UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

IULISLOI ZACARIAS

Employing Concepts of the SDN Paradigm to Support Last-Mile Military Tactical Edge Networks

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Edison Pignaton de Freitas

Porto Alegre
August 2018

CIP — CATALOGING-IN-PUBLICATION

Zacarias, Iulisloi

Employing Concepts of the SDN Paradigm to Support Last-Mile Military Tactical Edge Networks / Iulisloi Zacarias. – Porto Alegre: PPGC da UFRGS, 2018.

75 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2018. Advisor: Edison Pignaton de Freitas.

1. Software-Defined Networking. 2. Drones. 3. UAVs. 4. Battlefield Networks. I. Freitas, Edison Pignaton de. II. Título.

“Try not to become a man of success,
but rather try to become a man of value.”
— Albert Einstein

AGRADECIMENTOS

Agradeço à Deus por guiar o meu caminho e me dar forças e saúde para concluir esta etapa da minha vida.

Agradeço em especial à minha família pelo apoio às minhas escolhas. Ao meu pai, Sr. Loivir F. Zacarias, e à minha mãe, Dna. Giesla K. L. Zacarias, que são grandes exemplos e sempre me incentivaram, desde os primeiros passos da minha vida. Igualmente ao meu irmão, Eliakim Zacarias, que sei que sempre está torcendo por mim. Meu agradecimento especial com muito carinho e amor à minha namorada Karina Rosa de Deus, pela compreensão, incentivo, paciência e companheirismo. Conhecer você neste momento da minha vida me fez muito bem e jamais esquecerei tudo o que tem feito por mim, me apoiando nesta jornada.

Gostaria de agradecer também ao meu orientador, Prof. Edison Pignaton de Freitas, pelos conselhos, incentivo, companheirismo nos trabalhos e publicações pelo conhecimento que fez questão de me passar neste tempo e pelas oportunidades que me proporcionou. Cresci muito tanto nestes dois anos e meio tanto na área acadêmica como pessoal.

Meu agradecimento aos técnicos e aos docentes do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS), principalmente aos professores das disciplinas que cursei. Cabe aqui destacar meu agradecimento ao Prof. Luciano P. Gaspary que muito nos ajudou nos processos de elaboração de artigos e publicações. Exprimo aqui meus agradecimentos ao projeto FUTEBOL, pela oportunidade em trabalhar com esta excelente equipe. Agradeço à Universidade Federal de Santa Maria (UFSM), em especial à Direção do Campus de Frederico Westphalen (RS), por terem autorizado minha licença temporária para a realização deste mestrado.

Estendo meus agradecimentos ao grupo de Redes de Computadores e à todos os colegas que de alguma forma contribuíram para esta realização, principalmente aos colegas dos Lab. 210 e Lab. 212 onde passei boa parte do meu tempo. Peço desculpas por não citar nomes mas a lista ficaria muito extensa. Meu agradecimento especial ao Marcelo (Marotta) e ao Gustavo Araújo (Bob) pela ajuda nas tarefas do mestrado, pela amizade e pelos momentos de descontração que foram muito importantes para mim.

Por fim, agradeço à todos que de alguma forma, direta ou indiretamente, contribuíram para meu crescimento e para minha formação.

Muito obrigado à todos vocês!

ABSTRACT

The future battlefield tends to be populated by a plethora of "intelligent things". In some ways, this is already a reality, but in future battlefields, the number of deployed things should be orders of magnitude higher. Networked communication is essential to take real advantage of the deployed devices on the battlefield, and to transform the data collected by them into information valuable for the human warfighters. Support for human decision making and even a level of autonomy, allowing devices to coordinate and interact with each other to execute their activities in a collaborative way require continuous communication. Challenges regarding communication will arise from the high dynamics of the environment. The network adaption and management should occur autonomously, and it should reflect upper-level decisions. The large scale of the network connecting high-level echelons, troops on the field, and sensors of many types, beside the lack of communication standards turn the integration of the devices more challenging. In such a heterogeneous environment, many protocols and communication technologies coexist. This way, battlefield networks is an element of paramount importance in modern military operations. Additionally, a change of paradigm regarding levels of autonomy and cooperation between humans and machines is in course and the concept of network-centric warfare is a no way back trend. Although new studies have been carried out in this area, most of these concern higher-level strategic networks, with abundant resources. Thus, these studies fail to take into account the "last-mile Tactical Edge Network (TEN) level," which comprises resource constrained communication devices carried by troopers, sensor nodes deployed on the field or small unmanned aerial vehicles. In an attempt to fill this gap, this work proposes an architecture combining concepts from software-defined networking (SDN) paradigm and the delay-tolerant approach to support applications in the last-mile TEN. First, the use of SDN in dynamic scenarios regarding node positioning is evaluated through a surveillance application using video streaming and Quality of Experience (QoE) measures are captured on the video player. We also explore the election of nodes to act as SDN Controllers in the TEN environment. The experiments use emulator for SDN with support to wireless networks. Further investigation is required, for example, considering security requirements, however the results are promising and demonstrate the applicability of this architecture in the TEN network scenario.

Keywords: Software-Defined Networking. Drones. UAVs. Battlefield Networks.

Empregando Conceitos de Redes Definidas Por Software para Apoio à Redes Táticas Militares de Última Milha

RESUMO

Em um futuro próximo, "dispositivos inteligentes" serão massivamente empregados em campos de batalha. Essa já é uma realidade, porém, o número de dispositivos utilizados em campos de batalha tende a aumentar em ordens de magnitude. As redes de comunicação de dados serão essenciais para transmitir os dados que esses dispositivos coletam e transformá-los em informações valiosas utilizadas como suporte à atuação humana. O suporte à tomada de decisão, ou mesmo níveis de autonomia, permitindo que estes dispositivos coordenem outros dispositivos, exigem comunicação contínua. Desafios relacionados à comunicação surgirão devido à dinamicidade do ambiente. A configuração da rede deve refletir decisões superiores automaticamente. A grande escala das redes conectando os altos escalões, tropas, veículos e sensores, aliada à falta de padronização dos dispositivos, tornará a integração destes desafiadora. Em um ambiente tão heterogêneo, muitos protocolos e tecnologias coexistirão. As redes de campo de batalha são um elemento de suma importância nas operações militares modernas e conceito de guerra centrada em rede é uma tendência sem volta e influencia desde os altos escalões até o controle de tropas. Embora estudos tenham sido realizados nessa área, a maioria deles aborda redes estratégicas de alto nível e portanto não levam em conta as "redes táticas de última milha" (TEN), que compreendem dispositivos de comunicação com recursos limitados, como sensores ou ainda pequenos veículos aéreos não tripulados. Em uma tentativa de preencher esta lacuna, esse trabalho propõe uma arquitetura que combina conceitos dos paradigmas de redes definidas por software (SDN) juntamente com redes tolerantes à atraso/disrupçoes (DTN), para aplicação em redes táticas de última milha. O uso de SDN em cenários com nodos móveis é avaliado considerando uma aplicação de vigilância que utiliza *streaming* de vídeo e medidas de Qualidade de Experiência (QoE) de usuário são coletadas. Com base nos resultados obtidos, uma aplicação em conjunto dos conceitos de SDN e DTN é proposta, além disso abordamos a escolha do nodo que atuará como controlador SDN na rede. Os experimentos foram executados utilizando um emulador de redes. Apesar de pesquisas adicionais serem necessárias – considerado requisitos de segurança, por exemplo – os resultados foram promissores e demonstram a aplicabilidade destes conceitos no cenários das TENs.

