Fast and Seamless Handover in Software Defined Vehicular Networks

Nadia Mouawad
Lastre Laboratory
Lebanese University, Lebanon
Li-Parad Laboratory
University of Versailles, France
nadia.mouawad@uvsq.fr

Rola Naja
Lastre Laboratory
Lebanese University, Lebanon
Li-Parad Laboratory
University of Versailles, France
rola.naja@ul.edu.lb

Samir Tohme
Li-Parad Laboratory
University of Versailles, France
samir.tohme@uvsq.fr

Abstract

Tele-operated driving is a transition phase to the deployment of fully autonomous cars. Vehicles are supposed to communicate with a remote control station in order to exchange surrounding videos and location information. This communication is achieved through a vehicular network, and it requires low latency and high reliability. To this end, the handover management should be studied carefully. This paper devises a mobility management scheme for vehicular networks based on the Software Defined Networking (SDN) paradigm. The proposed scheme provides handover anticipation in order to guarantee a low-latency and a seamless handover. In addition, it proposes a strategy that aims at reducing control signaling overhead and network congestion. Performance analysis show that our approach reduces handover delays and losses.
Keywords: Software Defined Networking, Vehicular Networks, Seamless handover, Fast handover and Tele-operated Driving.

I. Introduction

Tele-operated driving is a bridge between human driving and fully autonomous driving. It consists of remotely controlling a vehicle from a tele-operation station. To this end, the vehicle should exchange with the tele-operation station information about its environment including video feed, location on a map and current weather conditions. The communication is accomplished through a vehicular network.

In this context, with the high spatial mobility of vehicles, exchanged messages with the tele-operation station should be delivered seamlessly and in a tight time window. Therefore, the mobility management should be efficiently studied.

SDN [1] is an efficient technique that attempts to manage the handover operation in vehicular networks. In fact, SDN is an emerging paradigm that aims at decoupling control and data planes.

Open Networking Foundation (ONF) [2] defines the SDN architecture by three main planes: control plane that consists of SDN controllers, data plane that comprises SDN switches and application plane that includes networks applications. In addition, Openflow protocol [3] is standardized for the control and data planes communications. We give an overview about this protocol in the following paragraph.

Openflow describes the interaction between controllers and switches. When a switch receives a packet, it parses its header to match it against installed flow table entries (rules). If there is a matching between the packet header and table entry match fields, the switch executes an action such as routing the packet to an output port. If several flow tables entries are found, the entry with highest priority is selected.
Whenever a switch receives a packet that cannot be matched with any installed rule, it sends the packet to the controller that calculates the path from source to destination and installs flow rules in the calculated path switches.

Several studies ([4], [5]) have tackled the SDN based mobility management. In [6], a SDN-based architecture is proposed, where a logically distributed control plane is devised for mobility management. In [7], SDN based handover consists of deleting flow rules from the previous route after Layer-2 handover (change of link attachment) establishment, and installing new rules on the route to the new user location.

In [8], the Mobile Node (MN) is assigned an identifier IP address that does not change while moving in addition to a locator IP address (MN's first hop switch IP address). The handover process is handled by forwarding the Correspondent Node (CN) packets to the new MN location.

In [9], the handover procedure proceeds as follows. After identifying the candidate target Access Points (AP), SDN mobility application calculates a route from the serving AP to each candidate AP. In addition, packets are duplicated on all calculated routes in order to allow

The work was funded by the Lebanese University and the AUF ' Projet de cooperation scientifique interuniversitaire' (PCSI).

MN to continue receiving packets. The proposed scheme is a hard handover anticipation procedure.

The aforementioned works deploy a hard handover which may be inappropriate for tele-operated driving use case. In addition, several studies lack of a routing optimization strategy that should reduce the control signaling overhead and network congestion.

In this paper, we orient our efforts to devise a seamless and fast handover solution. Our contributions are listed as follows:

1) We propose a SDN based mobility management scheme that deploys handover anticipation.
2) We present an algorithm that optimizes the routing procedure aiming at reducing handover signaling overhead and network congestion.
3) We introduce a Mobility Anchor (MA) where packets are duplicated in order to achieve soft handover for MN using two interfaces simultaneously.
4) We consider a tele-operated driving use case in order to evaluate the proposed handover scheme.

