Time-sensitive Multi-Flow Routing in Highly Utilized MANETs

Klement Hagenhoff, Eike Viehmann, and Gabi Dreo Rodosek
Universität der Bundeswehr München
{klement.hagenhoff, eike.viehmann, gabi.dreo}@unibw.de

***Abstract*—MANETs comprise several mobile nodes, wirelessly connected with each other. These networks are self-organized, each participant is responsible for routing and data forwarding. Routing protocols only provide local or outdated topology knowledge because participants are moving continuously. Also, transmission capacities are limited which often results in over-utilized network segments. Capacity conform path distribution is challenging since nodes route based on their incomplete topology knowledge. Recent work showed that an up-to-date and complete network topology representation can quickly be delivered to a controller, which is instantiated on an arbitrary node. Now, routing and path deployment can be outsourced to the controller. With this knowledge, we introduce several path finding approaches to answer the question if and to which extent non over-utilizing routes for several flows can be found where common MANET routing techniques would fail. Also, paths have to be computed quickly since topologies change due to the mobility of nodes. Our path finding techniques also focus on routes aiming for long connection lifetime. We compare our approaches regarding capacity usage, computation times, and connection lifetimes, taking into consideration typical MANET behavior.**

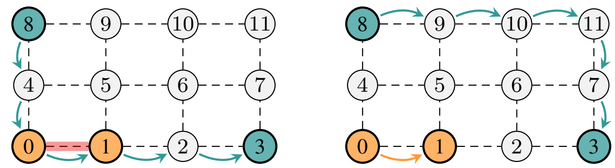
***Index Terms*—MANET, Path Finding, MIP, GP**

I. Introduction and Problem Description

Mobile ad-hoc Networks (MANETs) per technical definition comprise several mobile devices, for instance, smartphones or Internet Protocol (IP) radios, that build up a wireless network where no central instance is in place. In contrast to traditional network architectures, routing and data delivery are carried out by participants which characterizes these networks as self-organized [1]. Nodes are moving, leading to changing connections and network topologies. The distributed routing behavior reduces the transparency which is why MANETs find application in Hong Kong's political demonstrations [2] as well as in military missions where a single point of failure is very risky [3]. Also, MANETs are according to simulations frequently debated to be a promising alternative in catastrophic situations, such as forest fires or flash floods, where electricity of base stations is prone to fail at any time [4].

However, MANETs only provide restricted transmission capacities on the wireless channel, especially in the military field [5]. As a consequence, data paths are often over-utilized because participants route packets based on incomplete or outdated topology information.

The simplified example in Figure 1(a) demonstrates the incomplete routing knowledge of nodes. Suppose flow $f_{(8,3)}$ is already deployed and is relayed via nodes 4, 0, 1, and 2. Link



(a) Path of flow $f_{(8,3)}$ is deployed. Flow $f_{(0,1)}$ from 0 to 1 would over-utilize the link shown in red.

(b) Rerouting of flow $f_{(8,3)}$ to provide sufficient capacities to route flow $f_{(0,1)}$ from 0 to 1.

Fig. 1. Grid MANET topology to illustrate occuring over-utilized situation due to incomplete or outdated topology knowledge of nodes. A detailed description of how link utilizations are computed is provided in Section III-C. Black dashed lines illustrate connections (links) between nodes. Solid arcs display deployed paths from source to target which generate the utilization.

$(0, 1)$ is completely utilized which means that the available transmission capacity is entirely exhausted. At a later time, $f_{(0,1)}$ must also be deployed. This transmission from 0 to 1 would generate an over-utilized situation.

To overcome such a flow deployment problem, recent research showed that routing information, like up-to-date topology information and capacity characteristics of links can be delivered to an arbitrary participant in the MANET [6] or to an outsourced routing instance if a dedicated channel is available [7]. Based on that, the controller can receive route requests by participants and redistribute all flows in order to relieve over-utilized segments. According to the example in Figure 1(b), after receiving the route request from node 0 to destination 1, the controller could redistribute flow $f_{(8,3)}$ to provide transmission capacities for flow $f_{(0,1)}$.

Our contribution in this paper is to answer the question if and how efficiently the global knowledge of a controller helps to compute capacity conform paths of multiple flows if basic routing leads to over-utilized situations. Unlike wired networks, MANETs have restricted transmission capacities and nodes are moving. These force path finding techniques to respond quickly while focusing on long-living and utilization efficient paths, which to our best knowledge contributes the following novelties: We present several self-developed and extended path finding techniques to tackle the mentioned MANET challenges. These techniques are evaluated and compared regarding computation runtimes, number of found capacity conform paths, and robustness in terms of connection life times. For robust connections, we present a link metric that characterizes these as long-living.

In addition, we contribute a tailored evaluation environment

978-3-903176-51-5

to apply all path finding techniques in equal preconditioned situations. In particular, several MANET topologies including realistic connection characteristics as well as mobility patterns are constructed having paths deployed that over-utilize the network's transmission capacity. The objective is twofold: On the one hand to perform rerouting to relieve these over-utilized segments and on the other hand to select paths that seem to be long-living in terms of connection quality.

The remainder of this paper is organized as follows: Background is introduced in Section II. Section III discusses the flow distribution workflow and also preliminaries of the network architecture. Thereafter, Section IV elaborates on all path finding techniques. Results are being discussed in Section V. We close with a conclusion and thoughts on future work in Section VI.

II. Background

This section introduces general characteristics of the applied path finding techniques and discusses previous research.

The described challenge is known as Multi-Commodity Network Flow Problem (MCNFP) which was first introduced by Ford and Fulkerson [8] and has widely been researched over decades, see [9], [10] as well as nowadays [11]. MCNFPs describe optimization problems where multiple paths with certain constraints have to be found focusing on the same optimization criteria described by either minimization or maximization function. This paper focuses especially on the unsplittable MCNFP which sticks to a single path per flow. Research has shown, that splitting a flow into multiple paths in wireless networks achieves only negligible performance increase [12].

The MCNFP applied in wireless networks such as MANETs has barely been investigated so far as detected in the up-to-date survey of K. Salimifard [11] dealing mostly with wired network applications. Characteristics of wired connections overlap only marginal with wireless connection behavior. The difference becomes even more apparent when mobility of network participants comes into play which, as a sidenote, distinguishes our controller-equipped routing approach from Software-Defined Networking (SDN).