Palavras-chave: Redes Definidas por Software, Drones, VANTS, Redes Militares.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

API	*Application Programming Interface*
AppQoE	*Application Quality of Experience*
ARP	*Address Resolution Protocol*
BN	*Battlefield Network*
C2	*Command and Control*
COTS	*Commercial-Off-The-Shelf*
DASH	*Dynamic Adaptive Streaming over HTTP*
DIL	*Disconnected, Intermittent and Limited Network*
DTN	*Delay/Disruption Tolerant Network*
HD	*High Definition*
HTML5	*Hypertext Markup Language, version 5*
HTTP	*Hypertext Transfer Protocol*
IoT	*Internet of Things*
IP	*Internet Protocol*
LLDP	*Link Layer Discovery Protocol*
MAC	*Medium Access Control*
MOS	*Mean Opinion Score*
MST	*Minimum Spanning Tree*
QoE	*Quality of Experience*
QoS	*Quality of Service*
RN	*Relay Node*
RTT	*Round Trip Time*
RVT	*Remote Video Terminal*
SAR	*Synthetic Aperture Radar*

SDK *Software Development Kit*

SDN *Software-Defined Networking*

SOA *Services-Oriented Architecture*

SPOF *Single Point of Failure*

TCP *Transmission Control Protocol*

TEN *Tactical Edge Network*

UAV *Unmanned Aerial Vehicles*

UDP *User Datagram Protocol*

WAVE *Wireless Access in Vehicular Environments*

WSN *Wireless Sensor Network*

CONTENTS

1 INTRODUCTION

Due to advances in the manufacturing process, component miniaturization and continuous cost reductions, Unmanned Aerial Vehicles (UAVs) became a widely used technology nowadays. With the UAVs readily available to the public, new applications have emerged, in many fields, such as precision agriculture, remote sensing, weather monitoring, support for communication networks in disaster scenarios, wireless coverage expansion, and even delivery of goods (BEKMEZCI; SAHINGOZ; TEMEL, 2013). In addition to the use in civilian applications, UAVs are still widely employed in military applications, with early uses dating from 25 years ago. Typical examples of military applications are border surveillance, ground reconnaissance, and offensive missions (GUPTA; JAIN; VASZKUN, 2016). Although military missions often employ large UAV platforms capable of carrying big payloads and sophisticated sensors, small UAVs systems (DALAMAGKIDIS, 2015) are very effective when used in Tactical Edge Networks (TENs).

Contemporary military operations are ruled by interconnected units, from the strategic level – *i.e.,* the higher-level decision centers – to the tactical-operational level represented by the units on the battlefield. Battlefield Networks (BN) establish the connection among those different units (or nodes), using a variety of communication technologies, from short-range wireless links to satellites links. In the tactical-operational level, the involved networks are referred as Tactical Edge Networks (or TENs) (TORTONESI et al., 2013), which vary depending on the capabilities of nodes composing them. On one end, there are networks of high-end nodes, with little to none resource constraints, and no energy consumption restrictions (NOBRE et al., 2016). On the other end, networks of resource-constrained nodes, also called low-end nodes, represent the last-mile BN, composed of wireless sensor nodes deployed on the field, radios carried by special forces units, small UAVs for image acquisition, among others (ORFANUS; FREITAS; ELIASSEN, 2016). These different networks support a variety of applications, ranging from elastic ones, such as file and text message transfer, to non-elastic real-time applications, such as video streaming (KUROSE; ROSS, 2013). TENs are affected by different problems and have to meet different requirements depending how elastic the applications are. To tackle those problems, the overall communication system that composes the battlefield scenario can employ different technologies, such as Software-Defined Networking (SDN) paradigm and Delay/Disruption Tolerant Networks (DTN).

SDN is a promising technology, providing flexibility to network management by

separating the network infrastructure into distinct planes (WICKBOLDT et al., 2015). A control plane, which implements the network control, and a data plane, which just forwards packet according to control directives. The data plane can be programmed by the control plane to meet particular application requirements. Possible changes range from Quality of Service (QoS) optimizations to the deployment of new protocols and policy enforcements in a running network. The network can be programmed to support legacy or Commercial-Off-The-Shelf (COTS) applications without considerable changes in the application behavior. The network control is logically centralized in the SDN controller, which provides a programmatic interface to the network, allowing external applications to deal with the network as a single system (KREUTZ et al., 2015) using an Application Programming Interface (API). The API provided by the SDN controller allows easy integration with high-level non-networks systems. Those systems can deploy security, media or vendor specific features using the controller API, regardless of network protocols. Due to this abstraction, the SDN model can be deployed in heterogeneous networks (NUNES et al., 2014) and represents a suitable model to address the needs of the military networks. The use of SDN in the BN scenario was explored by Nobre et al. (2016) aiming to improve communication among devices in the dynamic and heterogeneous military environment. The authors propose a high-level approach to apply SDN concepts to military networks in which the communication nodes are resource-rich, such as battleships and airplanes, using links provided by satellites, among other.

DTN is a network architecture approach that aims to address the lack of continuous connectivity in dynamic networks with extended periods of link unavailability (FALL; FARRELL, 2008). TEN employs Relay Nodes (RN) to route the data from source to destination to acquire the coverage of larger areas (including rugged terrains and harsh environments) where they are usually deployed. However, RNs can be unavailable by energy depletion, technical failures or due to attacks by enemy forces (such as radio frequency jamming). Also, due to the high mobility of some nodes (UAVs and ground vehicles, for example) data routes should be often recalculated, and sometimes an end-to-end connection between the source and the destination nodes does not exist, or exists for a very limited interval of time. Long link outages lead to loss of connectivity, transmissions timeouts, and routing failures. DTN tackles the link outages problem by temporarily storing the data in RNs. Such data is then forwarded to the destination when opportunistic connections emerge. As presented by Amin et al. (2015), DTN is a technology largely explored in military networks, due to the need to augment the coverage of net-

works communication by using aerial high capacity backbones composed of manned and unmanned aircrafts. The links among aircrafts exhibit frequent intervals of outage compared to ground or satellite networks. However, the DTN approach presented in their work also consider networks at the strategic level, which counts on resource-rich nodes. In this sense, this approach does not cover the last-mile TEN, especially considering troops on the terrain.

Video transmission is widely employed in surveillance and reconnaissance missions, and it is often carried out in the environment where Tactical Networks are used (NIGHTINGALE et al., 2016). Additionally, UAVs can be used in TENs to support surveillance and reconnaissance missions. The wireless transmissions of those media must comply with QoS requirements to assure acceptable video quality from the user perspective. If those requirements are not met, the video streaming may only contribute to network jamming (JOSEPH; BORST; REIMAN, 2016). Moreover, connections used by video streaming applications must comply with rigorous requirements concerning latency, latency variation (jitter) and throughput. Although the requirements for a useful video stream transmission in military applications differ from entertainment applications (e.g., YouTube, Netflix), they can be measured in a similar way (ARMY, 2009). Measurements used to evaluate video stream playback perceived by the final user can be classified into objective and subjective metrics (JULURI; TAMARAPALLI; MEDHI, 2016). Objective measurements can be collected in the user video player (e.g., playback start time, the number of video interruptions, duration of interruptions). Subjective metrics, like Mean Opinion Score (MOS), are based on the user feedback, collecting measurements directly from the users. Researchers have demonstrated that it is possible to infer the user subjective evaluation (e.g., MOS) based on the observed objective measurements (LIU et al., 2016; ITU-T, 2016).

In entertainment applications, more attention is given to the overall user satisfaction (e.g., MOS), whereas in military applications some objective measurements are critical. For example, a long time freeze on the video playback may omit important events and lead to erroneous military decisions, whereas several short video freezes result in a low MOS evaluation by an ordinary user, but the provided information in this frozen video frame may be sufficient for specific military purposes. Observing these aspects, video quality assessments can be used as a feedback to adjust network settings and implement policy enforcement to improve or preserve video quality according to the application requirements (NAM et al., 2014; ABUTEIR; FLADENMULLER; FOURMAUX, 2016).

