Policy-based routing for Flying Adhoc Networks

Florian Wiedner
Department of Informatics
Technical University of Munich
Garching by Munich, Germany
wiedner@net.in.tum.de

Jonas Andre
Department of Informatics
Technical University of Munich
Garching by Munich, Germany
andre@net.in.tum.de

Paulo Mendes
Airbus
Munich, Germany
paulo.mendes@airbus.com

Georg Carle
Department of Informatics
Technical University of Munich
Garching by Munich, Germany
carle@net.in.tum.de

ABSTRACT

Strategic and profile-based communication in wireless adhoc networks is complex since it is important to distinguish between different traffic types and their routes to ensure the required performance over a dynamic network.

In order to face this challenge, we developed a new concept of a policy-based routing protocol for flying adhoc networks, having the Babel routing protocol as a starting point. Policies are identified based on the Type-of-Service field, allowing the augmented Babel protocol to select routes based on the properties of different types of traffic. Our approach differentiates route costs per traffic profile to consider different requirements in the route selection process. The evaluation shows that applying our approach improves traffic-specific performance; namely, tail-latency in video traffic can be significantly improved by one order of magnitude.

CCS CONCEPTS

• **Networks → Mobile ad hoc networks**; *Network mobility*; *Short-range networks*.

KEYWORDS

Network Mobility, Routing, FANET, Policies

ACM Reference Format:
Florian Wiedner, Jonas Andre, Paulo Mendes, and Georg Carle. 2022. Policy-based routing for Flying Adhoc Networks. In *Eighth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet '22), July 1, 2022, Portland, OR, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3539493.3539578

1 INTRODUCTION

Strategic wireless adhoc communication between flying vehicles is difficult. In this challenging scenario, routing is an important task to select an optimal path for different types of traffic; taking into account metrics such as the battery level, sending power, or the current positions improves the routing performance.

However, current approaches for routing in Flying-Adhoc Networks (FANETs) have one limitation in common: they do not differentiate between traffic types, such as traffic with low-latency or high-throughput requirements. This limitation means that traffic of different types uses the same route since the routing approaches do not consider traffic-specific requirements. This limitation leads to a low performance due to the impact that different flows have on each other over the same path. Moreover, these routing approaches cannot exploit potential better paths for different types of traffic, focusing on selecting the shortest path in terms of hop count.

To face this challenge, we come up with a profile-based routing concept based on the routing protocol Babel [10]. Babel is a loop-avoiding, distance-vector protocol suitable for wired, wireless, and adhoc networks, an essential property for seamless integration of FANETs into terrestrial networks. Moreover, Babel already has properties that can be further exploited to define a policy-based adhoc routing protocol, such as the capability to take link delays into account for the route selection. The proposed policy-based Babel routing protocol is evaluated using prototype implementation and Mininet WiFi. Our evaluation focuses on the influence that awareness of different traffic profiles has on end-to-end-delay.

The paper is structured as follows: Section 2 presents background and related work analysis. On Section 3 we describe the concept of a policy-based routing approach based on Babel, while Section 4 shows the different strategies to implement traffic policies. The concept of policy-based routing in FANETs is evaluated in Section 5, and in Section 6 we provide the information needed to reproduce our results. Section 7 concludes the paper.

2 BACKGROUND AND RELATED WORK

This section contains background information related to three emerging areas, routing in FANETs, Quality of Service (QoS)-aware routing, and policies in routing.

2.1 Routing in FANETs

Efficient end-to-end communication is a challenge in FANETs. Tareque et al. [20] provide an analysis of the challenges of devising networking frameworks in FANETs while comparing them to traditional adhoc networks. For instance, in their work Tareque et al. raise

---

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.
DroNet '22, July 1, 2022, Portland, OR, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9405-5/22/07...$15.00
https://doi.org/10.1145/3539493.3539578

that high mobility and different node types in terms of energy and lifetime expectations make routing in FANETs a challenge. In this scenario, routing protocols should be capable of updating routing tables very fast while reducing the impact on other aspects of the FANET system, such as energy consumption. Moreover, due to the constrained nature of FANETs in terms of resources, routing protocols must be able to select distinct paths to distribute traffic with different quality requirements.