The paper is structured as follows: in section II, we present the adopted software defined network topology. In section III, we describe the proposed algorithm. In section IV, we evaluate our proposed scheme and shed the light on its benefits. Finally, we conclude the paper in section V.

II. Network topology

We study the handover solution of a tele-operated driving use case where an autonomous vehicle is supposed to communicate with a tele-operation station. The vehicle is traveling on a highway covered by Road Side Units (RSUs). The message exchange is achieved through a software defined vehicular network composed of the following elements (depicted in figure 1):

- **Road Side Units:** RSUs are openflow enabled data plane elements that provide IEEE 802.11p [10] connectivity.
- **Openflow switches:** openflow enabled data plane elements that connect the RSUs to the tele-operated station.
- **SDN Controllers:** SDN controllers (SDNC) manage the RSUs and switches located in its domain. Each controller contains a Binding Cache (BC) that maps between MN identifiers and locations. SDNC presents a core module that computes an optimal path between all network switches. This is realized according to a routing algorithm that achieves a trade-off between the shortest and less link loaded path. The optimal path calculation is achieved periodically in normal network state. This computation is exceptionally triggered in case of remarkable link load variation. In addition, SDNC can manage the mobility procedures.
- **Global SDN controller (G-SDNC):** G-SDNC has a global view of the network.

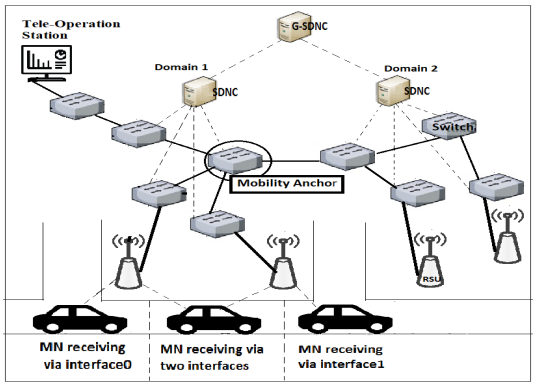

Figure 1. Network topology

III. Handover Solution

Our solution addresses the MN registration, MN-CN communication and intra/inter domain handover.

A. MN registration and communication with CN

Each MN presents two physical interfaces: interface0 and interface1. The MN is assigned a stable Identifier Address (IdAd). In addition, the controller assigns a Location Address (LocAd) to the connected interface.

When a MN connects to an RSU, the latter sends a message to SDNC including MN's identifiers (MN's connected interface MAC address, IdAd) which will be saved in the controller cache and binded with the assigned LocAd.

The CN communicates with the MN by sending packets to IdAd. SDNC installs flow rules on the optimal path switches between CN and MN by using openflow address rewriting in order to redirect the packets to MN LocAd.

B. Intra-domain handover solution

Whenever MN detects signal degradation, it triggers a Layer-2 handover that is performed through the following steps:

1) When a signal degradation is detected, the RSU sends an indication to SDNC.
2) SDNC identifies the target RSU taking into account vehicle direction, network load and security constraints.
3) After target RSU identification, SDNC derives an optimal routing strategy: A fork switch between the previous and new path (CN switch to target RSU path) is calculated, this switch is named the Mobility Anchor (MA).
4) The MN connects to target RSU via interface1 while keeping interface0 connected to previous RSU.

5) SDNC installs specific flow rules on the MA in order to duplicate packets to previous and target RSU. In addition, SDNC installs flow rules on path from the MA to the target RSU.
6) The handover is executed when the MN disconnects from previous RSU. In this case, SDNC modifies flow rules in order to redirect packets to the target RSU only.

C. Target RSU identification

As mentioned earlier, when signal degradation is detected, the RSU sends an indication to the SDNC.
SDNC identifies candidate RSUs as follows. SDNC considers that some RSUs may be inappropriate, i.e, a RSU may give a falsified signal to the vehicle that is traveling in the opposite direction. To this end, SDNC will choose a set of candidate networks among the vehicle available RSUs.
In fact, each vehicle is supposed to send its destination to a Location Server (LS), such as a GPS navigator, in order to download the route from its current location to its destination. The LS periodically forwards the calculated route information to the SDNC that omits, unsuitable RSUs according to the vehicle direction.
For the remaining set of candidate RSUs, the target RSU is chosen according to its load and some security constraints.