It is a major challenge to keep video quality requirements at acceptable levels, given the high complexity of managing the resources of the network in a highly dynamic environment, such as a multi UAV-based surveillance setup. Most of the solutions proposed regarding UAV surveillance networks (commercial or military) make use of ad-hoc solutions based on wireless mesh networks (ORFANUS; FREITAS; ELIASSEN, 2016). The use of conventional network solutions (non-SDN) in these applications makes the device reconfiguration process difficult, or dependent on proprietary solutions. Moreover, proprietary solutions add extra difficulty to interface new system to already deployed applications (TORTONESI et al., 2013). The configuration of new equipment joining the network also becomes more costly using conventional ad-hoc solutions, thus hindering network scalability.

Observing the gap in the literature regarding proposals for last-mile TENs issues, and taking into account the resource-constrained devices used by troops in the field, this work proposes the joint exploration of SDN and DTN concepts to address the needs of these tactical-operational networks. The proposed approach benefits from the programmability offered by SDN and the ability of DTN to handle link outages. These features are used in the context of network-centric military operations in the field, and primarily to fulfill the hard-to-meet requirements of low-end nodes. The main contribution of this work is to define an architectural solution to support the last-mile TEN by adapting and combining SDN and DTN technology, bringing flexibility and agility to this rugged environment.

Initially, to support the employment of the SDN paradigm in TENs, experiments regarding a surveillance application in a military scenario, making use of using video transmission were performed. A military scenario in which UAVs are collecting video to secure the presence of troops in the battled field was emulated using the Mininet-Wifi tool. Additionally, we introduced modifications in a video player application to collect data about the occurrence of events that affect the video quality perceived by the final user – such video stalls and the playout initialization time – and based on the collected values, a prediction of the MOS was performed. The results were promising and show that SDN can be employed in such scenarios. Additional research was performed to realize the needs of military communication regarding networks with mobile nodes. Based on the research, an architecture combining features of the SDN paradigm and the DTN paradigm were proposed. The architecture employs a leader election algorithm to select nodes to act as the SDN Controller in the network, thus overcoming problems like

Single Point of Failure (SPOF); thus a leader election algorithm used in Wireless Sensor Networks (WSNs) were adapted to our architecture. Although further investigation is required regarding aspects as security, the obtained results show that the proposed approach can handle the challenging UAV-based military surveillance operational scenario.

This work is organized as follows: Chapter 2 presents and discusses the related work. Chapter 3 introduces the application scenario for UAV-based military surveillance used in this work. Chapter 4 presents the proposed SDN approach to support video streaming in dynamic networks composed of mobile nodes. Chapter 4 describes the proposed architecture combining the SDN and the DTN approaches in last-mile TENs. Chapter 5 reports the experiments carried out and the obtained results, while Chapter 6 draws conclusions and provides directions for future work.

2 RELATED WORK

This chapter provides an overview of the related work regarding the use of SDN and DTN in the military networks, as well as concepts of QoE in video applications. The Section 2.1 explores the application of SDN and DTN concepts in military networks. Additionally, the use of these paradigms to support the integration of COTS hardware and software in the TEN environments are discussed. Section 2.2 provides an overview of QoE metrics and their use to assess the user experience on video applications. Also, the use of QoS configurations in an SDN-enabled network aiming to provide the final user a better experience is explored. Finally, the related work is discussed in Section 2.3.

2.1 Software-Defined Networking and Delay/Disruption Tolerant Networks in the Military Context

Tortonesi et al. (2013) have discussed the increasing interest in the adoption of Commercial Off the Shelf (COTS) hardware and software in military Tactical Edge Networks (TEN), which is noteworthy in UAV-based systems (TORTONESI et al., 2012). In fact, there is a growing research interest in the use of IoT equipment and COTS hardware and software in modern UAV applications. In (MOTLAGH; BAGAA; TALEB, 2017), the authors present a video surveillance application that uses facial recognition techniques to allow remote monitoring of crowds in places of interest. The use case reported in this work consists on offloading the data, through a wireless connection, to process the video in a mobile edge computing environment. While the data offloading process can deal with flexible network requirements, a live video transmission requires a very strict delay bound and high bandwidth.(BEKMEZCI; SAHINGOZ; TEMEL, 2013).

The use of standards designed for wired connections or corporate networks and the reliance on TCP connections can often lead to low performance when the COTS solutions are running in TEN, especially when highly QoS sensitive applications are taken into account(TORTONESI et al., 2013). COTS solutions are not prepared to cope with some specific characteristics of TEN, as the frequent disconnections of nodes caused by several reasons. The use of legacy systems can add extra difficulties to the scenario, since these applications were not developed to deal with intermittent links. Additionally, the UAVs need to opportunistically explore new resources in the network, switching to different services providers. As the UAV moves to an area to provide network connectivity to a

remote network partition, or when there is a need to select links offering different QoS characteristics, new connections will appear and existing connections can be dropped.

A work from Nobre et al. (2016) highlight some of the benefits of using SDN in the military scenario. The authors propose an architecture to apply the software-defined networking paradigm to cope with the inherent properties of military networks. The co-operation of legacy networks and software-defined networks is addressed in the proposed architecture. The authors argue that battlefield networks can take advantage of the benefits of software-defined networks, such as ease of setup, flexibility in policy enforcement, flow optimization, and network adaptation. To demonstrate some of the benefits, a use case involving a real-time video application, with end-to-end delay and video quality constraints was described. The network controllers, with the help of a flow optimization application, can select the most appropriate path to forward the video stream, meeting the task requirements. Despite the consistent argumentation, the paper brings no experimental evidence supporting the proposed approach.

2.2 Quality of Experience

The QoE is the subject of an extensive survey conducted by Juluri, Tamarapalli and Medhi (2016). In the first part of the work, a tutorial overview of the existing video delivery methods are shown. In the second part, the authors offer a tutorial overview of measurement techniques of video QoE. According to the employed measurement mechanism, the QoE metrics are classified into objective or subjective metrics. As Juluri, Tamarapalli and Medhi (2016) point out, traditional network Quality of Service (QoS) metrics are not sufficient to determine the satisfaction of the users. Instead, it is necessary to collect metrics perceived by the users, allowing to determine their QoE. Objective metrics, as playback start time, the number of interruptions, and duration of interruptions can be collected by measurement tools on the video-player software. Meanwhile, subjective metrics are based on the experience reported by the user while using a video service. The most popular subjective and default metric for subjective video application assessments is the MOS. Recent investigations have shown a correlation between objective and subjective metrics. Therefore it is possible to predict the QoE of the final user based on objective metrics (SEUFERT et al., 2015; HOSSFELD et al., 2012).

Recent research work discusses video flow optimizations through networks configuration changes based on statistics collected on the video player application. Aim-

ing at providing a better QoE to the end user, network optimization techniques, policy enforcements and appropriate path selection are proposed by Nam et al. (2014). The decision-making process makes use of statistics collected directly from video players or from network measurements. The authors propose an architecture and an SDN controller that can adjust the network parameters to deliver video traffic to the final user with better QoE. An improved HTML5 player obtains the video QoE measurements on the client side. The video player collects information about the status of the player, selected video resolution, video buffer rate, and playback start time and report it to the delivery node. Buffering events and packet loss thresholds trigger network reprogramming routines, and according to the gathered information, new routing paths are selected. The authors use the Junos Space SDK from Juniper Networks to implement the prototype.