Furthermore, many routing protocols are optimized for wireless adhoc networks that may be used in FANETs. These routing protocols include topology-based, position-based, and swarm-based routing approaches [18]. These protocols are highly specialized and require additional information such as the overall topology, the position of nodes, or the distance between them [18]. These approaches are either proactive or reactive, aiming to predict or react to fast-changing conditions in the network [17]. To reduce resource consumption, we selected to focus on distance-vector protocols, which hold only a local view of the network and reduce the amount of data sent to every node as well as with the changing nature of FANETs, a partly-proactive routing approach is required to react on changes as early as possible. Moreover, topology-based routing protocols use the information already contained in each routing process, source, and destination and do not require additional sensor-based information such as positional routing protocols, reducing the resource requirements for routing.

One example for a topology-based distance vector routing protocol is Babel [10], which is available as an open-source solution [9]. Babel can support end-to-end communications in wireless and wired, ad-hoc, and fixed networks, ensuring a seamless integration of FANETs into terrestrial networks. The Babel concept is based on the Destination-Sequenced Distance Vector routing (DSDV) and the Ad hoc On-Demand Distance Vector routing (AODV), which find routes to a destination on demand [13]. The combination of a proactive approach (DSDV) and a reactive one (AODV) within the same routing protocol brings advantages to Babel, such as the capability of discarding route announcements if their acceptance may lead to routing loops. This loop-avoiding characteristic of Babel, as well as the fact that Babel is a local-view topology-based routing protocol, not requiring nodes to gather information about the overall network, makes Babel suitable for fast-changing networks [10]. However, Babel cannot select distinct routes taking into account the QoS requirements of traffic.

2.2 QoS-aware routing

QoS-aware routing is a reasonably researched area, especially in what concerns Ethernet or broadband networks [11], where finding an optimal path that guarantees QoS-specific limits is NP-hard without further redesign of input parameters. Chen et al. [7] highlight the focus of current research on QoS in adhoc networks, as the rapidly changing nature of these networks makes it harder to guarantee QoS end-to-end. As early as 2005, Akkaya and Younis [1] described the need for additional efforts to guarantee QoS-specific requirements in adhoc networks. However, none of them have addressed the integration of QoS awareness into existing routing algorithms suitable for FANETs. Moreover, QoS-aware routing in vehicular adhoc networks is available and provides similar metrics for evaluation, such as end-to-end delay [3]. These previous works lead to the need to integrate QoS-aware or rule-based routing approaches to enable features such as Internet sharing using unmanned aerial vehicles (UAVs).

bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	type = sub-TLV-Policy								length = 1								Policy							
bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Figure 1: Sub-TLV for policy-specific routing in Babel adapted after [8].

2.3 Routing policies

Not for every traffic requirement are QoS classes available. Further, in the area of routing, the need for policies is rising to support dividing connections into slices for clients based on their needs [5]. However, money is not always the fundamental reason for setting policies; different operators may classify traffic differently, not available as QoS-classes, and require specific customizable policies. QoS policies are a special case with well-defined requirements [15].

3 POLICY-BASED BABEL ROUTING PROTOCOL

As mentioned before, Babel [10] has shown that it outperforms other FANET routing protocols in areas such as throughput [13], but it misses policy-awareness in terms of managing different types of traffic. As it already outperforms other routing protocols and a reference implementation is available as an open-source variant, we have used Babel as a baseline for developing a policy-based routing protocol. Therefore, we propose to extend the Babel protocol aiming to: i) take into account delay when computing link costs per-class on all links [16]; and ii) use Type of Service (ToS)-specific routing in Babel [8]. The latter enables the new Babel routing protocol to distinguish routes based on the ToS-field value in the Internet Protocol (IP) header.