Often used solving techniques are exact methods like Branch & Cut and Branch & Bound [13], applied by solvers like the IBM Cplex Modeler[1]. Such techniques guarantee global optimal solutions if runtime is not a major concern. Otherwise, the runtime can be restricted and results till that point are returned if exist. The majority of previous research tackles the MCNFP with exact methods because of reliability.

Heuristic approaches are more tailored when going for minimal runtime results since these techniques can be more easily customized to suit the desired purpose. For instance, the speed of execution can be increased with search space boundaries like defining maximum path costs [14] or minimum path similarity constraints [15]. Previous research exhibits promising results [16]–[19], which is why heuristics have been investigated in this context as well. Akin et al. [20] followed such an approach and proposed a Breadth-First Dijkstra combination to detect and avoid links that are chosen for several flows with high probability. Based on their approach, transmission resources can be exploited more efficiently which is proven by their evaluations.

Nature-inspired Meta-Heuristics, such as Ant Colony Optimization (ACO) [21] and Genetic Programming (GP) [22] pursue more unsupervised approaches that are often chosen to generate more diversity which increases the probability of better results. Genetic Algorithms (GA) convert a population of individuals with their respective fitness through operations including crossover and mutation following the Darwinian principle. GP is an artificial approach mimicking evolution in computer science and has been extensively studied for decades [22]–[24]. GP terminates based on a defined criterion, such as the number of generations or quality of found solutions. Solving MCNFP with GP is promising since initial populations can be generated in a way to cover both fast and efficient solution creation and high diversity regarding various paths. Masri et al. [25] and Alavidoost et al. [26] applied and designed several GP approaches and their evaluations show promising results.

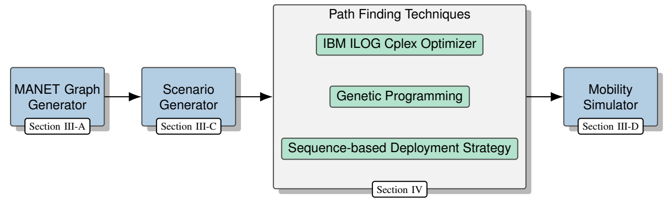


Fig. 2. MANET flow distribution framework

III. MANET Graph Model

To compute paths of several flows, we designed and implemented a MANET flow distribution workflow that comprises all steps which are necessary to determine if paths can be found that do not over-utilize the MANET transmission capacity. This workflow is illustrated in Figure 2.

The *MANET Graph Generator* creates various topologies that represent MANETs including computation of link and node properties. The *Scenario Generator* generates over-utilized situations in the previously computed MANETs. *Path Finding Techniques* comprises all approaches that find a setting that solves the previously generated over-utilized situations. Next, the *Mobility Evaluator* simulates the movement and speed of participants before and after path setting deployment, to evaluate how long the computed paths are connected.

All steps from creating topologies to simulating mobility as well as all path finding techniques are bundled in Java projects available on GitHub[2][3][4]. `JGraphLib` represents the graph library which transforms the MANET into a directed graph.

[1]https://www.ibm.com/de-de/products/ilog-cplex-optimization-studio

[2]https://github.com/F1-C0D3/JGraphLib
[3]https://github.com/F1-C0D3/MANETModel
[4]https://github.com/F1-C0D3/MANETRoutingInstance

`MANETModel` extends `JGraphLib` with MANET properties like realistic transmission channels, data rates of participants, mobility models among others. `MANETRoutingInstance` contains all path finding techniques.

A. MANET Graph Generator

The *MANET Graph Generator* provides various types of topologies varying in the number of nodes as well as the graph density which together results in the number of connections of participants as well as distances between them.

B. MANET Properties

The MANET's topology is described as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ are the set of nodes and $\mathcal{L}$ the set of links in the network. The former represents participants of the MANET while the latter represents connections between them. To be more precise, behind each vertex in the graph exists a MANET participant which is connected via a wireless channel with surrounding nodes. Connectivity between participants is represented via edges of the graph. $\mathcal{G}$ is defined as a directed graph without loops, meaning $\forall (x, y) \in \mathcal{L} : x \neq y$.

Node and Link Properties Each node is equipped with an antenna whose radio waves propagate omnidirectional. Also, there exists one unique communication channel all participants access via Carrier Sense Multiple Access Collision Avoidance (CSMA/CA) [27] to transmit and receive data. As a consequence, if an arbitrary node accesses the channel to transmit payload, all surrounding nodes within communication range will have to stay idle. Otherwise, the transmissions are interfered and corrupted. A link $(n, m), \in \mathcal{L}$ between the nodes n and m exists, if the received radio waves of a signal transmitted by n are above the configured sensitivity threshold of the antenna of m. Each link $l \in \mathcal{L}$ has a transmission capacity c_l which varies according to the reception power r_l. c_l is based on the 2.4 Ghz band and is computed with the Shannon-Hartley theorem [28] to generate realistic conditions. Also, each link has the property utilization u_l which increases by at least the data rate of flow (d^f) if flow f traverses via link l. A link l is over-utilized, if the utilization u_l is greater than the transmission capacity ($u_l > c_l$). The utilization model is formally described in Section III-C.

C. Scenario Generator

This section formally describes flows that are deployed on the MANET topologies. We also look at how over-utilization scenarios are generated. Generating paths that over-utilize the MANET is necessary to evaluate how proposed path finding techniques relieve these stressed network segments. Next, the utilization model which represents the propagation of radio waves is introduced followed by the link quality characteristics.

Flow Characteristic Participants request routes to destinations in order to deliver their data. Flows are formally described as $\mathcal{F} = \{f_0, \dots f_{|\mathcal{F}|-1}\}$. Such an f_h has a source s^{f_h} and a target node t^{f_h}, where $s^{f_h}, t^{f_h} \in \mathcal{N}$. Each f_h is assigned the requirement d^{f_h} which represents the configured datarate (Mbit/s). Also, each f_h has a loopless path $p^{f_h} = \langle n_0, \dots, n_{|p^{f_h}|-1} \rangle$ which defines a sequence of nodes, where $\forall n_i, n_j \in p^{f_h} : n_i \neq n_j \land (n_i, n_{i+1}) \in \mathcal{L}$. In addition, f_h has a path quality q^{f_h} which represents the connection lifetime of the path. Path and link qualities are described in Section III-C. In the subsequent sections f always represents an element of $\mathcal{F}$.

