the network degree of connectivity. Edges' capacity depends on the link type, and we distinguish among three cases: high-, low-, and medium-capacity network as reported in Table 1. Low- and medium-capacity networks have been set so that not always there is sufficient capacity to satisfy all data-traffic demands, while the high-capacity network is always able to support all control and data-traffic demands. Also for what concerns the degree of connectivity of the network (i.e., the mean number of edges per cell) we consider three cases: (i) 2 edges - low-connected, 3 edges medium-connected, 5 edges highly connected. In particular, the number of edges per cell is derived as an integer value uniformly distributed in the intervals defined in Table 1. Finally, cells are randomly distributed in the considered area, A, and the number of cells is selected according to a homogeneous spatial Poisson point process, with intensity λ_s. Numerical results are given as a function of the mean number of cells, which is $S = \lambda_s A$. Moreover, data-traffic demands are uniformly distributed on given intervals that are different for each kind of cell as reported in Table 1. The control traffic is a percentage of the data traffic, in particular, in simulations we have assumed 1%.

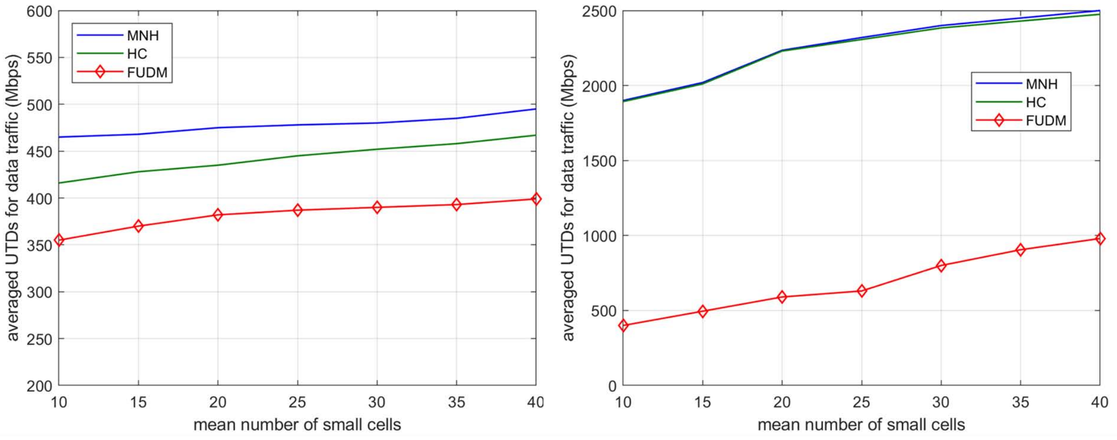
We want to underline that varying the network capacity while maintaining the traffic load constant leads to the same considerations as fixing the network capacity and varying the network load.

Since the control traffic is bandwidth-constrained, the control-traffic demands are always satisfied. Moreover, in high-capacity network scenarios, all traffic demands are satisfied (i.e., UTD is zero). For these reasons when we show results in terms of UTDs, we refer only to data traffic in low- and medium-capacity networks. The high-capacity network is considered for delay evaluations.

Table 1. Simulation parameters.

Considered area (A)	1 km^2		
Number of cells	random value distributed as a PPP with intensity λ_s Mean number $S = \lambda_s * A$		
Number of edges per node	Low-connected: unif. distributed in [1, 2] Medium-connected: unif. distributed in [1, 5] High-connected: unif. distributed in [2, 7]		
Data traffic demand (D_v^D)	macrocell unif.dist. in [1–2] Gbps	microcell unif.dist. in [100–1000] Mbps	picocell and femtocell unif.dist. in [10–50] Mbps
Edge capacity ($c_{e_{i,j}}$)	macrocell	microcell	picocell and femtocell
Low-capacity network	200 Gbps	10 Gbps	unif.dist. in [200–500] Mbps
Medium-capacity network	20 Gbps	2 Gbps	unif.dist. in [20–50] Mbps
Low-capacity network	2 Gbps	100 Mbps	unif.dist. in [10–20] Mbps

Figure 5 reports the total amount of UTDs for the data traffic as a function of the mean number of cells in the area when the network has medium (a) and low (b) capacity. Figures report the behavior of the proposed algorithm (FUDM) and the two benchmarks (MNH, HC). We can see that the proposed algorithm always outperforms the two benchmarks. In particular, the effectiveness of FUDM algorithm is more evident when the network has low capacity (or similarly, is overloaded), indeed in this case the gain in terms of UTDs increases. This means that the proposed method is more able to distribute the traffic among the available paths, thus reducing the links' congestion. In all cases we can observe that increasing the number of cells introduces only a slight increase of the UTDs value because also the number of possible paths in the network increases accordingly.