Basic concept. The basic principle for extending Babel passes is creating new so-called sub-type length values (sub-TLVs) for routing messages such as Update messages; Babel uses sub-TLVs to implement optional extensions, which are ignored when unavailable in the current system. Each sub-TLV is assigned a type number during standardization, with a type range for experimental use [10].

We use the sub-TLV shown in Figure 1, adapted from [8], which includes the ToS-number for identifying the corresponding policy next to the length and type value.

In all data structures that store connection-related or path-related information, the policy is a primary key field that identifies routes by destination prefix length, destination prefix, and policy (ToS number) to compute the optimal path for all available triples. Babel uses information stored in these data structures to feed the customized Bellmann-Ford algorithm to calculate an optimal path to forward a specific type of traffic towards each destination.

However, the concept ToS specific routing in Babel [8] faces three main problems: the route for all policies to the same destination is always the same with the same link cost; the policy values are never defined and thus neither distributed nor used; the delay metric is only used on tunneled interfaces, though it is helpful for wireless interfaces as well. To overcome these limitations, we extended the Babel protocol, making usage of the new sub-TLVs and adapting data structures.

Delay-based metric. Delay is an essential component in defining paths for latency-based policies since traffic such as video or sensor data require low-delay connections to receive data in real-time.

The delay-based metric extension for Babel [16] derived a method for using timestamps contained in Babel messages sent in regular intervals, e.g., Hello and IHU messages, to calculate the Round-Trip-Time (RTT) per connection in tunneled links. This method adds additional cost to the link cost when the delay is higher than a predefined minimum delay value. The delay cost is used per link and is added to the path cost announced to neighbors. For other connection types, such as Wireless connections, the delay is smaller and will never reach the proposed maximum RTT of 120 ms, as defined in the reference implementation *babeld* [9].

In the proposed policy-based Babel protocol, we use the delay-based metric for all interfaces and adjust the delay parameter to be usable within the indicated connections.

Distribution of policies. In the proposed policy-based Babel protocol, each node advertises its local addresses with a pre-defined list of policies. This way, we ensure that all policies supported on a node are advertised, and that traffic can reach the destination with policy-aware path selection. This selection is a recursive process since neighboring nodes will announce the new path option, including the policy to their adjacent neighbours [10].

A ToS-value is added to each prefix announcement to identify the policy associated with the announced traffic type. However, this ToS-value may be interpreted differently on different nodes but should generally be distributed with the software or by a central instance in the network, such as a base station. A particular case is policies defined by the ToS definition, which can be directly used as a policy scheme. To distinguish link costs between policies and enable different traffic to follow the path according to their requirements, we need to adapt our concept further.

Our prototype automatically announces any locally added interfaces using a fixed list of policies to ensure comparability and repeatability without needing an additional protocol for distributing policies. This ensures that all destinations known in the network are reachable with any policy.

Differentiating link-costs between policies. In Babel, a cost is calculated per link for direct neighbors [10]. The cost only depends on the direct connection as only a local view is available.

Our approach uses the policy as a new argument for the local cost calculation to distinguish between traffic types crossing the same link. After the computation of link costs based on policy arguments, routes are announced to neighboring nodes with a tuple of destination address, policy, and cost for the announced paths. The cost of each announced path is the sum of all traversed links, including the local one.

	Video	VoIP	Critical	Sensor	BE
Min	1/2x	1/2x	1/4x	1/4x	1x
Max	1x	1x	1/2x	1/2x	1x
Penalty	1x	2x	1x	2x	1x

Table 1: delay penalty configuration.

The cost function per policy can be based on different metrics, e.g., delay for low-latency policies or bandwidth for high-throughput policies. The concrete cost function is policy-dependent, and the next section describes examples of such strategies.

After computation, for each path cost, only one next-hop is added to the Forwarding Information Base (FIB) for each tuple destination address and ToS-value. Compared to the original Babel version, the computational overhead for policy-based Babel routing is increasing linearly with the number of policies.

4 POLICY-BASED ROUTING STRATEGIES

Different strategies are possible to implement routing policies. Policies can be based on a different metric set or on specific algorithms, such as using machine learning to estimate link costs [19].