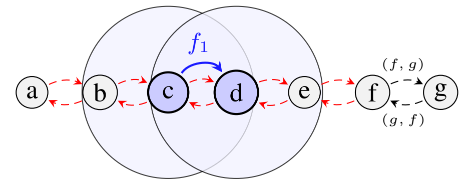


Fig. 3. Occuring link utilization based on d^{f_1} of flow f_1 from nodes c to d using CSMA/CA. Nodes b and e passively receive RTS and CTS, respectively, since both nodes are inside communcation ranges of c and d (bright blue circles). Hence, b and e are occupied to not disturb frame exchange. As a consequence, node e would not respond to a transmission request of node f since e has previously received the control message of d. Neither would b. It follows that all red links are utilized due to the transmission of flow f_1.

Utilization Model As mentioned earlier, a single channel is assumed that all participants access to transmit and receive data. Various media access control protocols exist to distribute the available capacity. CSMA/CA is one of the most frequently used protocols since it is scalable and manages the channel access self-organized which is suitable for decentralized networks [27]. CSMA/CA uses control frames Request-to-Send (RTS) and Clear-to-Send (CTS) to ensure that the channel is reserved for data delivery. Surrounding nodes also receive such control frames and stay idle as long as communication takes place. Thus, all nodes which receive such frames will not be able to receive or transmit data. Figure 3 explains, which links (red colored links) increase their utilization u_l based on d^{f_1} of flow f_1 using CSMA/CA.

More formally, $N(n) := \{m \in \mathcal{N} | (m, n) \lor (n, m) \in \mathcal{L}\}$ represents the set of nodes which are directly connected to n. $N(n, m) := N(n) \cup N(m)$, where $(n, m) \in \mathcal{L}$ unions all neighboring nodes. Based on that, the set of utilized links $\mathcal{U}(n, m)$ with respect to link $(n, m) \in \mathcal{L}$ is determined with Equation 1.

$$\mathcal{U}(n, m) := \{(i, j) \in \mathcal{L} | i \in N(n, m) \lor j \in N(n, m)\} \quad (1)$$

Each link $l' \in \mathcal{U}(l)$ increases its utilization $u_{l'}$ based on the configured data rate d^f of an arbitrary f. In addition, it can be verified if and how much capacity is left based on the chosen paths of all flows and also if the computed paths exceed the transmission capacity of any link that actively transmits data. Our research assumes a utilization monitoring concept on the Media Access Control (MAC) that obtains the utilization of links. Research of Streit et al. [29] propose a concept for wireless networks which could be applied to this work.

Link Quality Despite finding non over-utilizing path settings, the objective is also to identify links which seem to be long living in terms of connectivity. This is important since mobility of nodes is the most frequent reason for link breaks. Streit et al. [30] showed that promising indicators of long living connections are reception power, velocity of nodes, and directions of nodes. The link quality of a link $l \in \mathcal{L}$ is defined as q_l where $0 \leq q_l \leq 1, q \in R$. q is a normalized value which indicates a high link quality of l if q_l is minimal. Equation 2 shows the composition.

$$q_{(n,m)} = z(r_{(n,m)}) \cdot w_z + s(m) \cdot w_s + a(n,m) \cdot w_a \tag{2}$$

Function $z(r_{(n,m)})$ represents the connection strength of link (n,m) which is based on the reception power $r_{(n,m)}$. Speed $s(m)$ of m is computed using the weighted moving average. Function $a(n,m)$ represents the angle of movement directions of nodes n and m. Weights (w) associated to each function increase and decrease the value of each property. We obtained the best weight setting during test runs and used it for all evaluations.

Based on q_l, the path quality of f is represented by q^f which is the sum of all links obtained from $n \in p^f$, see Equation 3.

$$q^f = \sum_{i=0}^{|p^f|-1} q_{(n_i,n_{i+1})} \tag{3}$$

Over-Utilized Situation Generation An over-utilized situation is present if at least one transmitting $l \in \mathcal{L}$ exceeds its predefined transmission capacity ($u_l > c_l$).

The *Scenario Generator* initializes $d^f = 0, \forall f \in \mathcal{F}$ and computes paths of all flows with Dijkstra [31] using q_l as shortest path metric. The next step is to deploy and undeploy all paths round based while successively increasing d^f of all f. Deploying paths reduces the MANET transmission capacity since individual links increase their utilization u_l based on the utilization model, see Section III-C. Undeploying results in the opposite. The over-utilization generator continues until the defined over-utilization threshold (measured in %) is reached. Thereafter, the setting containing all flows with source-target pairs and their determined d^f, as well as the used MANET topology is returned. Based on that setting, Section IV deals with path finding techniques that try to find paths of these flows with the objective not to over-utilize any transmitting link in the MANET.

D. Mobility of Nodes

Our framework provides the Random Waypoint and the Gauss-Markov Mobility since these models are well suited for MANETs [32]. The former defines speed and direction of nodes randomly. After elapsed times, nodes randomly determine directions and speeds and move accordingly. The latter takes previous speeds and directions of each participant into consideration when computing next mobility ticks.

Nodes move based on the configured model before path computation takes place and continue moving thereafter. After computation, we measure the time, that paths stay connected.

Algorithm 1: Path distribution using SbDS

```
1  begin EVOLVESbDS(F)
2      s = ⟨⟩
3      while |s| < |F| do
4          q^best = ∞
5          foreach f ∈ {f ∈ F | p^f ∉ s} do
6              q^f, p^f = SCOREANDPATHOF(f)
7              if q^f ≤ q^best then
8                  q^best = q^f
9                  p^best = p^f
10             end
11         end
12         DEPLOYPATH(p^best)
13         s ← p^best
14     end
15     return s
16 end
```

IV. Proposed Path Finding Techniques

This section introduces path finding techniques applied to relieve initial generated over-utilized situations. The developed approach named Sequence-based Deployment Strategy (SbDS) focuses on minimum runtimes to respond quickly to route requests. GP is designed to provide fast runtimes and high quality results whereas IBM ILOG Cplex Optimizer (Cplex) prioritizes high quality results, but can be configured to stop after predefined runtimes.