D. Optimal routing strategy

Traditionally, routing algorithms aim at installing flow rules on the optimal (shortest and less congested) path between CN and the target RSU. However, this method may cause performance drawbacks in terms of control signaling overhead. Therefore, we extend our efforts to derive a strategy that can guarantee the following goals:

- Keeping the shortest path to the new RSU.
- Minimizing signaling overhead by reusing the maximum number of previous path switches as a part of the new calculated path. This can avoid new flow rules installation and thus can reduce time consumption and signaling load.
- Considering the traffic load as an important constraint in order to avoid network congestion and guarantee a low latency end to end communication.

To this end, we formulate an algorithm that identifies a fork switch on the previous path. We list the symbols used in this algorithm in table I. In addition, figure 2 illustrates the fork switch calculation scenario.

This algorithm takes as input the optimal path P_{prev} between cn and s_{prev}. It omits the switches s_i where $b_i \geq a$ (in order to avoid installing a number of flow rules larger than the one used in case of P_{opt}). For the remaining switches, the algorithm omits s_i where $d_i/a \geq \beta$. This step guarantees the length similarity between P_{target} and P_{opt}. For each remaining switch s_i, a value r_i

Table I
Fork switch calculation symbols

Symbol	Signification
s_{prev}	previous switch (RSU)
cn	corresponding node switch
s_{target}	target switch (RSU)
s_{fork}	fork switch
P_{prev}	optimal path between s_{prev} and cn
P_{opt}	optimal path between s_{target} and cn
P_{target}	calculated path using fork switch
s_i	switch on P_{prev}
β	similarity factor between P_{opt} and P_{target}
w_1, w_2 and w_3	weighting factors
a	optimal path length between cn and s_{target}
c_i	optimal path length between s_i and cn
b_i	optimal path length between s_i and s_{target}

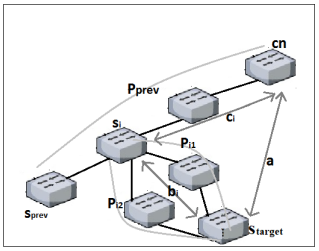

Figure 2. Fork switch calculation illustration

Algorithm 1 Routes calculation algorithm

Fork switch calculation
a=Optimal-Path-length[cn-s_{target}]
$P_{prev'}$=P_{prev}
for each $s_i \in$P$_{prev'}$ **do**
  c_i = Optimal-Path-length[s_i-cn] (Distance on P_{prev})
  b_i = Optimal-Path-length[s_i-s_{target}]
  if ($b_i \geq a$) **then**
    $P_{prev'}$.delete(s_i)
  end if
end for
for each $s_i \in P_{prev'}$ **do**
  $d_i = b_i + c_i$
  if ($d_i/a \geq \beta$) **then**
    $P_{prev'}$.delete(s_i)
  end if
end for
for each $s_i \in P_{prev'}$ **do**
  r_{1i}=c_i/d_i
  r_{2i}=b_i/d_i
  N_i : Traffic load on the path (cn to s_i to s_{target})
  $r_i = r_{1i}^{w_1} * r_{2i}^{-w_2} * N_i^{-w_3}$ where $\sum w_i = 1$
end for
s_{fork}= s_i where r_i is maximal
Fork Path calculation
P_{target} = optimal path from s_{fork} to s_{target}
$P_{target} = P_{prev}(cn$ to $s_{fork})$+P_{target}

is calculated. s_i with the highest r_i is chosen as s_{fork}. r_i calculation presents the following objectives:

1) r_i is a Multiplicative Exponent Weighting (MEW) used in the Multiple Attributes Decision Making (MADM) problems.
2) Finding the maximal value of r_i leads to find the fork switch that conserve the maximal number of switches from P_{prev} and the minimal number of newly installed flow rules on P_{target} (optimal path from s_{fork} to s_{target}). In addition, to a minimal traffic load on the chosen path.