Kleinrouweler, Cabrero and Cesar (2016) explore the overall view of the network, provided by the SDN controller, to assist the Dynamic Adaptive Media Players (DASH) to select the best video resolution supported by the network. The Linux built-in traffic control mechanisms are used to apply the QoS configurations and enable concurrent video players to use the available bandwidth in a fairness way. The DASH player has been extended to report media information and buffer status to a Service Manager. The Service Manager interacts with the SDN controller that is in charge of applying the necessary changes to the network hardware using the OpenFlow (MCKEOWN et al., 2008) protocol. The work presented in (ABUTEIR; FLADENMULLER; FOURMAUX, 2016) proposes a solution to cope with network bandwidth competition among concurrent video flows in the network. The authors aim to reduce the player instability (e.g. the need for DASH players to switch between video streams with different resolutions frequently) and maximizing the fairness among different video clients on the same network. A peculiarity of these solutions presented above is the acquisition of network measurements to estimate user QoE and perform adjustments in the network behavior. As Juluri et al. (JULURI; TAMARAPALLI; MEDHI, 2016) pointed out, traditional network QoS measurements are not sufficient to determine users' satisfaction. Instead, it is necessary to collect measurements perceived by the users, allowing to determine their quality of experience (QoE).

2.3 Discussion

The present work differs from the previous ones by performing SDN within a new model to meet temporal requirements, considering extremely dynamic networks and

applications. While the studied literature focuses on video stream optimizations in local wired networks or the application of the SDN paradigm in a military network composed of high-end nodes with abundant resources, the scenario in which TENs run may differ regarding node characteristics. Furthermore, TENs are often classified as disconnected, intermittent and limited networks (DIL), and therefore the current end-to-end TCP/IP model is not well suited for these environments (AMIN et al., 2015; NIGHTINGALE et al., 2016). Table 2.1 summarizes and presents the main characteristics of the related works.

The integration of the SDN and the DTN paradigms which is the primary focus of the present work aims to fill the gap found in the present solutions, and it was designed observing the needs of military TENs. Although the proposed solution is useful for less resource intensive applications like monitoring solutions composed of small sensors which can make use of opportunistic network links to transmit their valuable data, applications that require near real-time forwarding of data poses additional challenges.

The next chapter describes the scenarios studied to design the proposal of the present work. A scenario describing a military surveillance application employing software-defined network to forward and route the data among the devices in the battlefield is explored. Following, a more generic scenario employing concepts of the DTN paradigm alongside with the SDN paradigm is described.

Table 2.1: List of related works and main characteristics

Author	SDN	DTN	Type of Nodes	Considers QoS / QoE	Enable COTS Integration	Node Mobility	Evaluation Method
(TORTONESI et al., 2013)	No	Yes	High-end and low-end nodes	-	Yes	Yes	Deployment / Emulation
(TORTONESI et al., 2012)	No	Yes	High-end and low-end nodes	-	Yes	Yes	Deployment / Real hardware
(MOTLAGH; BAGAA; TALEB, 2017)	No	No	Low-end nodes	No	Yes	Yes	Deployment / Real hardware
(BEKMEZCI; SAHINGOZ; TEMEL, 2013)	-	-	Low-end nodes	-	-	Yes	Survey only
(NOBRE et al., 2016)	Yes	No	High-end nodes	-	-	No	Proposal
(JULURI; TAMARAPALLI; MEDHI, 2016)	-	-	-	Yes	-	No	Survey only
(SEUFERT et al., 2015)	-	-	-	Yes	-	No	Survey only
(HOSSFELD et al., 2012)	-	-	-	Yes	-	No	Deployment
(NAM et al., 2014)	Yes	No	High-end nodes	Yes	-	No	Deployment
(KLEINROUWELER; CABRERO; CESAR, 2016)	Yes	No	High-end nodes	Yes	-	No	Deployment
(MCKEOWN et al., 2008)	Yes	No	High-end nodes	No	-	No	Proposal
(ABUTEIR; FLADENMULLER; FOURMAUX, 2016)	Yes	No	High-end nodes	Yes	-	No	Deployment
This work	Yes	Yes	High-end and low-end nodes	Yes (for assessment)	Yes	Yes	Deployment / Emulation

Source: The Author, 2018

3 APPLICATION SCENARIO

This chapter describes the application scenario for military reconnaissance missions in which video streaming are used to survey an area of interest. A combination of small UAVs and ground vehicles is explored to gather information about the environment and to provide awareness of threats. Also, the use of TENs is explored to integrate heterogeneous communication devices in a harsh battlefield environment, such WSNs, small UAVs and ground vehicles in which the network is likely to be disrupted.

3.1 Intra-Partition Scenarios

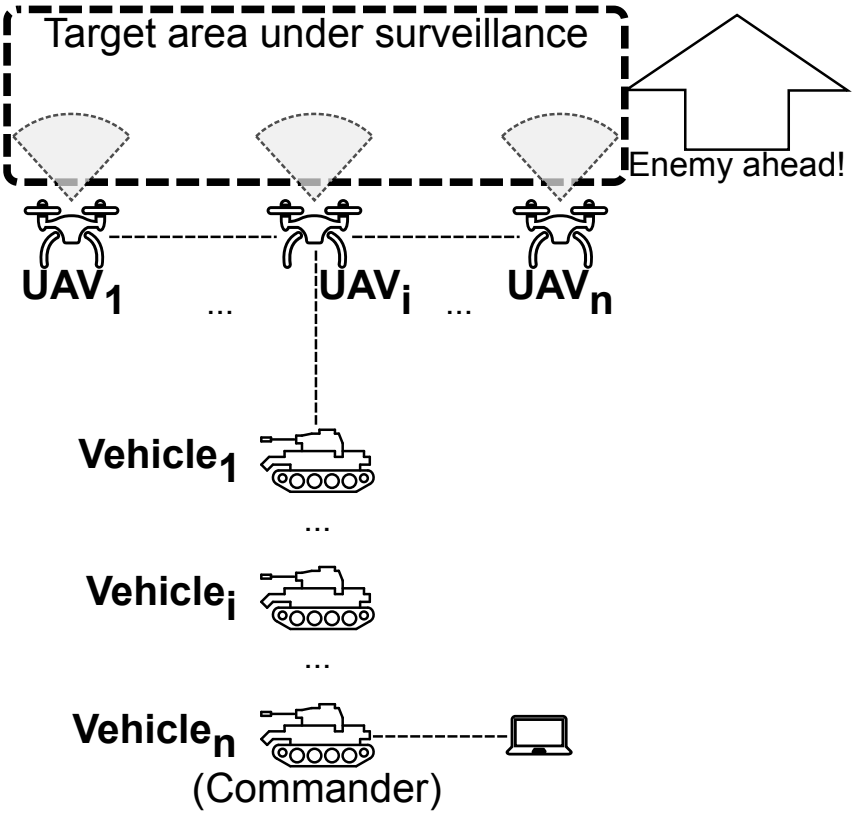
Several areas need to be monitored by the military forces (e.g. borderline, critical infrastructure, enemy-occupied areas, and other harsh environments). To carry out this monitoring, a suitable proposal is to use a fleet of UAVs. UAVs equipped with visible light or infrared cameras should obtain surveillance video of the monitored site. The UAV sends the captured multimedia data to a command center, where the video will be examined and the information gathered from it will be used for decision making.

The UAVs and the Command and Control (C2) systems form a network which may be partially disconnected or disrupted due to the possibility of wide UAV movements. On one hand, wide movements allow the surveillance of a larger area. On the other hand, by flying too far from the access points, UAV may be temporarily disconnected from the remaining of the network. A study considering the trade-off between coverage area and maintenance of a relay network was addressed in (ORFANUS; FREITAS; ELIASSEN, 2016).

Focusing on military reconnaissance missions, in which military troops have to survey an area to gather information about the enemy occupation, the combined use of small UAVs and conventional ground military vehicles is a promising setup. This combination can provide awareness of threats expected ahead of the troop's line of sight. In this kind of reconnaissance missions, the military vehicles move along an axis of advance in the direction of the enemy, and the enemy is located ahead of the platoon as shown in the schematic scenario in Figure 3.1.