A. Sequence-based Deployment Strategy

SbDS uses the tree based Greedy Approach to determine local best path quality scores which is shown with Algorithm 1. Between lines 5 to 11, the algorithm successively determines the path p^f for each flow with the best q^f. In particular, the algorithm computes p^f and q^f for each flow using function SCOREANDPATHOF(f) applying Dijkstra [31] with q_l as metric and stores the path with best q^f in p^{best}. SCOREANDPATHOF(f) sets q^f to ∞ if the MANET would be over-utilized with the deployment of p^f. After p^{best} has been identified, its path is deployed which affects further path finding processes since links of the MANET increase their utilization and might not provide sufficient transmission capacities for further flows. Lastly, p^{best} is stored in s which represents the sequence of found path settings for all flows. This also has the effect that p^{best} is not considered for further greedy path finding operations.

To sum up, SbDS selects best link quality paths of all flows and deploys the paths with best computed score in the MANET. After one path has been deployed, SbDS determines the best path for all residual flows based on the current MANET utilization situation. However, this approach does not identify conflict links which have high probability to be over-utilized after all flows are deployed, since each path of flows is computed individually.

B. IBM ILOG Cplex Optimizer

Cplex is an optimization technique that approximates towards the global optimum if no runtime limitation is configured. Because of that, Cplex can prove if a capacity conform path setting exists. Runtimes matter though since MANET participants move continuously, which changes the topology. Cplex is applied in this research because runtimes can be restricted. This might reduce the quality as well as the number of found solutions. Solvers like Cplex include the formalization of the optimization objective which comprises of an optimization function and its associated constraints. The following optimization model aims to find robust paths for an arbitrary number of flows that do not over-utilize the MANETs capacities. In doing so, optimization function 4 minimizes the link qualities q_l of links which are chosen for data delivery. As mentioned in Section III-C, a link l is supposed to be robust if q_l is minimal.

$$\min_{(x)} \sum_{f\in\mathcal{F}} \sum_{l\in\mathcal{L}} x_l^f q_l \tag{4}$$

x_l^f elects those links $l \in \mathcal{L}$ whose q_l is minimal regarding all flows f. This binary variable is set to 1 if link l forwards data of flow f, otherwise 0, see constraint 5.

$$x_l^f \in \{0,1\}, \forall f \in \mathcal{F}, \forall l \in \mathcal{L} \tag{5}$$

Expression 6 describes an additional binary variable which is set to 0 if any flow f traverses the link l, otherwise 1. y_l is initially set to 1.

$$y_l \in \{0,1\}, \forall l \in \mathcal{L} \tag{6}$$

Constraint 7 verifies if link l is elected for transmission of any flow f and sets the binary y_l accordingly. This is important since each $l' \in \mathcal{U}(l)$ increases its utilization based on the data rate d^f of flow f if $l \in \mathcal{L}$ is elected to tramsit data. The passive utilization must be considered if l' is also elected to forward any flow.

$$0 < (\sum_{f\in\mathcal{F}} x_l^f) \Rightarrow y_l = 0, \forall l \in \mathcal{L} \tag{7}$$

Constraint 8 defines not to over-utilize any transmitting $l \in \mathcal{L}$ with data rate demand d^f, if x_l^f of any flow is set to 1. This also takes passive utilization into account.

$$\sum_{f\in\mathcal{F}} \sum_{l'\in\mathcal{U}(l)} d^f (x_{l'}^f - y_l) \le c_l, \forall l \in \mathcal{L} \tag{8}$$

Constraint 9 defines the range of q_l.

$$0 \le q_l \le 1, \forall l \in \mathcal{L} \quad , q \in R \tag{9}$$

The set $N^+(m) := \{n \in \mathcal{L} | (n,m) \in \mathcal{L}\}$ returns all incoming neighbors of node m. $N^-(m) := \{n \in \mathcal{L} | (m,n) \in \mathcal{L}\}$ returns all outgoing nodes of m.

$$\sum_{m\in N^+(m)} x_{(n,m)}^f d^f - \sum_{n\in N^-(m)} x_{(m,n)}^f d^f = \begin{cases} 0, m \in \mathcal{L} \setminus \{s^f, t^f\} \\ -d^f, m = s^f \\ d^f, m = t^f \end{cases} ,\forall f \in F \tag{10}$$

Constraint 10 verifies that the data rate d^f of each flow f that reaches node m must be equal to the amount of data rate that leaves note m, if m is not source (s^f) or target (t^f) of f. Otherwise, the amount of data rate must be $-d^f$ or d^f.

$$\sum_{n\in N^+(m)} x_{(n,m)}^f + \sum_{n\in N^-(m)} x_{(m,n)}^f = \begin{cases} 2, m \in \mathcal{L} \setminus \{s^f, t^f\} \\ 1, m = s^f \\ 1, m = t^f \end{cases} ,\forall f \in F \tag{11}$$

This ensures that Cplex constructs loopless paths from s^f to t^f that are completely connected. Constraint 11 ensures that only a single path is constructed from s^f to t^f regarding each $f \in \mathcal{F}$.

C. Genetic Programming

GP is characterized by its ability to create new solutions based on existing ones. Additionally, this learning enriched approach enables a compromise between diversity and minimum runtime. The following sections introduce the modifications to obtain path settings that represent possible solutions.

Initial Population Initial populations are often created at random. Each individual represents a possible solution. This set is thereafter used as starting point to create modified and improved solutions. Applied to our objective, the initial population $I := \{i_0, \ldots, i_{|I|-1}\}$ defines a set of possible solutions. For GP we now define a list of paths $P^{f_h} = \{p_0^{f_h}, \ldots, p_{|I|-1}^{f_h}\}$ for each flow, instead of a single path (p^{f_h}) for each flow. Each path is equally defined as in Section III-C. Based on that, each individual $i_j := \langle p_j^{f_0}, \ldots, p_j^{f_{|\mathcal{F}|-1}} \rangle$, where $i_j \in I$ represents a sequence of ordered paths where each path in i_j corresponds to exactly one $f \in \mathcal{F}$. The $i_j \in I$ with the best fitness value represents the result of GP in case a capacity conform path setting is found.

Test runs have proven that strictly relying on randomness is not promising. However, not to give up diversity but also aim to evolve capacity conform results, the population is generated with random paths as well as with k-shortest paths [33]. The number of entries in I depends on the objective. Its entries change or rather are modified during runtime since crossover and mutation techniques are applied. The more entries in I the higher the probability to generate high quality results. However, the number of entries in I influences the runtime of GP.