The calculated fork switch is referred by the MA. It plays an important role in the handover procedure as explained in the next paragraph.

E. Flows duplication

The MA is an important agent in handover procedure. Flow rules are installed on this switch as follows:

- **Rule1:** rule1 is responsible of duplicating packets to the previous and target RSU. It is noteworthy that this rule presents the highest priority compared to the previously installed rules. In addition, this rule has an idle-timeout in order to be removed after the handover execution.

The MN can receive with both interfaces and the handover can be executed seamlessly as explained in the following paragraph.

F. Handover execution

When the MN disconnects from previous RSU (s_{prev}), it keeps receiving packets through interface1. The controller installs the following rule in the MA:

- **Rule2:** rule2 has the highest priority compared to the previously installed rules, responsible of redirecting packets on P_{target} only.

Since Rule2 has a higher priority than Rule1, the packets will be redirected according to Rule2 and the Rule1 will be inactive until its idle-timeout expires.

G. Inter-domain handover

The inter-domain handover occurs in two cases: 1) when the target RSU belongs to another administrative domain controlled by a target SDNC. 2) When the vehicle triggers the need of a forced inter-domain handover. We elaborate each case in the following:

1) After low signal detection and target RSU identification, SDNC finds that target RSU is not in its own domain. In this case an inter-domain handover is triggered.
2) We assume that vehicles can subscribe to the following service: When a vehicle connects to a RSU, it sends a message to the tele-operation station. This message includes a measurement about the interference level sensed by the vehicle in its domain. The tele-operation station can calculate the average spectral occupancy in each domain based on the information sent by subscribed vehicles.
   In case of a significant change in the spectral occupancy of a given domain, the tele-operation station sends a message to vehicles (only subscribed vehicles to the service can receive the message). When receiving this message, the vehicle can trigger a forced inter-domain handover by sending a message to the SDNC. This trigger is achieved on a random based operation by giving a value $p_{inter} \in [0,1]$ to each vehicle. That is if $p_{inter} = 0$ the algorithm is inactive while, whenever $p_{inter} = 1$ the algorithm is always active.

In both cases of inter-domain handover trigger, SDNC sends a message to G-SDNC that will send MN's information to the target SDNC who saves MN's information in its cache.
Regarding fork switch and MA computation, we let the routing algorithm to fall back in the optimal path where we have s_{fork} =cn and P_{target}=P_{opt}.
Packets duplication and handover execution are similar to intra-domain handover procedure.

IV. Performance evaluation

We implement our proposed scheme using mininet-wifi [11]. The controller is implemented using Ryu controller [12] platform.
For the simulation parameters, we set the link capacity to 100Mbps. And we give the algorithm variables the following values: β=1.2, w_1=0.3, w_2=0.3, w_2=0.4 and p_{inter}=0. We consider a tele-operated driving use case, where a tele-operation station exchanges video flows with an autonomous vehicle.

We adopt the network topology illustrated in figure 3, where switches s_{55}, s_{50}, s_1, s_{17}, s_{16}, s_{15}, s_{14}, s_{33}, s_{44} are considered as RSU placed on a highway. Two RSU are separated by a distance of 500 meters. In addition, due to high spatial mobility, we set each RSU communication range to 300 meters. The tele-operation station (CN) is connected to switch s_7. The MN is firstly connected to an RSU (s_{55}) in figure 3.

In the following, we evaluate the overall control traffic overhead and the handover execution time. In addition, we discuss the cost-performance tradeoff.

Overall Traffic Overhead: We consider that the MN is moving from s_{55} to s_{15} while communicating with the CN. We calculate in each case the fork switch. We compare the number of flow rules installed in each handover occurrence considering two cases: 1) installing flow rules on P_{opt}: the optimal path between the CN and the new RSU (s_{target}), 2) installing flow rules on P_{target}: the path from the MA to s_{target}. Finally, we measure the fork switch calculation time. Results are shown in table II.