The most straightforward strategy is calculating link costs using metrics such as delay, link loss, or bandwidth to define each policy. The downside of this strategy is that no further reaction to the current situation without changing the values of the used metrics is possible.

To evaluate the proposed policy-based Babel routing protocol, we choose a static strategy based on traffic classes to ensure repeatability of results. Five classes are considered: critical control, Voice-over-IP (VoIP), video, sensor data, and best-effort (BE) traffic. The selected strategy uses metric weights based on delay costs to define each traffic class.

Our default weights used for evaluation are shown in Table 1. According to [9], the following basic values are used, represented as x in Table 1: 10 ms minimum latency, 120 ms maximum latency, and a maximum penalty of 96 with a linear function between the minimum and the maximum.

5 EVALUATION

A fine-grained analysis is required to evaluate policy-based routing in dynamic environments like FANETs. This section evaluates the proposed policy-based Babel protocol and uses policy strategies in an emulation setup.

5.1 Setup

Evaluations of solutions based on real FANETs are complex, as devices are expensive and further effort is needed to fulfill legal requirements. For this reason, we decided to build our evaluation setup based on the emulation of Unmanned Aerial Vehicle (UAV) networks while using realistic mobility models.

Mobility models. Bujari et al. [4] describe various possible mobility models and tools for FANETs. From those, we choose the Bonnmotion [2] tool to generate our evaluation scenarios due to

Type	Rate	Proto.	Dir.	Seconds
Video	3 to 8	UDP	Unidir	5 s to 120 s
VoIP	0.5 to 2	All	Bidir	5 s to 120 s
Critical	0.1 to 0.8	TCP	Bidir	1 s to 20 s
Sensor	0.01 to 20	UDP	Unidir	1 s to 20 s
Mixed	0.01 to 20	All	Mixed	5 s to 120 s

Table 2: Traffic generation borders per class with rates in Mbit s^{-1}.

the available command-line interface and support of different mobility models typical for FANETs.

Bonnmotion creates mobility scenarios based on predefined mobility models and their configuration parameters. We selected the Nomadic Community Mobility Model (Nomadic) [6] as one mobility scenario for our experiments. In this model, a group of UAVs follows a reference point within a predefined radius. In addition, we use the reference point group mobility model, in which UAVs head for a reference point and group around it when reached [14]. We use the two mobility models as typical UAV behavior in different missions.

FANET emulation. For emulation of UAV networks, we use a customized version of Mininet able to emulate wireless and mobile networks - Mininet WiFi [12]. The mac80211 HWSim module simulates WiFi connections and interfaces in the emulation environment. This module offers benefits such as realistic WiFi behavior, broadcast domains, and distance-dependent behavior [12]. One of two UAVs emulates a high-latency node to simulate node type variations in the emulated setup.

Traffic generation. We use iperf3 processes to generate traffic based on randomly generated connection pairs based on predefined boundaries per traffic class shown in Table 2. We use both popular transport layer protocols, User Datagram Protocol (UDP) and Transmission Control Protocol (TCP).

Traffic is randomly generated up to a predefined limit of concurrent connections. We use iperf3 and TShark for collecting measurement data, such as the amount of routing traffic.

5.2 Protocol evaluation

To evaluate our approach, we use a mobility scenario based on the Nomadic mobility model and compare the results between the original Babel version—*babeld*—and the proposed policy-based Babel protocol. We evaluate their performance in a scenario with 24 moving nodes, shown in Figure 2 and Figure 3 on a selected time representing the current movement status at that time, showing that the nodes move based on the mobility model Nomadic during the time of the measurement and for comparison are always the same time-based positions used when evaluating the same scenario.