Evaluate Fitness Initial populations as well as during process generated solutions are evaluated in order to distinguish between promising and inadequate solutions. The Fitness Function 12 computes a score which rates each $i \in I$.

$$\lambda(i) = \rho(i) + o(i) + \upsilon(i) \tag{12}$$

$\rho(i)$ returns an averaged quality score over all links used by all $p^f \in \mathcal{F}$, described in Equation 13.

$$\rho(i) = \sum_{p^f \in i} \frac{q^f}{|p^f|} \tag{13}$$

Solutions must compute non over-utilizing paths settings in the first place before going after path or link quality score improvements. Hence, $\lambda(i)$ also evaluates i based on the arisen utilization which is determined with $\upsilon(i)$. In particular, $\upsilon(i)$

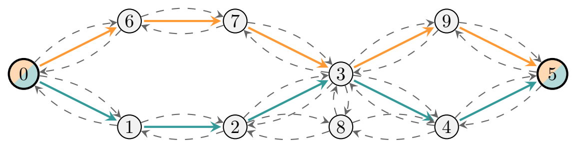


Fig. 4. MANET graph example to introduce Crossover and Mutation. Orange and teal arcs illustrate possible paths $p_a^{f_h}$ and $p_b^{f_h}$ from nodes 0 to 5, respectively. Directed dashed arcs represent edges (links) between nodes

represents the actual produced utilization of i compared to the utilization of all links that would be utilized by $d^f, \forall f \in \mathcal{F}$ including all passively utilized links, see Equation 14.

$$\upsilon(i) = \frac{\sum_{p^f \in i} \sum_{l \in p^f} |\mathcal{U}(l)| \cdot d^f}{\sum_{f \in \mathcal{F}} \sum_{l \in \mathcal{L}} |\mathcal{U}(l)| \cdot d^f} \tag{14}$$

To ensure that good results computed by $\lambda(i)$ exclusively characterize capacity conform path settings, $o : i \rightarrow \{0, 1\}$ returns 1 if setting $i \in I$ creates an over-utilized MANET situation and 0 if otherwise. Over-utilizing settings are rated worse to exclude them from further computations.

Crossover During each generation, crossover operations take place on specific $i \in I$ which are elected by the configured Tournament Selection which prefers better solutions over worse ones. This approach extends the One-Point Crossover technique where crossover points are defined randomly. Newly constructed path-based chromosomes must represent exclusively complete paths from source to target after crossover operations. In case crossover takes place on an individual, all paths of this individual will be considered for this operation.

Figure 5 illustrates the custom path conform crossover technique where $i_a, i_b \in I$ are selected for crossover operation. We only introduce the crossover operation for paths $p_a^{f_h}$ and $p_b^{f_h}$ of i_a and i_b of flow f^h as all others follow the same procedure. GP searches after common nodes in $p_a^{f_h}$ and $p_b^{f_h}$ excluding source and target nodes of f^h to combine a new path out of both. To maintain coincidence and diversity that make up GP, one common node of set $C_{ab}^{f_h} = (p_a^{f_h} \cap p_b^{f_h}) \setminus \{s^{f_h}, t^{f_h}\}$ is randomly chosen as crossover point. Suppose node $c \in C_{ab}^{f_h}$ is chosen which is node 3 according to Figure 5. The next

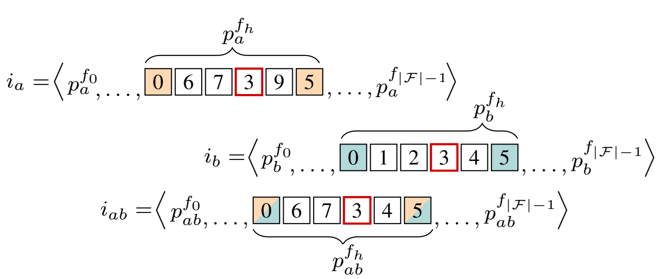


Fig. 5. Crossover operation with paths $p_a^{f_h}$ and $p_b^{f_h}$ of exemplary MANET in Figure 4. GP searches after common nodes in both paths for crossover operation to maintain complete paths from source to destination. Node 3 is selected, which results in path $p_{ab}^{f_h}$ consisting of the left part of $p_a^{f_h}$ and the right part of $p_b^{f_h}$.

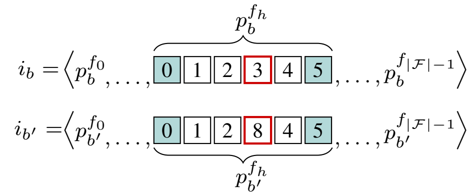


Fig. 6. GP randomly selects node 3 of $p_b^{f_h}$ for mutation process. Next, mutation process searches after directly connected nodes of node 3 that also complete $p_b^{f_h}$ from node 0 to 5, which are $3, 8, 1,$ and 5 according to Figure 4. Also, selected nodes must not be participant of $p_b^{f_h}$. Node 8 fulfills requirements and is exchanged with node 3 resulting in $p_{b'}^{f_h}$.

step is to extract a complete path out of both chromosomes. Therefore, GP combines the left part of $p_a^{f_h}$ till c (excluding c) with the right part of $p_b^{f_h}$ starting at node c which generates the new path $p_{ab}^{f_h}$. No modification is performed, in case $C_{ab}^{f_h} = \emptyset$.

Mutation If mutation takes place at a randomly chosen individual, GP modifies one node of every path in this individual.

Figure 6 illustrates the single node mutation process of individual i_b based on sample path $p_b^{f_h}$. To begin, a random node n_r is picked of $p_b^{f_h}$ which is not source or target node. In order to meet the complete path requirement, all neighboring preceding (n_{r-1}) and successive (n_{r+1}) nodes of n_r are elected that are not part of $p_b^{f_h}$. The result is stored in the set of mutable nodes $M_b^{f_h}$, where each element is potentially exchangeable with n_r since complete path connectivity from source to target of $p_b^{f_h}$ is given. $M_b^{f_h}$ is defined with Equation 15, where $\{n_0, \dots, n_{|p_b^{f_h}|-1}\}$ is the set representation of $p_b^{f_h}$.

$$M_b^{f_h} = (N^-(n_{r-1}) \cap N^+(n_{r+1})) \setminus \{n_0, \dots, n_{|p_b^{f_h}|-1}\} \tag{15}$$

This mutation process takes place at all $p_b^{f_h} \in i_b$ which creates $i_{b'}$. Also, if $|p_b^{f_h}| < 3$ or if $M_b^{f_h}$ is empty, this custom mutation process generates a new random path for this flow.