(a)

(b)

Figure 5. Averaged amount of total UTDs of data traffic vs. mean number of small cells for (**a**) a low-capacity network and (**b**) a medium-capacity network. Medium degree of connection of the network.

To see how the degree of connection of the network impacts on system performance, Figure 6 reports the total amount of UTDs when the mean number of edges per node varies (i.e., low, medium and highly connected network), for both low- and medium-capacity networks. As the network connection degree increases, also the number of edges in the network increases. Hence, the data traffic has more alternative paths to reach the destination and the links' congestion decreases. This is particularly true for the proposed method that can better distribute the traffic among the possible routes. Benefits are more evident in critical situations, i.e., when the network has low-capacity links.

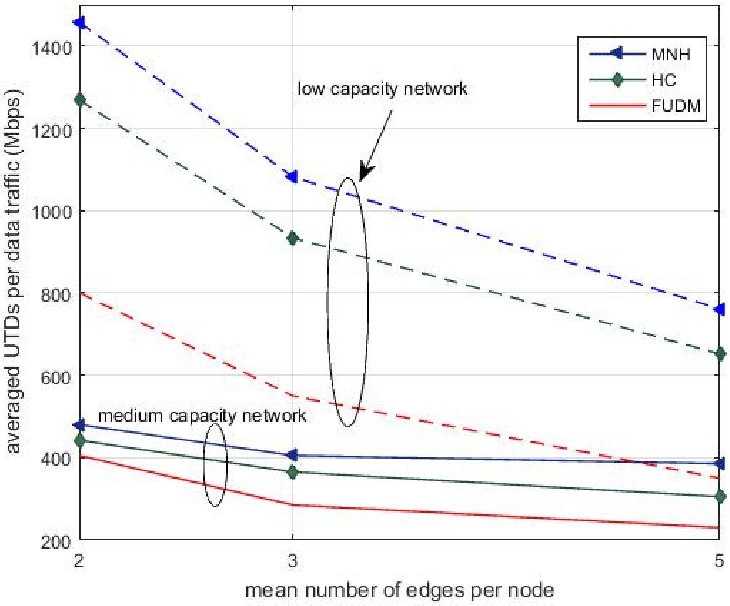


Figure 6. Averaged amount of total UTDs of data traffic vs. mean number of edges per node. The mean number of small cells is 30.

Moreover, the proposed algorithm aims at fairly distributing UTDs among different data flows. To evaluate this aspect, we have derived the Jain index of the percentage of the data-traffic demand that is not satisfied (i.e., following the proportional fair approach). Jain index provides a measure of the fairness of a given performance x, and is defined as $\frac{(\sum_{i=1}^{N} x_i)^2}{N \sum_{i=1}^{N} x_i^2}$. Figure 7 represents the fairness index of the proposed scheme as a function of the number of small cells in the area. We can see that the proposed scheme achieves a higher proportional fairness, while the other two schemes have almost the same performance. Indeed, even if the HC approach allows distribution of the traffic mainly in high-capacity links, it does not take into account the fair distribution of UTDs among the flows. The behavior is almost constant with the number of cells, there is a little increase, because there are more paths, and hence, a reduction of UTDs that can be better distributed.

When the network has very high capacity, all the traffic (both data and control) is supported, with any algorithm. Nevertheless, in this case it is interesting to evaluate the average delay of the network. We can see that using a more efficient routing algorithm allows reduction of the delay because traffic is routed in less congested links thus speeding up the transmission. With a rough approximation the average delay of the control and data traffic has been estimated as

$$\bar{\delta}^C = \frac{1}{\sum_{(v,w)\in\mathcal{V}^2} d^C_{v,w}} \sum_{e_{i,j}\in\mathcal{E}} \frac{\sum_{(v,w)\in\mathcal{V}^2} d^C_{v,w} p^C_{v,w,e_{i,j}}}{c_{e_{i,j}} - \sum_{(v,w)\in\mathcal{V}^2} d^C_{v,w} p^C_{v,w,e_{i,j}}} \quad (7)$$