In Figure 4 the number of routing table entries over time is shown with each color representing an individual node. The same color represents the same node for both the original and the proposed concept versions. The number of routing table entries increases with six traffic profiles, but not linearly, as we never reach more than 72 entries compared to the 14 entries in *babeld*. Figure 4 shows that

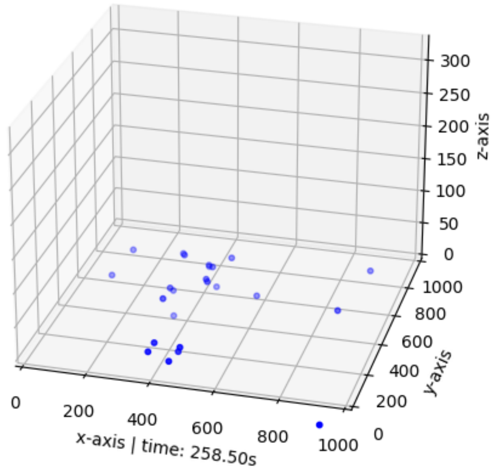

Figure 2: Model of the primary used mobility scenario at 258.5 s with each blue dot representing an UAV.

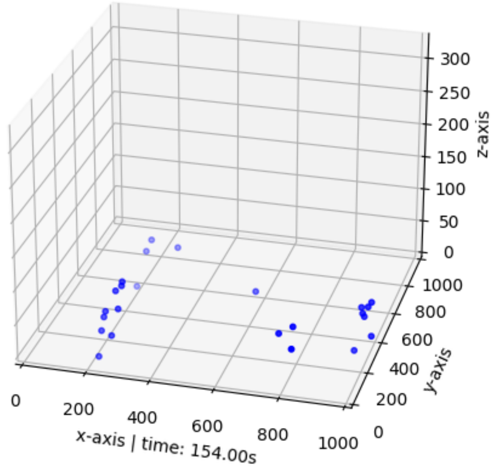

Figure 3: Model of the primary used mobility scenario at 154 s with each blue dot representing an UAV.

with this amount of routing table entries, even when not increasing linear, the performance of the underlying system in routing table lookups is becoming significantly more important.

In addition, we divide the tail latency shown in Figure 5 into profiles to analyze their impact. Overall, the delay is lower with our approach for almost all traffic classes, but in particular, the delay is improved for Video traffic holding 1 s at 99.99 %. Only the delay suffered by sensor traffic is worse when using the policy-based Babel protocol, suggesting that this configuration can be improved, as sensor traffic is, according to Table 1 a low-latency class. The reason is that the maximum latency threshold for low-latency traffic was always reached, resulting in the same penalty for all links. Moreover, critical traffic is better in the original version than our proposed version, resulting from the case that for the critical traffic, the infinity value for costs is reached at some points,

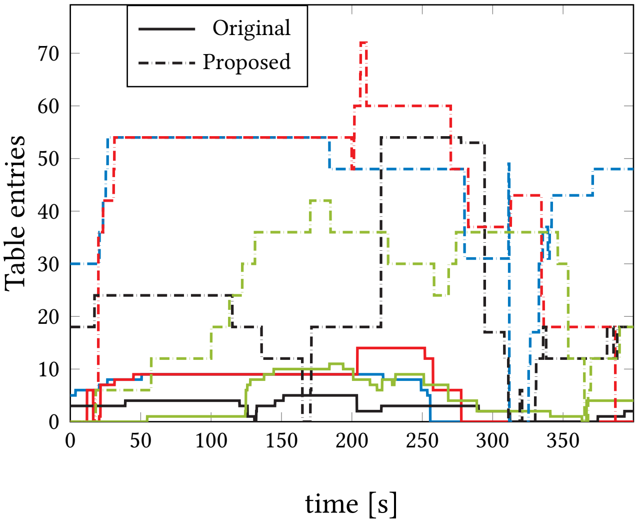

Figure 4: Routing table entries on selected nodes.