Termination Condition GP terminates based on the following criteria: (i) Runtime or number of generations exceed its configuration. (ii) Any individual reaches a certain fitness value. (iii) Same best fitness level is computed successively after multiple generations (can be configured).

V. Evaluation

Our evaluation is threefold. The first part compares all approaches regarding runtimes and found solutions. Secondly, we investigate which approach obtains the best link quality score regarding selected links that form the paths. Last but not least, we verify if link quality scores guarantee longer lifetimes of connected paths.

All approaches mentioned in Section IV use equal simulation configurations with regard to MANET topologies, source-target pairs, as well as data rates of flows. The average node degree is roughly 5.5 which corresponds to a graph density of 0.0726. Also, the physical distance between neighbors is between $[35.0, 100.0]$ meters. This configuration offers best conditions to stand out pros and cons of all approaches since

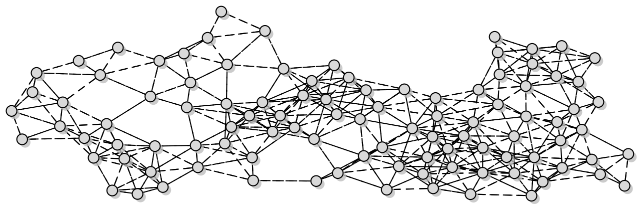


Fig. 7. Example MANET constructed with our *MANET Graph Generator*

these topologies offer the most alternative paths. An example topology is visualized with Figure 7. As mentioned earlier, the radio model resembles the 2.4 GHz band providing approx. 12 Mbit/s which slightly varies on link distances. Nodes move based on the Gauss-Markov Mobility Model between $[2km/h, 7km/h]$.

Early results of SbDS have shown that applying link quality score q_l as mentioned in Section III-C results in worse number of found capacity conform paths compared to the link quality score $q'_l = 1$. This score basically computes the path for each flow with least number of transmitting links. The exact results of these findings are illustrated in Figure 8 comparing found path settings and associated link quality scores of SbDS and SbDS' using $q'_l = 1$, respectively. SbDS' on the other hand decreases the robustness of each link which is shown in Plot 8(b). Results in this plot are inverted for better readability which means higher results represent better link qualities. Both, SbDS' and SbDS are discussed during successive sections since the different link quality scores of both approaches can be used to determine if better link qualities increase the life time of paths.

A. Runtimes and Found Solutions

For this evaluation, we generated 100 topologies containing 100 nodes, based on mentioned configuration parameters. Figure 9 shows on the left side the number of found capacity conform path settings with 5% and 15% initial over-utilization and 5 and 15 flows, respectively. All approaches find almost the same number of valid paths with 5% initial configured over-utilization. However, that differs with 15%, since the search space does provide least valid solutions which are harder to find. SbDS and SbDS' find notably less path settings since no path combinations are performed. This is due to both approaches do not change the already defined path deployment sequence even if the current computed path as well as successor paths over-utilize the MANET. Cplex outperforms all other approaches with increasing over-utilization. All runs of Cplex are limited to 8 min runtime which is sufficient to find capacity conform path settings.

The right plot of Figure 9 shows runtimes with 5 flows and 5% initial over-utilization and with 15 flows an 15% over-utilization. SbDS, SbDS', as well as GP do not exhibit notable deviations. SbDS and SbDS' increase only slightly since local search space exploration is not time consuming. GP relies on the Termination Condition mentioned in Section IV-C which results in more uncontrolled runtimes. The pre-configured runtime of 8 min are almost fully utilized by Cplex especially when paths of 15 flows must be computed.

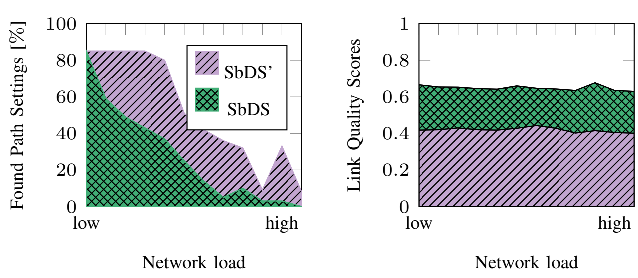
Path computation times of our approaches matter, since our focus lies on MANETs whose topologies change continuously due to mobility. Because of that, we used the averaged runtimes of SbDS as baseline and limited computation times of Cplex and GP accordingly.

Figure 10 shows the number of valid path settings found by all approaches having configured averaged runtimes of SbDS. It turns out that Cplex needs more time to reach competitive results compared to SbDS, SbDS', and GP. Contrary to this, GP finds approximately as much path settings compared to SbDS'. The initial population was mainly generated with k-shortest paths [33] since random generated paths take more time to find non over-utilized solutions.

These plots verify that with increasing inital over-utilization till 15% and more, proposed path finding techniques reach their limits if runtimes are adapted to SbDS although plenty capacity conform path settings exists. This is illustrated with gray bars in the background. These results are obtained with Cplex running over several days. All in all, Cplex more often finds paths for all flows that do not over-utilize the MANET if computation time is not restricted. If runtimes matter, SbDS' and GP, return the most capacity conform path settings followed by SbDS.

B. Link Quality Scores and Robustness

This section firstly compares the obtained link quality scores of all approaches and secondly discusses if and how these scores influence the life times of found path settings. Also, we



(a) Capacity conform path settings (b) Socres of valid path settings

Fig. 8. Found capacity conform paths settings and link quality scores during increasing initial over-utilization