$$\bar{\delta}^D = \frac{1}{\sum_{(v,w)\in\mathcal{V}^2}\left(d^C_{v,w} + d^D_{v,w}\right)} \sum_{e_{i,j}\in\mathcal{E}} \frac{\sum_{(v,w)\in\mathcal{V}^2}\left[d^C_{v,w}p^C_{v,w,e_{i,j}} + d^D_{v,w}p^D_{v,w,e_{i,j}}\right]}{c_{e_{i,j}} - \sum_{(v,w)\in\mathcal{V}^2}\left[d^C_{v,w}p^C_{v,w,e_{i,j}} + d^D_{v,w}p^D_{v,w,e_{i,j}}\right]} \tag{8}$$

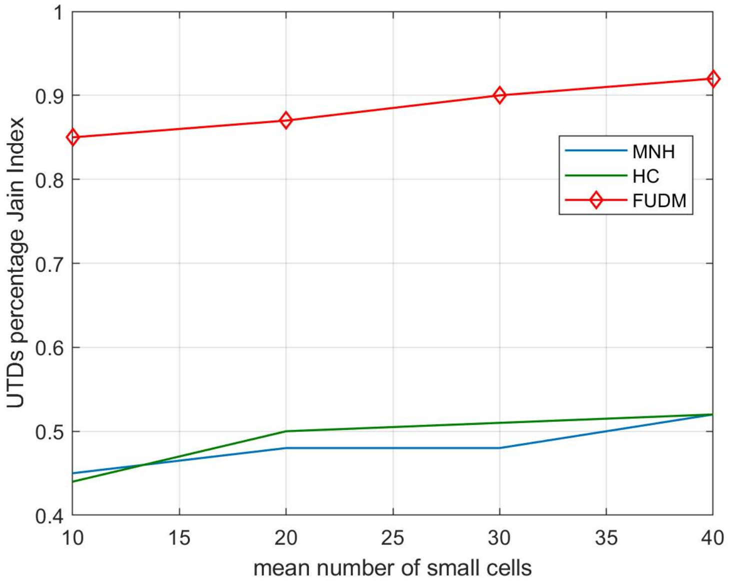


Figure 7. Jain index of the percentage of UTD of each flow respect its data traffic demand.

Table 2 reports the average network delay of the network for the proposed and benchmark methods when the network has high capacity and UTDs is zero for both data and control traffic. We have considered a medium-connected network.

Table 2. Average transmission delay of the network for data traffic.

Mean Number of Cells	FUDM	HC	MNH
20	88 µs	100 µs	105 µs
30	93 µs	109 µs	116 µs
40	101 µs	120 µs	126 µs

8. Conclusions

This paper dealt with backhaul routing in 5G UDN.

UDN paradigm is considered a pillar of future 5G networks but its deployment on large scale is challenging. In particular, UDN deployment could lead to a massive traffic forwarding on the backhaul network that may not be able to support all access-point traffic demands.

This paper investigated the use of the SDN paradigm to have efficient and flexible routing policies. A problem specific for UDNs backhauling has been presented, and a new solution based on a modified Dijkstra algorithm has been proposed with the aim of reducing the traffic demand that the backhaul network is not able to support. The solution is defined for a centralized implementation and for splitting control and data planes following the SDN-approach. In particular, first, the control-flow routes in the backhaul network are identified using a bandwidth-constrained Dijkstra algorithm. The control flows must be always satisfied by the network. The second step allows finding of the routes for the data flows. The algorithm searches successively for each cell for the route that has the minimum cost and that satisfies the data-flow request. If this route does not exist the algorithm looks for the path that minimizes the weighted UTD, i.e., the amount of traffic that the backhaul network is not able to support due to the congestion of one or more links. The UTD is weighted to achieve a fair distribution of the unsatisfied demand among different data flows that are allocated on the congested link.

Presented numerical results have shown the effectiveness of the proposed scheme in case of backhaul network congestion in comparison with traditional shortest-path routing algorithm.

This work showed the effectiveness of a centralized routing protocol that follows the SDN principles. However, the work should be extended to evaluate the network reactivity, meaning that the reaction time to changing conditions should be evaluated taking into account also the overhead introduced by the SDN message flow.