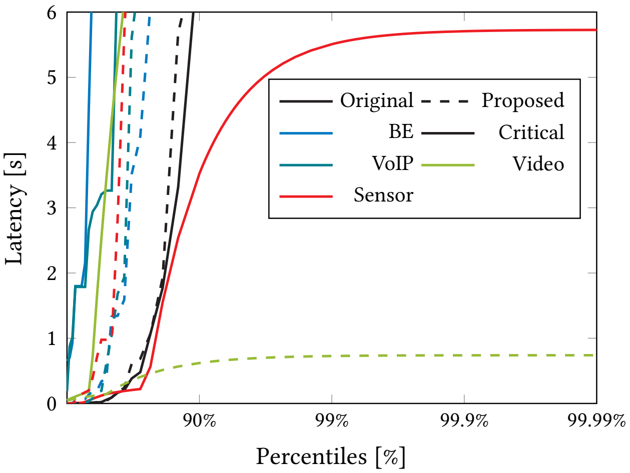

Figure 5: HDR-Plot for delay.

resulting in worse routing behavior than the original algorithm. This is another reason leading to the idea of improving the proposed weights for the different traffic classes.

We further analyzed the routing traffic generated by the policy-based Babel protocol, which presents an average increase from 2.2 kbit s^{-1} to 4.7 kbit s^{-1} when compared with the original Babel protocol. The usage of six policies justifies this increase, which means that overhead of around twice the messaging is generated with six times the amount of policies. As more messages in regular intervals need to be sent, the energy consumption will be presumable higher, but we have not further performed experiments on the energy consumption of individual nodes. However, although the amount of transmitted traffic is six times higher than when using just one type of traffic, the exchanged routing information increased at a lower rate—around two times.

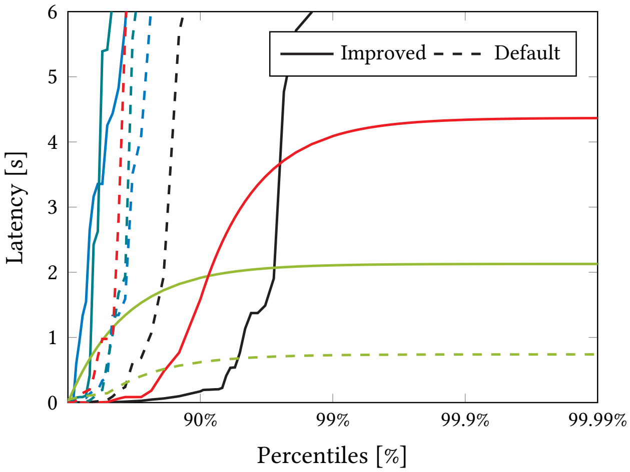

Figure 6: Delay results for delay configuration according to Table 3 with colors representing same traffic classes as in Figure 5.

5.3 Strategy evaluations

In this section, we aim to analyze the impact that policy strategies have on the performance of a policy-based routing protocol. We evaluate several configurations of the proposed cost function. Results are, however, not presented in detail due to limited space.

	Video	VoIP	Critical	Sensor	BE
Min	1/2x	1/2x	1/4x	1/4x	1x
Max	1x	1x	1/2x	1/2x	1x
Penalty	1x	1x + 10	1x	1x + 10	1x

Table 3: settings with best performance.

Table 3 shows the delay configuration set for the variant with the best performance. We modified that the expected transmission count, used for wireless links in Babel, is disabled for low-latency traffic, and the maximum delay cost is 106 instead of 2x96 ensuring to mark feasible links. With this, we analyze the impact of changing policies on the results.

Figure 6 shows the delays for the configuration in Table 3, evaluated in the same mobility scenario to enable comparison, with an focus on worst-case delays of the individual traffic classes. With worst-case delays, we focus on the higher end of the measured delays. The video traffic reaches 2 s delay in the worst-case, which is twice as before, but especially the critical and sensor data show significant improvements. Only VoIP and BE traffic worsens in tail-latency, which shows that the selection needs to be careful as other traffic worsens. The critical and sensor data are, according to Table 3 the significant low-latency policies, which are significantly improved. To take away from this result is carefully selecting weights and metrics based on the requirements of the profiles. Further optimization per policy is required.