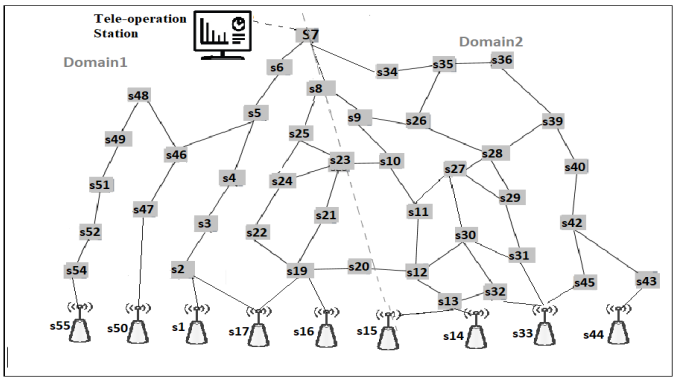

Figure 3. Network topology adopted in simulation

Table II
FORK SWITCH CALCULATION ALGORITHM EVALUATION

case	s_{prev}	s_{target}	s_{fork}	using P_{target}	using P_{opt}	calculation time(ms)
1	s_{55}	s_{50}	s_{46}	4	7	0.05
2	s_{50}	s_1	s_5	6	8	0.07
3	s_1	s_{17}	s_2	3	8	0.05
4	s_{17}	s_{16}	s_7	8	8	0.09
5	s_{16}	s_{15}	s_8	7	9	0.07

As we can see in table II, according to the adopted topology, the fork node varies from one handover occurrence to another. In some cases the algorithm falls back to the optimal path between *cn* and target RSU. However, thanks to the proposed routing algorithm, the overall number of installed flow rules is reduced.
In addition, the fork switch algorithm execution time has an average of 0.07ms, which is adequate for the tele-operated driving use case.

Handover execution time: We consider that the MN is moving from s_{55} to s_{50} with a speed of 100 km/h. We note that high spatial mobility should be tackled since it is considered as an unfavorable case for the handover procedure. At the same time, this spatial mobility limits the number of handover requests arrival and as a result it reduces the handover requests waiting time.

In this environment, we measure the overall handover time. Results, shown in table III, indicate that the intra-domain handover is achieved in 10.075 ms, and the inter-domain handover is accomplished in 17.5 ms. Compared with the MN sojourn time in the overlapping zone, and considering the low handover request rate due to high mobility, obtained results prove that our algorithm guarantees a fast and soft handover.

Table III
HANDOVER PROCEDURE EXECUTION TIME

Symbol	Meaning	value (ms)
t_{intra}	Time for intra domain handover signaling	2
t_{target}	Time to calculate target RSU	2
t_{inst}	Time to install a flow rule	1.5
t_{MA}	Time to calculate Mobility Anchor	0.075
t_{inter}	Time for inter-domain handover signaling	3
$T_{intra-total}$	Total intra-domain handover time= t_{intra}+ t_{target}+ $4t_{inst}$+ t_{MA}	10.075
$T_{inter-total}$	Total inter-domain handover time= t_{inter}+ t_{target}+ $7t_{inst}$+ t_{intra}	17.5

Table IV
PACKET LOSS RATIO (%)

D(m) / Speed (Km/h)	50	80	100
50	15	25	35
100	0	0	0
200	0	0	0

Table V
PACKET DUPLICATION COST (%)

D(m) / Speed (Km/h)	50	80	100
50	5	4	3
100	15	12	10
200	30	25	20

Performance-Cost tradeoff analysis: Packet duplication (PD) is an essential method in order to achieve seamless handover. However, utilizing PD comes with a certain price: an excessive use of the radio resources and traffic overhead. In fact, packets are duplicated in the overlapping area between cells. Therefore, a cost performance tradeoff should be realized based on the variation of the distance D of this area. This is explained as follows: in order to reduce channel allocation time and traffic overhead caused by packets duplication, the overlapping area distance D should be reduced. At the same time, reducing this distance may cause packet loss since the MN will not have sufficient time in order to terminate the handover operation before the disconnection from the previous RSU. Therefore, the choice of D is critical. Hence, we measure the packet loss ratio and the duplication cost ($\frac{Duplicated\ \ packets}{Total\ \ packets\ \ sent}$) in terms of D while varying the MN speed. Results are shown in tables IV and V.