Author Contributions: Conceptualization, D.M. and R.F.; Methodology, D.M.; Software, L.S.; Investigation, D.M.; Data Curation, L.S.; Writing–Original Draft Preparation, D.M.; Writing–Review & Editing, R.F.; Supervision, D.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

5G	fifth-Generation
API	Application Programming Interface
BS	Base Station
CN	Core Network
FD	Forwarding Device
FUDM	Fair Unsatisfied Demands Minimization
HC	Higher Capacity
MNH	Minimum Number of Hops
OSPF	Open Shortest Path First
QoS	Quality-of-Service
RAN	Radio Access Network
SDN	Software-Defined Network
UDN	Ultra-Dense Network
UE	User Equipment
UTD	Unsatisfied Traffic Demand

References

1. Agiwal, M.; Roy, A.; Saxena, N. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1617–1655. [CrossRef]
2. Ge, X.; Tu, S.; Mao, G.; Wang, C.; Han, T. 5G Ultra-Dense Cellular Networks. *IEEE Wirel. Commun.* **2016**, *23*, 72–79. [CrossRef]
3. Marabissi, D.; Bartoli, G.; Fantacci, R.; Pucci, M. An Optimized CoMP Transmission for a Heterogeneous Network Using eICIC Approach. *IEEE Trans. Veh. Technol.* **2016**, *65*, 8230–8239. [CrossRef]
4. Kreutz, D.; Ramos, F.M.V.; Veríssimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76. [CrossRef]
5. Zaidi, Z.; Friderikos, V.; Yousaf, Z.; Fletcher, S.; Dohler, M.; Aghvami, H. Will SDN Be Part of 5G? *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3220–3258. [CrossRef]
6. Nguyen, V.; Brunstrom, A.; Grinnemo, K.; Taheri, J. SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1567–1602. [CrossRef]
7. Costa-Perez, X.; Garcia-Saavedra, A.; Li, X.; Deiss, T.; de la Oliva, A.; di Giglio, A.; Iovanna, P.; Moored, A. 5G-Crosshaul: An SDN/NFV Integrated Fronthaul/Backhaul Transport Network Architecture. *IEEE Wirel. Commun.* **2017**, *24*, 38–45. [CrossRef]
8. Kitindi, E.J.; Fu, S.; Jia, Y.; Kabir, A.; Wang, Y. Wireless Network Virtualization With SDN and C-RAN for 5G Networks: Requirements, Opportunities, and Challenges. *IEEE Access* **2017**, *5*, 19099–19115. [CrossRef]
9. Arslan, M.Y.; Sundaresan, K.; Rangarajan, S. Software-defined networking in cellular radio access networks: Potential and challenges. *IEEE Commun. Mag.* **2015**, *53*, 150–156. [CrossRef]