Furthermore, we analyzed the performance in a mobility scenario built based on the reference point group mobility model. The scenario results show that using this mobility model, the configuration as in Table 3 is better than the original since the delay measured is generally higher and therefore exceeds the limit for the maximum penalty mostly in different scenarios as well.

5.4 Limitations and Future Work

Our evaluation is limited in the area of evaluated cost functions as only static policies from compile-time were used. This limitation reduces the applicability in connected scenarios, where the base station should be able to adapt policies on-the-fly. Furthermore, due to external limitations, the analysis was only performed with a setup in Mininet WiFi and could not be evaluated in real scenarios in a flying environment.

As a result of these limitations, the presented concept represents a way to use a routing protocol available for wireless and wired links in UAV networks but lacks further in-depth evaluations. In future work, we will further analyze integrating different metrics and algorithms in the cost function to optimize the potential of policies in FANETs. Further, we aim to analyze the routing traffic impact on the potential of the FANET itself. Similarly, have we not analyzed the computational complexity of our proposed changes such that this is left for future work.

6 REPRODUCIBILITY

To be able to reproduce our results, have we published the prototype implementation of our proposed policy-based Babel routing concept at *https://github.com/tumi8/babeld*. It is possible to verify the results with this prototype and Mininet WiFi.

7 CONCLUSION

Managing strategic wireless adhoc communications between moving vehicles while fulfilling the requirements of different types of traffic is difficult since different flows have different requirements, such as low latency or high throughput, resulting in a challenging task to route the traffic. To face this challenge, we propose a new policy-based routing protocol for FANETs, based on the Babel protocol, together with the usage of Type-of-Service tags to distinguish between different policies while using delay as a metric to implement policies in each link.

Using the proposed concept offers advantages over the current Babel protocol and other routing algorithms presented in related work, namely: custom profiles can be defined; both wireless and wired networks are supported; additional metrics and cost estimation algorithms can be added on a per-profile basis.

Evaluation results show significant performance improvements for different policies depending on the settings emulated in Mininet WiFi, such as reducing the tail-latency for video traffic to 1 s with less than linear scaling of routing table entries and routing traffic towards the number of policies. Results also show that each policy needs to select cost functions carefully.

We aim to analyze a method to allow policies to be adapted on-the-fly as future work. Moreover, we plan to analyze the convergence time, which is a challenge in FANETs, since it is unknown what the optimal path is at any moment in time. Furthermore, we aim to analyze the impact of additional metrics and algorithms such as a machine-learning approach and analyze our concept in a real environment. Finally, measuring the impact on energy consumption or required computational power on UAVs is an experiment for future work.

REFERENCES

[1] Kemal Akkaya and Mohamed Younis. 2005. Energy and QoS Aware Routing in Wireless Sensor Networks. *Cluster Computing* 8, 2 (2005), 179–188.

[2] Nils Aschenbruck, Raphael Ernst, Elmar Gerhards-Padilla, and Matthias Schwamborn. 2010. BonnMotion: a mobility scenario generation and analysis tool. In *3rd International Conference on Simulation Tools and Techniques*, L. Felipe Perrone, Giovanni Stea, Jason Liu, Adelinde M. Uhrmacher, and Manuel Villén-Altamirano (Eds.). Malaga, Spain, 1–10. https://doi.org/10.4108/ICST.SIMUTOOLS2010.8684

[3] Fatima Belamri, Samra Boulfekhar, and Djamil Aïssani. 2021. A survey on QoS routing protocols in Vehicular Ad Hoc Network (VANET). *Telecommun. Syst.* 78, 1 (2021), 117–153. https://doi.org/10.1007/s11235-021-00797-8

[4] Armir Bujari, Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni, Claudio Enrico Palazzi, and Daniele Ronzani. 2017. Flying ad-hoc network application scenarios and mobility models. *International Journal of Distributed Sensor Networks* 13, 10 (2017). https://doi.org/10.1177/1550147717738192