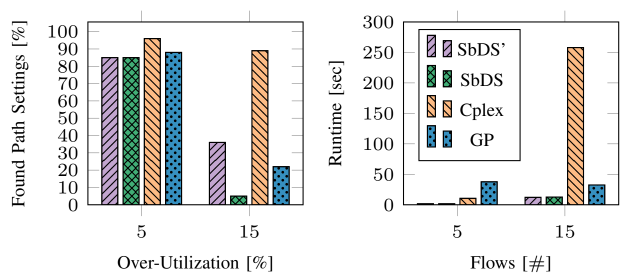


Fig. 9. Found capacity conform path settings and runtimes of all approaches

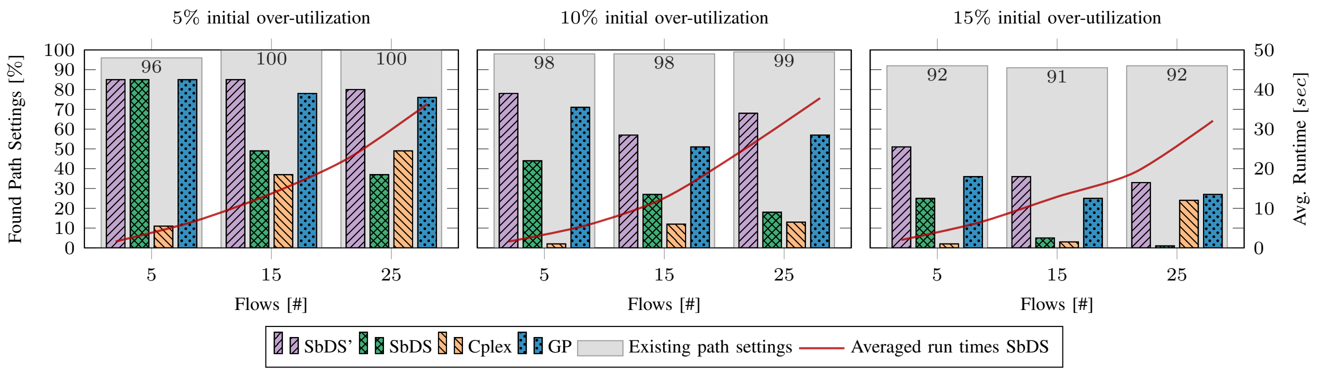


Fig. 10. Found non over-utilizing path settings with limited runtimes. Maximum runtimes of Cplex and GP adapted to the runtime of SbDS. Red lines represents computation runtimes of SbDS corresponding to right y-axises.

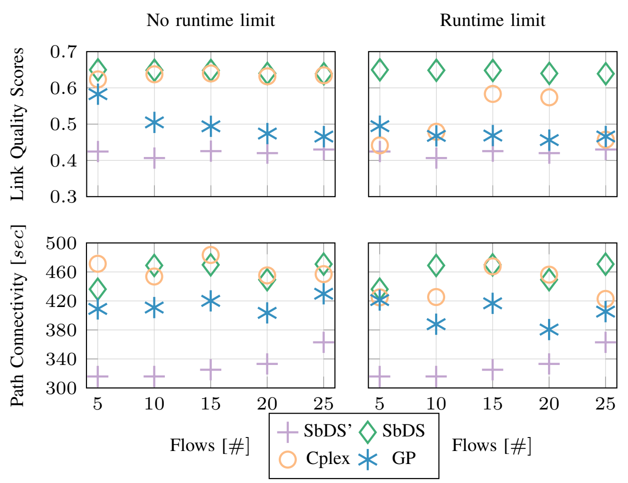


Fig. 11. Link Quality Scores without and with time restriction

investigate, if path lengths influence the robustness. We started multiple runs of all path finding techniques to obtain at least 25 scenarios where all approaches found capacity conform path settings. It is important to compare all techniques based on the same scenarios (equal topologies, source-target pairs, and initial over-utilization) in order to generate profound results regarding which technique optimizes better towards the desired goal. Hence, we compare link quality scores, path life times, and paths length of all path finding techniques. Again, Cplex and GP are both evaluated with averaged SbDS runtimes as well as with maximum 8 minutes runtime. Figure 11, shows link quality scores and path life times of all approaches with different number of flows and an initial over-utilization of 5%. Link quality scores on the upper left and right plot of this figure are inverted to present them in more readable form. It turns out that Cplex with no runtime restrictions and SbDS compute best scores followed by GP whereas those deteriorate with increasing number of flows. SbDS' computes the worst scores since our link quality metric is not applied.

The upper right plot of Figure 11 shows obtained scores where Cplex and GP limit their runtimes to the averaged runtime of SbDS, The restricted runtimes force Cplex to abort the computation and to return the current solution if available. This results in worse link quality scores especially with 5, 10, and 25 configured flows. GP again is configured to generate the Initial Population with almost only k-shortest paths [33] which results in less diversity. Such limitation combined with the time constraint shortens the link qualities slightly.

The next step is to figure out if higher link quality scores actually increase the life times of found paths. This evaluation measures the times active links of paths stay connected until transmitting or receiving nodes get out of each others coverage range. In particular, we averaged the connection life times of all active links of all paths of all common runs. Both lower plots of Figure 11 show connection live times from 5 to 25 flows each again configured with an over-utilization of 5%.

The lower left plot shows life times with no runtime restrictions. Results prove that better link qualities result in longer path connectivity. GP unexpectedly shows worse but still acceptable results compared to Cplex and SbDS although computed link qualities of GP are less promising. Comparing connectivity life times with restricted runtimes, shown with the right lower plot verifies that the worsen link qualities decrease the connection life-times of GP and Cplex slightly.

Minimum path lengths are not indicators for robust connections which is illustrated in Table I. Also, maximizing link robustness and decreasing the number of transmitting links is a promising compromise to compute capacity conform and robust paths. This especially turns out when comparing link quality scores with path lengths of SbDS and Cplex configured with runtime restrictions. Both approaches archive longest path

TABLE I
PATH LENGTHS OF APPROACHES USING 5% OVER-UTILIZATION

Flows	**Uniform runtime**		**No runtime limit**		**Runtime limit**	
[#]	*SbDS'*	*SbDS*	*Cplex*	*GP*	*Cplex*	*GP*
5	27.9	34.1	32.0	38.18	34.36	32.91
10	46.35	58.23	54.64	66.0	64.47	57.17
15	62.05	71.88	70.0	80.41	74.76	77.59
20	82.43	91.125	88.19	103.31	91.93	97.31
25	97.58	107.79	105.3	129.94	121	127.11

connectivity life times.

VI. Conclusion

In this paper, we proposed several centralized path finding techniques for MANETs and answered the question if and to which extent these approaches find non over-utilizing routes for several flows where distributed routing fails. MANETs are easily over-utilized since nodes are moving continuously and flows are routed based on incomplete topology knowledge. Previous research proposed an algorithm capable to deliver an up-to-date topology snapshot for routing purposes to an arbitrary node. Based on that, we designed and developed a tailored simulation environment to evaluate our path finding approaches regarding found path settings and computation runtimes. Since mobility is the principal reason for link breaks, computed paths must be long-living regarding connectivity. Hence, we also designed and presented a link quality metric to identify long-living links during path computations.