Another reconnaissance situation in which UAVs can help the troops on the ground occurs when a reconnaissance platoon arrives at a particular area and explores the area to secure it for the installation of incoming additional troops, as schematically presented in

Figure 3.1: Schematic scenario in which UAVs provide clearance to the ground military vehicles advance along the axis.



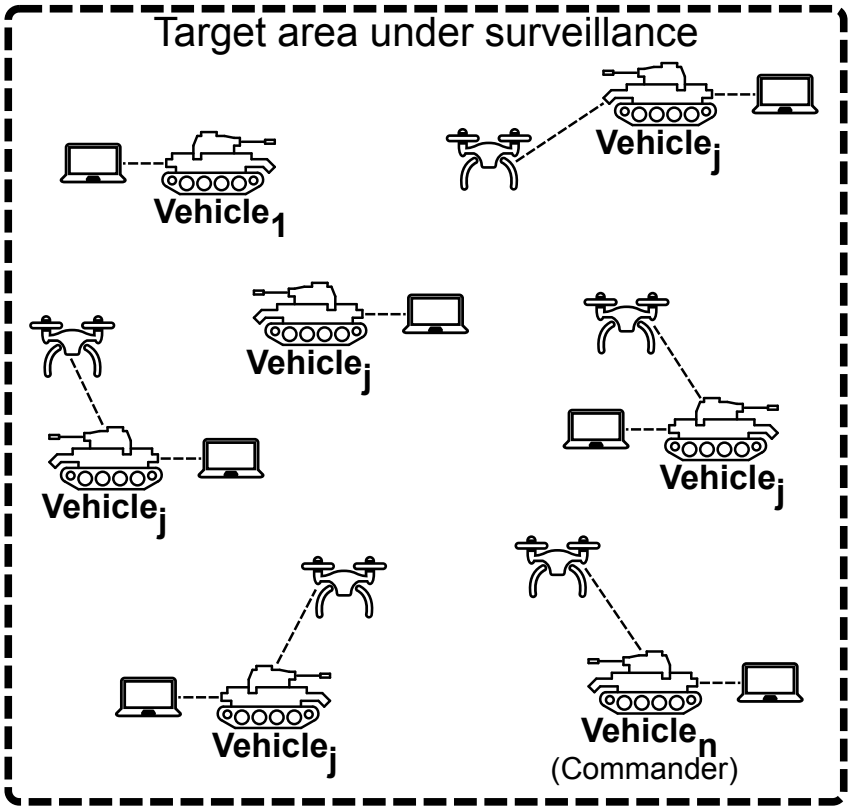
Source: The Author, 2018

Figure 3.2. The UAVs in this scenario are not organized ahead of a platoon, as in the first scenario, as well as the ground military vehicles themselves do not form a platoon. Both the ground military vehicles and the UAVs move freely in the area being explored searching for a possible threat.

In the situation presented in Figure 3.1, the goal is to acquire visual information about the clearance of the area ahead. Using this information the platoon commander can decide about the advancement of his troop. The vehicle occupied by the platoon commander is in general one of the last vehicles in the row. The video acquired by the UAVs has to be delivered then to this vehicle and played in the commander's C2 terminal. In the situation presented in Figure 3.2, the idea is to provide immediate response to hostile elements that may be detected in the area that is being explored. Thus, the idea here is to deliver the video primarily to the closest vehicle. Additionally, the captured video is forwarded to the C2 terminal, located in the commander's vehicle.

A particular behavior should be noticed in the described scenario: The data (*i.e.,* video) source consist of UAVs carrying video cameras collecting images and videos on a surveillance mission. Therefore the video sources are in a highly mobile environment.

Figure 3.2: Schematic scenario in which UAVs provide clearance while the ground military vehicles explore an area.



Source: The Author, 2018

Further, the degradation of communication links directly affect the video quality experienced by the user and therefore, the application evaluation, as it occurs in commercial applications. The QoE objective and subjective metrics studied literature can be used to obtain an overall quality assessment of the service provided to the final user. Additionally, the collected QoE metrics can be used to optimize or select the appropriate path for data transmissions through the network, taking advantage of easy reconfiguration provided by the SDN paradigm, as proposed in this work (NAM et al., 2014).

The video transmission in the described reconnaissance scenarios must meet restrict requirements, under the risk of loosing significant events about the enemy movements. A very important requirement relates to the duration of each possible video interruption. A long interruption in the monitoring video allows an undetected enemy to come close the platoon, posing security risks. For video transmissions triggered by some event (such as the detection of a person or a vehicle), the video playback start time is also important. Due to intermittent characteristics of these type of video stream, delays in the begin of the stream can lead to the same situation caused by long term video freezes.

Taking a concrete example in the first scenario presented in Figure 3.1, the pla-

toon moves at a speed of 60 km/h along the axis of advance. The enemies approach at the same speed in the opposite direction and that the UAVs can provide videos of the enemies movements 1 km ahead. It gives only 30 seconds before the line of contact. Thus, video freezes closer to this period make a reaction practically impossible. Another example, but related to the second scenario presented in Figure 3.2, may consider the video acquisition to gather detailed information about possible enemies infiltrated in the area under surveillance being explored. In this case, considering the second scenario, differently from the first one, possible freezes are not too important, but the resolution of the provided images once something is detected is of primary importance.

3.2 Inter-Partition Scenarios and Partitioned Networks

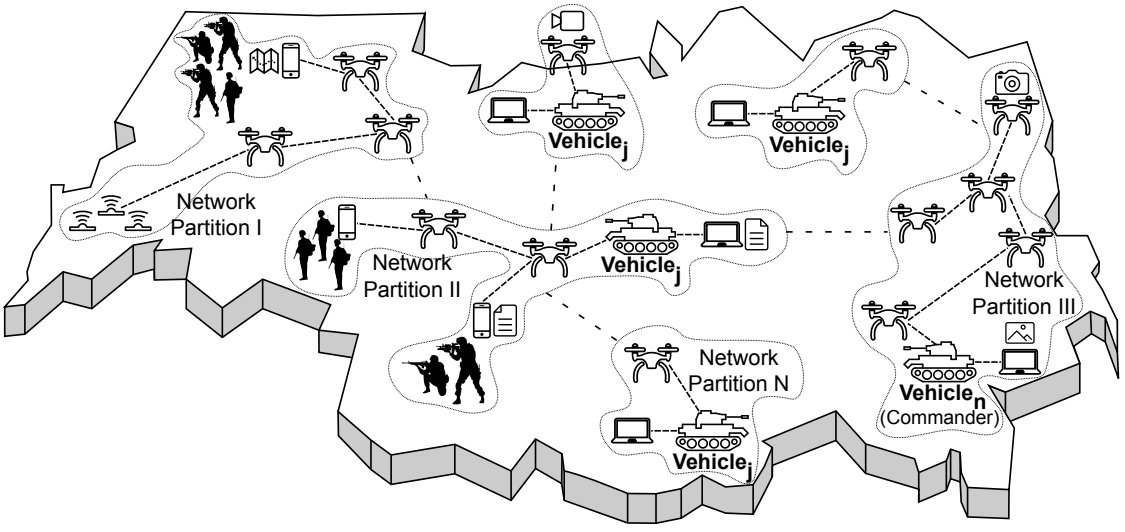
TENs are used to integrate communications devices in harsh battlefield environment. These application scenarios include borderline surveillance and exploration of enemy-occupied areas, for instance. Access to the environment may be limited or restricted, and often there is no preexistent communication infrastructure like cellular 4/5G networks. The communication network in these scenarios is very dynamic by their inherent nature, due to the presence of mobile nodes such as small UAVs, ground vehicles, and devices carried by soldiers. Small and energy constrained devices also join the TEN, such as piezoelectric sensors used to detect the presence of enemy forces. In order to benefit from the data acquired by these nodes, there is a clear need to network them allowing data exchange and forward. The importance of this data exchange and forwarding is to feed commanders with raw-material for their decision-making process supporting the management of the actions in the assigned missions.