[5] Matthew Caesar and Jennifer Rexford. 2005. BGP routing policies in ISP networks. *IEEE Network* 19, 6 (2005), 5–11. https://doi.org/10.1109/MNET.2005.1541715

[6] Tracy Camp, Jeff Boleng, and Vanessa Davies. 2002. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing* 2, 5 (2002), 483–502. https://doi.org/10.1002/wcm.72

[7] Zheng Chen, Wenli Zhou, Shuo Wu, and Li Cheng. 2020. An Adaptive on-Demand Multipath Routing Protocol With QoS Support for High-Speed MANET. *IEEE Access* 8 (2020), 44760–44773. https://doi.org/10.1109/ACCESS.2020.2978582

[8] Gwendoline Chouasne and Juliusz Chroboczek. 2017. TOS-Specific Routing in Babel. https://datatracker.ietf.org/doc/html/draft-chouasne-babel-tos-specific.

[9] Juliusz Chroboczek. 2022. Ubuntu Manpage: Babeld - ad-hoc network routing daemon. http://manpages.ubuntu.com/manpages/bionic/man8/babeld.8.html

[10] Juliusz Chroboczek and David Schinazi. 2021. The Babel Routing Protocol. *RFC* 8966 (2021), 1–54. https://doi.org/10.17487/RFC8966

[11] Eric S. Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. 1998. A Framework for QoS-based Routing in the Internet. *RFC* 2386 (1998), 1–37. https://doi.org/10.17487/RFC2386

[12] Ramon dos Reis Fontes, Samira Afzal, Samuel Henrique Bucke Brito, Mateus A. S. Santos, and Christian Esteve Rothenberg. 2015. Mininet-WiFi: Emulating software-defined wireless networks. In *11th International Conference on Network and Service Management*, Mauro Tortonesi, Jürgen Schönwälder, Edmundo Roberto Mauro Madeira, Corinna Schmitt, and Joan Serrat (Eds.). Barcelona, Spain, 384–389. https://doi.org/10.1109/CNSM.2015.7367387

[13] Antonio Guillen-Perez, Ana-Maria Montoya, Juan-Carlos Sanchez-Aarnoutse, and Maria-Dolores Cano. 2021. A comparative performance evaluation of routing protocols for flying Ad-Hoc networks in real conditions. *Applied Sciences* 11, 10 (2021), 4363.

[14] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. 1999. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. 53–60.

[15] Paul E. Jones, Subha Dhesikan, Cullen Jennings, and Dan Druta. 2021. Differentiated Services Code Point (DSCP) Packet Markings for WebRTC QoS. *RFC* 8837 (2021), 1–9. https://doi.org/10.17487/RFC8837

[16] Baptiste Jonglez and Juliusz Chroboczek. 2019. Delay-based Metric Extension for the Babel Routing Protocol. https://datatracker.ietf.org/doc/html/draft-ietf-babel-rtt-extension-00.

[17] Muhammad Asghar Khan, Inam Ullah Khan, Alamgir Safi, and Ijaz Mansoor Quershi. 2018. Dynamic routing in flying ad-hoc networks using topology-based routing protocols. *Drones* 2, 3 (2018), 27.

[18] Omar Sami Oubbati, Abderrahmane Lakas, Fen Zhou, Mesut Günes, and Mohamed Bachir Yagoubi. 2017. A survey on position-based routing protocols for Flying Ad hoc Networks (FANETs). *Vehicular Communications* 10 (2017), 29–56. https://doi.org/10.1016/j.vehcom.2017.10.003

[19] Kuldeep Singh and Jaspreet Kaur. 2017. Machine learning based link cost estimation for routing optimization in wireless sensor networks. *Advances in Wireless and Mobile Communications* 10 (2017), 39–49.

[20] Md. Hasan Tareque, Md. Shohrab Hossain, and Mohammed Atiquzzaman. 2015. On the routing in Flying Ad Hoc Networks. In *2015 Federated Conference on Computer Science and Information Systems*, Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki (Eds.). Lódz, Poland, 1–9. https://doi.org/10.15439/2015F002