As we can see, small values of D lead to packet loss in high speed scenarios. In addition, large overlapping distances result in a high cost. Therefore, in order to achieve the performance cost trade-off we choose an overlapping distance equal to 100 m.

In order to evaluate our algorithm, we study the following network performance parameters: Packet loss ratio and Average Round Trip Time (RTT) between MN and CN. Moreover, we compare our work with the literature study in [7] called SDN-Mobility.

1) Packet loss ratio in terms of MN speed: In this scenario, we measure the packet loss ratio in terms of MN speed and compare the results with SDN-mobility. Figure 4 depicts the obtained results.
As we can see in this figure, SDN-mobility gives a high packet loss ratio due to the lack of handover anticipation.

However, packets loss ratio presents a null value with our proposed algorithm. This is attributable to the seamless handover and PD achieved at the MA..

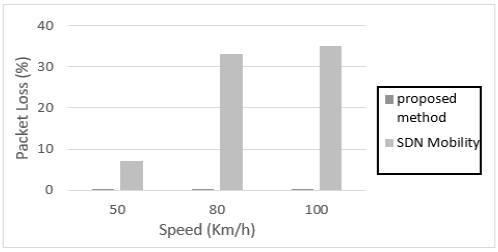

Figure 4. Packet loss ratio

2) Round Trip Delay (RTT): In the following, we measure the RTT between the MN and CN in 3 scenarios: 1) in case of SDN-Mobility, 2) in case of the proposed scheme without considering network congestion in the fork calculation, 3) in case of the proposed scheme with the network congestion avoidance.
Figure 5 shows that our method reduces RTT due to the consideration of the traffic load in the calculation of the fork switch.

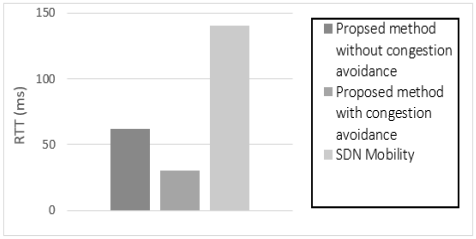

Figure 5. Average RTT

V. Conclusion

In this paper, we presented a handover algorithm for Software Defined vehicular networks. Our proposed scheme proceeds by the Layer-2 handover anticipation. In addition, it presents a flow rules placement strategy aiming at minimizing control messages exchange and network congestion. Moreover, the proposed scheme considers inter-domain handover procedure.
Experimental results showed that our approach can manage handover procedure with a null packet loss ratio and a significantly reduced handover delays.
In future works, we will evaluate performance parameters for various values of p_{inter}.

References

[1] "Sdn architecture overview," *Open Networking Foundation*, December 2013.
[2] https://www.opennetworking.org/, "Open networking foundation," 2013.
[3] "Openflow switch specification," *Version1.5.0*, vol. 1, no. 0, 2015.
[4] S. Jeon, C. Guimarães, and R. L. Aguiar, "Sdn-based mobile networking for cellular operators," in *proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*, 2014, pp. 13–18.
[5] M. Idri, "Mobility management based sdn-ipv6 routing header," in *proceedings of the Fourth International Conference on Software Defined Systems (SDS)*, 2017, pp. 150–155.
[6] K. S. Atwal, A. Guleria, and M. Bassiouni, "Sdn-based mobility management and qos support for vehicular ad-hoc networks," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 659–664.
[7] K. Tantayakul, R. Dhaou, and B. Paillassa, "Impact of sdn on mobility management," in *proceedings of IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, pp. 260–265.
[8] Y. Wang, J. Bi, and K. Zhang, "Design and implementation of a software-defined mobility architecture for ip networks," *Mobile Networks and Applications*, vol. 20, no. 1, pp. 40–52, 2015.
[9] C. Chen, Y. T. Lin, L. H. Yen, M. C. Chan, and C. C. Tseng, "Mobility management for low-latency handover in sdn-based enterprise networks," in *proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2016, pp. 1–6.
[10] I. . W. Group *et al.*, "Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments," *IEEE Std*, vol. 802, no. 11, 2010.
[11] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *proceedings of the 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 384–389.
[12] https://osrg.github.io/ryu/, "Ryu sdn framework."