The data exchanged among TEN nodes are very diverse and range from simple and single-valued data collected by small sensors to a large amount of data generated by video capture devices (e.g. visible light and infrared cameras, SAR devices, and electro-optical sensors) and file transfers. The data flows have different QoS requirements according to the applications. For example, a file transfer application has flexible requirements related to delay, delay variation (jitter), and bandwidth compared to a video surveillance application that requires near real time transfers. A delayed video stream transmission of a surveillance application may impact the decision-making process which may compromise the action in the area under surveillance. In this context, small UAVs can be used to fly ahead of the troops gathering information about the ground, threats and enemy

positioning.

Examples of the above-mentioned applications in last mile TEN are depicted on Figure 3.3. An important aspect to be observed in this figure is that there is not a single (big) network, but several small networks which occasionally and opportunistically interconnect with each other. For instance, a group of small UAVs launched from a ground military vehicle to video survey the terrain ahead forms themselves their own network. This network may not be connected to the launching vehicle for a given time interval, due to interferences or range limits. However, this connection may be eventually restored and a UAV closer to a launching ground vehicle can deliver the video acquired by one of them. Another example illustrated in the left-hand side of Figure 3.3 is a wireless sensor network (WSN) for motion detection which may be completely disconnected from other nodes outside the WSN. These sensors acquire and process information about enemies movements in the area and when a friendly node comes into the range, e.g. a UAV, they deliver the acquired data. Following this example, this UAV by its turn can forward the data through a UAV-relay network to C2 applications running on mobile devices carried by troops on the field. As all these applications have different requirements, it is possible to systematically classify them so that a judicious analysis can be performed to address these specific requirements.

Figure 3.3: Last-mile TEN application scenario.



Source: The Author, 2018

Information dissemination is a critical part of C2 systems, which directly affects the battlefield scenario. The use of information grids is a popular approach in network-centric warfare, allowing situation awareness sharing through data exchange among different applications and systems (TORTONESI et al., 2013). Legacy and COTS appli-

cations are also often used in C2 systems and messages exchange among them usually benefit from the usage of Services-Oriented Architecture (SOA) in TENs (BLOEBAUM et al., 2016). SOA messages exchanged among different devices in TEN can deal with limited bandwidth and long round trip time (RTT) of packets. The use of intermediary nodes caching the information and forwarding them over opportunistic wireless links introduces small delays in the communication. However, these delays do not represent a major problem in this kind of application and can be tolerated. File transfers, such as photos and topographic maps, among devices in the TEN belong to this same type of application and are classified as *elastic applications* (KUROSE; ROSS, 2013).

Applications that make use of video to survey or secure an area should meet more strict network requirements. They are used in situations in which quick decision-making is required, thus they are not as tolerant to network service degradation as the elastic applications. By nature, video transmission demands higher throughput and the application cannot support bigger network instability, like throughput limitation and significant latency variation (jitter). The bandwidth and latency variations result in inefficient video data transfers and low-quality video playout with freezes and degradation of displayed images, thus leading to loss of important details which may lead to wrong decisions. This refers to the value of images and video in the decision-making process, which is an important concern that cannot be disregarded (FERNANDES; HIEB; COSTA, 2016). Considering these characteristics, the group of applications dealing with video transmission and continuous data flows with strict QoS network requirements can be classified as *non-elastic applications*.

Applications using bi-directional data transfers, such as video conferencing and voice over IP, are also valuable in networked warfare systems. The network requirements are very strict in this kind of applications because the communication is an interactive process. The network should support a minimum throughput allowing the exchange of video and/or audio and additionally meet strict requirements of latency variation (jitter). For these applications, end-to-end connections must exist between the points that wish to communicate. Therefore, the temporary storage of data at intermediate nodes using DTN solutions is inappropriate. Applications with these restrictive characteristics are classified as *interactive non-elastic applications*.

Table 3.1 summarizes the main characteristics of these three application classes, comparing their demands concerning network requirements and their suitability to use DTN intermediary storage.

Table 3.1: Applications classes and network requirements

	Application class		
Parameter	**Elastic**	**Non-elastic**	**Interactive non-elastic**
Network throughput usage	Low	High	Medium – High
Network latency tolerance	High	Low	Low
Jitter tolerance	High	Medium	Low
Packet loss tolerance	Low	Medium – High	Medium
DTN storage allowed	Yes	No	No
Example applications	File transfer, messaging	Video surveillance	Video and/or audio conference

Source: The Author, 2018

In addition to network resources requirements, other important aspects that need to be observed are:

- The deployment of new devices should meet strict timing requirements. Easy of configuration and quick device replacements are an important desired feature.
- Considering its high dynamicity, the network has to be prepared to promptly respond to changes in the topology, being able to provide even self-healing properties.
- The different data flows demand different QoS levels according to applications classes.
- Security is a requirement of primary order considering military networks. Thus, the immediate employment of security mechanisms to ensure security policies is a must.
- The nodes operating in TENs are often under energy constraints. Optimized routing algorithms and energy-aware protocols are desirable to extend the operation time of battery powered devices.
- There are a lot of already deployed and legacy applications operating in military networks. The need for significant changes in applications' source code should be avoided. Existent applications should be considered and seamlessly integrated.
- There is a need to avoid SPOFs in military systems. Considering this need, link redundancy mechanisms are essential components in military networks.

4 THE PROPOSED SDN-DTN MILITARY NETWORK ARCHITECTURE

Military networks, such as TENs, operate in harsh environments and therefore constitute a DIL environment. The transmission of data in such environment is a challenging task due to the occurrence of events that can lead to delay in the data transmission, loss of data, and intermittent and limited data links between the network nodes (NIGHTINGALE et al., 2016). Applications are affected by the link characteristics depending on the application class (see Table 3.1). Video applications, which are a collaboration service according to the NATO C3 Taxonomy (BRANNSTEN et al., 2015), are classified as non-elastic or Interactive non-elastic applications, and therefore to hold acceptable quality on the transmission of these type of data is not trivial.

The next sessions of this chapter propose the application of SDN to improve the quality of video transmission in TENs. Next, a combination of concepts from the SDN and DTN paradigms is presented to tackle the problem of the intermittent connections on disrupted networks and at the same time, to take advantages of the centralized management which is a characteristic of the SDN paradigm. Our application also makes use of a leader election algorithm to choose the node that will act as the SDN controller, which is presented in the last part of this chapter.

4.1 An SDN Architecture to Enhance Video Streaming in Dynamic Networks

The expansion of the ground coverage area maintaining connectivity among UAVs and the ground platoon is one of the problems to be addressed in the first scenario described in Session 3.1 and depicted in Figure 3.1. The increase of the coverage area while retaining connectivity to the rest of the ground squad can be acquired using intermediate UAVs as relay nodes. The relay nodes act forwarding the data until it reaches the destination node. The use of intermediate relay nodes has a drawback: the overload of the intermediate nodes. Intermediate nodes send data collected by their sensors (e.g. video cameras) and additionally the data sent by the neighbors to the intermediate node. The neighbors are not able to establish an end-to-end connection with the destination nodes, therefore they should use other nodes within their communication range to deliver the acquired data (the video stream) to the destination node. In this arrangement, the throughput of the communication channels is shared among different video streams.