We used the modeler IBM ILOG Cplex Optimizer (Cplex), adapted heuristics like Genetic Programming (GP), and also developed our own approach, named Sequence-based Deployment Strategy (SbDS) as path finding techniques to provide diversity.

Our findings show, that if runtime is not a concern, Cplex outperforms all other approaches in terms of found solutions and computed link qualities. Surprisingly, GP and SbDS find more path settings than Cplex when runtimes are restricted. Also, results show that our developed link quality metric notably extends the connection lifetimes of paths.

VII. Acknowledgments

This work was partially supported by the EU project CONCORDIA (Grant agreement No. 830927).

References

[1] C.-K. Toh, *Wireless ATM and Ad-Hoc Networks: Protocols and Architectures*. USA: Kluwer Academic Publishers, 1996.

[2] J. Wakefield. Hong kong protesters using bluetooth bridgefy app. Accessed on: October, 3rd 2021.

[3] D. Barker, "Bringing mobile ad hoc networks to the battlefield using cots open standards," *Extreme Engineering Solutions*, pp. 22–26, 2013.

[4] Y. Jahir, M. Atiquzzaman, H. Refai, A. Paranjothi, and P. Lopresti, "Routing protocols and architecture for disaster area network: A survey," *Ad Hoc Networks*, vol. 82, 08 2018.

[5] M. S. Marwick, C. M. Kramer, and E. J. Laprade, "Analysis of soldier radio waveform performance in operational test," in *Defense Technical Information Archive*, 2015, pp. 1–92.

[6] K. Streit, E. Viehmann, F. Steuber, and G. D. Rodosek, "Improving routing with up-to-date and full topology knowledge in manets," in *2020 Military Communications and Information Systems Conference, MilCIS 2020*. IEEE, 2020, pp. 1–8.

[7] K. Streit and G. D. Rodosek, "Cetup: Controller-equipped topology update process for tactical ad-hoc networks," in *PE-WASUN '20, Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, Alicante, Spain, November 16-20, 2020*. ACM, 2020, pp. 57–66.

[8] L. R. Ford and D. R. Fulkerson, "A suggested computation for maximal multi-commodity network flows," *Management Science*, vol. 50, no. 12_supplement, pp. 97–101, 2004.

[9] J. L. Kennington, "A survey of linear cost multicommodity network flows," *Operations Research*, vol. 26, no. 2, pp. 209–236, 1978.

[10] A. A. Assad, "Multicommodity network flows—a survey," *Networks - An International Journal*, vol. 8, no. 1, pp. 37–91, 1978.

[11] K. Salimifard and S. Bigharaz, "The multicommodity network flow problem: state of the art classification, applications, and solution methods," *Operational Research*, pp. 1–47, 04 2020.

[12] S. Henri and P. Thiran, "Optimal number of paths with multipath routing in hybrid networks," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2018, pp. 1–7.

[13] J. E. Mitchell, *Integer programming: branch and cut algorithmsInteger Programming: Branch and Cut Algorithms*. Boston, MA: Springer US, 2009, pp. 1643–1650.

[14] A. V. Bemten, J. W. Guck, C. M. Machuca, and W. Kellerer, "Bounded dijkstra (BD): search space reduction for expediting shortest path subroutines," *CoRR*, vol. abs/1903.00436, 2019.

[15] T. Chondrogiannis, P. Bouros, J. Gamper, and U. Leser, "Alternative routing: K-shortest paths with limited overlap," ser. SIGSPATIAL '15. New York, NY, USA: Association for Computing Machinery, 2015.

[16] V. Kolar and N. Abu-Ghazaleh, "A multi-commodity flow approach for globally aware routing in multi-hop wireless networks," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, 2006, pp. 10 pp.–317.

[17] N. Katayama, "A combined fast greedy heuristic for the capacitated multicommodity network design problem," *Journal of the Operational Research Society*, vol. 70, no. 11, pp. 1983–1996, 2019.

[18] J. M. Kleinberg and M. X. Goemans, "Approximation algorithms for disjoint paths problems," Ph.D. dissertation, USA, 1996, aAI0597431.

[19] K. Walkowiak, "New algorithms for the unsplittable flow problem," in *Proceedings of the 2006 International Conference on Computational Science and Its Applications - Volume Part II*, ser. ICCSA'06. Berlin, Heidelberg: Springer-Verlag, 2006, p. 1101–1110.

[20] E. Akin and T. Korkmaz, "Rate-based dynamic shortest path algorithm for efficiently routing multiple flows in sdn," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[21] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[22] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Complex Adaptive Systems, 1992, second edition.

[23] L. Spector, W. B. Langdon, U.-M. O'Reilly, and P. J. Angeline, Eds., *Advances in Genetic Programming: Volume 3*. Cambridge, MA, USA: MIT Press, 1999.

[24] W. Banzhaf, P. Nordin, R. Keller, and F. Francone, *Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications*. Morgan Kaufmann, 01 1998.

[25] H. Masri, S. Krichen, and A. Guitouni, "Metaheuristics for solving the biobjective single-path multicommodity communication flow problem," *International Transactions in Operational Research*, vol. 26, 02 2017.

[26] M. H. Alavidoost, M. Tarimoradi, and M. H. F. Zarandi, "Bi-objective mixed-integer nonlinear programming for multi-commodity tri-echelon supply chain networks," *Journal of Intelligent Manufacturing*, vol. 29, no. 4, pp. 809–826, April 2018.

[27] J. Garcia-Luna-Aceves, "Carrier-sense multiple access with collision avoidance and detection," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '17. New York, NY, USA: ACM, 2017, pp. 53–61.

[28] E. Desurvire, *Gaussian channel and Shannon–Hartley theorem*. Cambridge University Press, 2009, p. 264–282.

[29] K. Streit, N. Rodday, F. Steuber, C. Schmitt, and G. D. Rodosek, "Wireless sdn for highly utilized manets," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 226–234.

[30] K. Streit, C. Schmitt, and C. Giannelli, "Sdn-based regulated flow routing in manets," in *IEEE International Conference on Smart Computing, SMARTCOMP 2020, Bologna, Italy, September 14-17, 2020*. IEEE, 2020, pp. 73–80.

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

[32] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriyawong, "A comparative study of random waypoint and gauss-markov mobility models in the performance evaluation of manet," in *2006 International Symposium on Communications and Information Technologies*, 2006, pp. 894–899.

[33] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.