10. Jaber, M.; Imran, M.A.; Tafazolli, R.; Tukmanov, A. 5G Backhaul Challenges and Emerging Research Directions: A Survey. *IEEE Access* **2016**, *4*, 1743–1766. [CrossRef]
11. Saif, D.; Arsiwala, F.; Khanna, I. Software Defined Networking in Next Generation Mobile Backhauls: A Survey. In Proceedings of the 2018 IEEE 5G World Forum (5GWF), Silicon Valley, CA, USA, 9–11 July 2018; pp. 106–111.
12. Alippi, C.; Fantacci, R.; Marabissi, D.; Roveri, M. A Cloud to the Ground: The New Frontier of Intelligent and Autonomous Networks of Things. *IEEE Commun. Mag.* **2016**, *54*, 14–20. [CrossRef]
13. Jain, A.; Lopez-Aguilera, E.; Demirkol, I. Enhanced Handover Signaling through Integrated MME-SDN Controller Solution. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–7.
14. Chen, C.; Lin, Y.; Yen, L.; Chan, M.; Tseng, C. Mobility management for low-latency handover in SDN-based enterprise networks. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–6.
15. Prados-Garzon, J.; Adamuz-Hinojosa, O.; Ameigeiras, P.; Ramos-Munoz, J.J.; Andres-Maldonado, P.; Lopez-Soler, J.M. Handover implementation in a 5G SDN-based mobile network architecture. In Proceedings of the 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 4–8 September 2016; pp. 1–6.
16. Chartsias, P.K.; Amiras, A.; Plevrakis, I.; Samaras, I.; Katsaros, K.; Kritharidis, D.; Trouva, E.; Angelopoulos, I.; Kourtis, A.; Siddiqui, M.S.; et al. SDN/NFV-based end to end network slicing for 5G multi-tenant networks. In Proceedings of the 2017 European Conference on Networks and Communications (EuCNC), Oulu, Finland, 12–15 June 2017; pp. 1–5.
17. Ordonez-Lucena, J.; Ameigeiras, P.; Lopez, D.; Ramos-Munoz, J.J.; Lorca, J.; Folgueira, J. Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Commun. Mag.* **2017**, *55*, 80–87. [CrossRef]
18. Park, G.S.; Song, H. Cooperative Base Station Caching and X2 Link Traffic Offloading System for Video Streaming over SDN-enabled 5G Networks. *IEEE Trans. Mob. Comput.* **2018**, *99*, 1. [CrossRef]
19. Liu, Y.; Point, J.C.; Katsaros, K.V.; Glykantzis, V.; Siddiqui, M.S.; Escalona, E. SDN/NFV based caching solution for future mobile network (5G). In Proceedings of the 2017 European Conference on Networks and Communications (EuCNC), Oulu, Finland, 12–15 June 2017; pp. 1–5.
20. Rostami, A.; Ohlen, P.; Wang, K.; Ghebretensae, Z.; Skubic, B.; Santos, M.; Vidal, A. Orchestration of RAN and Transport Networks for 5G: An SDN Approach. *IEEE Commun. Mag.* **2017**, *55*, 64–70. [CrossRef]
21. Guck, J.W.; Bemten, A.V.; Reisslein, M.; Kellerer, W. Unicast QoS Routing Algorithms for SDN: A Comprehensive Survey and Performance Evaluation. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 388–415. [CrossRef]
22. Gopi, D.; Cheng, S.; Huck, R. Comparative analysis of SDN and conventional networks using routing protocols. In Proceedings of the 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, 21–23 July 2017; pp. 108–112.
23. Guo, Y.; Wang, Z.; Yin, X.; Shi, X.; Wu, J. Optimize Routing in Hybrid SDN Network with Changing Traffic. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–8.
24. Ramdhani, M.F.; Hertiana, S.N.; Dirgantara, B. Multipath routing with load balancing and admission control in Software-Defined Networking (SDN). In Proceedings of the 2016 4th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 25–27 May 2016; pp. 1–6.
25. Tomovic, S.; Radusinovic, I. Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks. In Proceedings of the 2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, Korea, 6–10 June 2016; pp. 303–311.
26. Yao, Y.; Tian, H.; Nie, G.; Wu, H.; Jin, J. Multi-path Routing Based QoS-aware Fairness Backhaul-Access Scheduling in mmWave UDN. In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1–7.
27. Liu, H.; Hao, S.; Li, J. Routing and Heuristic Scheduling Algorithm for Millimeter Wave Wireless Backhaul Networks. In Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 December 2017; pp. 300–304.

28. Seppänen, K.; Kilpi, J.; Paananen, J.; Suihko, T.; Wainio, P.; Kapanen, J. Multipath routing for mmWave WMN backhaul. In Proceedings of the 2016 IEEE International Conference on Communications Workshops (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 246–253.
29. Zola, E.; Kassler, A.J.; Kim, W. Joint User Association and Energy Aware Routing for Green Small Cell mmWave Backhaul Networks. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
30. Pateromichelakis, E.; Shariat, M.; Quddus, A.U.; Tafazolli, R. Joint routing and scheduling in dense small cell networks using 60 GHz backhaul. In Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW), London, UK, 8–12 June 2015; pp. 2732–2737.
31. Al-Sadi, A.M.; Al-Sherbaz, A.; Xue, J.; Turner, S. Routing algorithm optimization for software defined network WAN. In Proceedings of the 2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA), Baghdad, Iraq, 9–10 May 2016; pp. 1–6.
32. Nunez-Martinez, J.; Mangues-Bafalluy, J.; Baranda, J. Anycast Backpressure Routing: Scalable Mobile Backhaul for Dense Small Cell Deployments. *IEEE Commun. Lett.* **2013**, *17*, 2316–2319. [CrossRef]
33. Bartoli, G.; Marabissi, D.; Pucci, R.; Ronga, L. AI Based Network and Radio Resource Management in 5G HetNets. *J. Signal Process. Syst.* **2017**, *89*, 133–143. [CrossRef]
34. Marabissi, D.; Bartoli, G.; Stomaci, A. Low-Complexity Distributed Cell-Specific Bias Calculation for Load Balancing in UDNs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1056–1060. [CrossRef]
35. Dijkstra, E.W. A note on two problem in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]