The concurrency of the throughput by many video flows directly affects the video

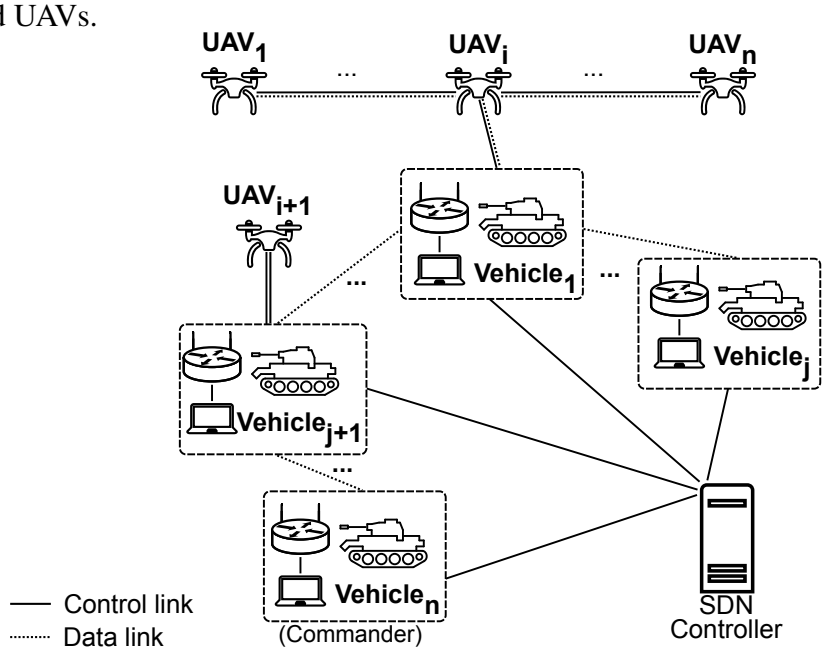
QoE perceived by the user, negatively impacting video applications and leading to a bad evaluation of the application by the user. The negative feedback of a video application is expressed by a low MOS score. The selected path to forward the video stream and the number of hops needed to reach the destination has a significant impact on this assessment. A management entity with global view of network status is able to select optimal link paths to route the data, avoiding congested links and choosing paths with fewer hops. A software defined network based on the OpenFlow (MCKEOWN et al., 2008) architecture enables the network controller to gather information about the global state of the network. The centralized controller can detect congested and deteriorated communications links that lead to low throughput and high latency, therefore insufficient resources for the video stream. The features present in OpenFlow allow the network reprogramming, applying a routing protocol capable of distributing the data streams among redundant links (SINGH; DAS; JUKAN, 2015) or select a path that offers the appropriate resources for the video transmission avoiding the video freeze, for example.

Video streams used in military surveillance applications show slightly different characteristics to those used in home entertainment systems. The purpose of the captured video in military missions is the rapid detection of threats, gather information of the resources or activities of a potential enemy, to obtain reliable information about certain areas or assist in the organization of the units. The features of the remote video terminals (RVTs) and displays on embedded systems also influence the choice of the videos resolutions. The images are usually obtained by visible light cameras, synthetic aperture radar (SAR), electro-optical and infrared devices. The videos used in the military context often range from images with resolutions of few Kilopixels to HD images (e.g. 1920x1080 pixels)(ARMY, 2006; L-3 WESCAM, 2015; Stark Aerospace Inc, 2015). Focusing on reconnaissance missions in which the UAVs are used to survey an area ahead of troop line of sight, like that presented in Chapter 3, Figure 3.1, videos with low resolution (e.g. 640x512 pixels) are enough to provide the information to enable decisions about the current situation, while this resolution make them more appropriate for transmission using disadvantaged networks (NIGHTINGALE et al., 2016). Whereas, in the combination of UAVs and ground vehicles used to secure an area for the installation of additional troops, like depicted in Chapter 3, Figure 3.2, a higher video resolution is required to detect and identify threats nearby ground troops (ARMY, 1991; ARMY, 2009).

The proposed network architecture provides a software-defined environment to manage the data and the SDN controller to handle the network of ground vehicles. This

network of the ground vehicles can be considered stable compared to the network of UAVs, which enables an efficient usage of the SDN controller. Each of the ground vehicles are equipped with a data forwarding device supporting the OpenFlow protocol enabling the creation of a programmable network among them. All of the data forwarding devices are connected to a same SDN controller, enabling a centralized management of the network. Thus, in this proposed architecture each ground vehicle is considered a network switch controlled by the SDN controller, as presented in Figure 4.1. This figure presents the general idea of the proposed architecture, which can be implemented both for the first and the second scenario configurations presented in Figure 3.1 and Figure 3.2.

Figure 4.1: Overview of the proposed SDN employment in network of ground military vehicles and UAVs.



Source: The Author, 2018

The UAVs move around the area where they are conducting the surveillance mission, and therefore, the connection to the network composed of ground vehicles is not stable. During the mission, the UAVs can often reconnect to the same ground vehicle or connect to a different one. The SDN controller must optimize the network to make the reconnection process as smooth as possible, causing minimal impact on the quality of the video being displayed to the user.

When starting the network operation, a topology discovery process is performed. The topology discovery process allows the controller to select the most suitable path for the transmission of the video streams generated by each of the UAVs to the final users' remote video terminal. Once an OpenFlow-enabled device connects to the SDN controller through a TCP connection, the controller sends a feature request message to the device

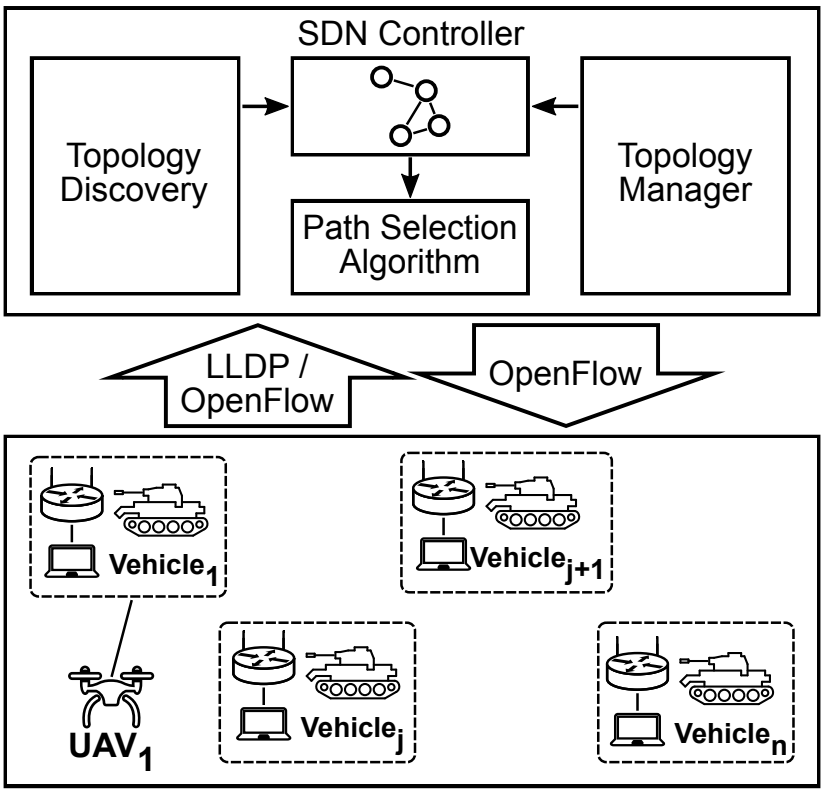
and waits for a response. The device answers with a feature reply message containing its characteristics (e.g. datapath id and ports). This message exchanges is part of the OpenFlow protocol handshaking process and allows the controller to be aware of all forwarding devices in the SDN data plane. The LLDP protocol performs the discovery of data links among the already discovered devices.

In the SDN controller, the topology data gathered is represented by an undirected graph. The graph vertices represent the forwarding devices and access points found, and the graph edges represent the data links among discovered devices. The controller computes the routes among the devices based on the information contained in the network graph representation. During the mission, the topology of the network can change. The network devices (forwarding devices and access points installed in the ground military vehicles) can establish new links among them, and between network devices and UAVs. Additionally, due to UAV moves, existing connections may disappear. In this case, the graph needs to be updated to correspond to the new network topology. The controller detects OpenFlow-enabled devices that are joining or leaving the data plane through the OpenFlow Channel. Furthermore, the controller check for updates in the links by periodically checking their state using the LLDP protocol. When a topology event occurs, the network graph representation is updated.

After the initial topology discovery of the OpenFlow-enabled devices, the SDN controller is ready to process the requests from forwarding devices in the data plane. The OpenFlow / LLDP discovery process does not detect the UAVs in the TEN network, and the prototyped SDN controller identifies the UAVs based on a Layer 2 learning process. When the UAV joins the TEN through a wireless connection with a ground vehicle, the OpenFlow-enabled access point does not have a configured matching rule to forward the packets. Following the default behavior of an OpenFlow specification, and due to the existence of a previous installed table miss entry, the forwarding device sends the data packet to the SDN controller. The key components of the SDN Controller that perform these tasks and their relations among each other, are presented in the schematic SDN controller architecture illustrated in Figure 4.2.

Each data packet that arrives at the SDN controller triggers the *Packet_in* event. The SDN controller follows the algorithm depicted in the SDN controller flowchart, Figure 4.3. As a UAV joins the TEN, the network graph representation does not contain information about the UAV, thus the SDN controller updates the network graph representation. Next, the SDN controller performs a search for the destination address host

Figure 4.2: Schematic SDN Controller Building Blocks



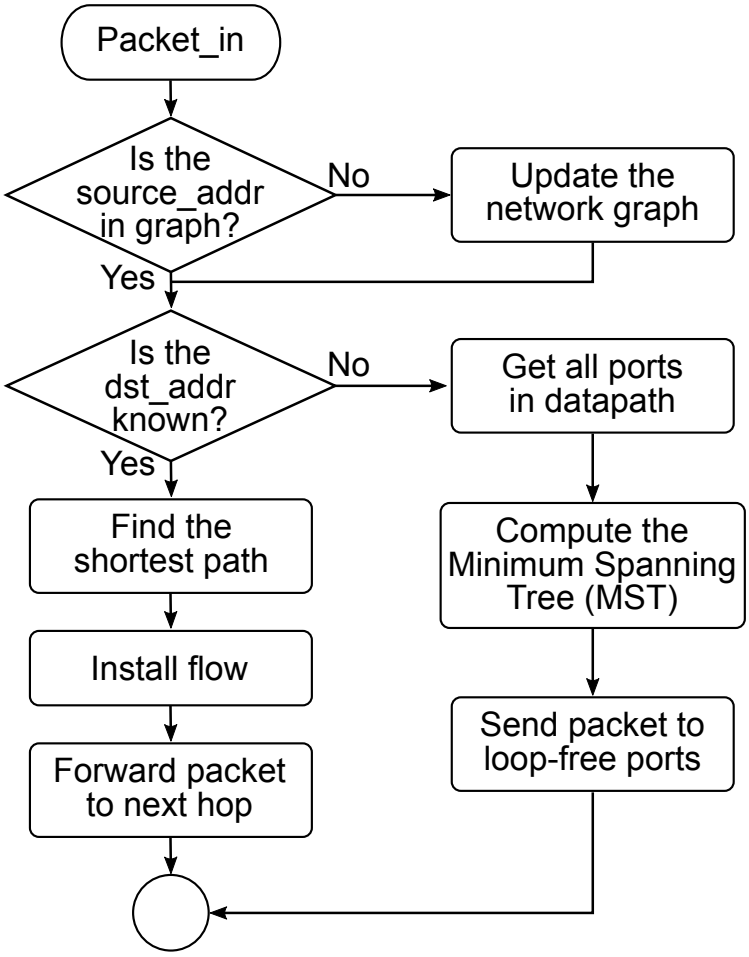
Source: The Author, 2018

dst_address in the network graph. If the target host specified by the *dst_address* field in the data packet is found, the network controller installs an OpenFlow entry in the forwarding device that originates the *Packet_in* event and forwards the packet to the next hop, or to the destination host. Otherwise, if the network graph representation does not have information about the host specified by the *dst_address*, the controller floods the network with an ARP Request packet until the destination host answers the request with an ARP Response packet. To prevent packets from being forwarded in network loops, the controller gets information about all ports in the *datapath* (logical representation of the forwarding device), compute the ports that may cause loops in the network using a minimum spanning tree (MST) algorithm and sends the ARP Request package only to loop-free ports.

4.2 Combining SDN and DTN in Military Networks

The last-mile TEN represents a challenging network environment that can be explained by the high mobility of the nodes, device heterogeneity, resource constraints, and specific (and hard to meet) requirements of military applications. This network con-

Figure 4.3: Path Selection Algorithm.



Source: The Author, 2018

nects applications with different requirements with regard to QoS, security, and reliability. When the concepts of SDN are merged with those provided by DTN, they offer the means to tackle this challenge. This section describes the objectives of the proposed architecture together with the best features offered by each approach.

By following the SDN paradigm, forwarding devices are remotely managed by a centralized SDN controller, although this does not imply that it is a physically centralized entity. TENs must be robust and tolerate attacks so that they are able to continue operating even with damaged components. Redundancy mechanisms for the SDN controller are required to achieve the desired robustness.

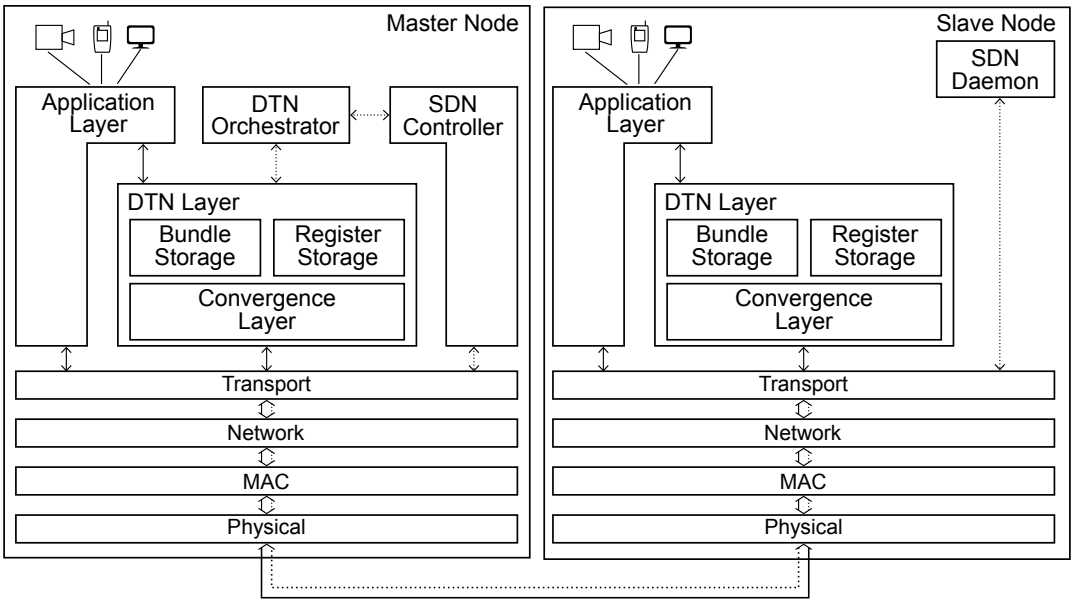
The use of multiple controllers in the TEN overcomes the SPOF problem by allowing the network to continue operating when it is disrupted. By selecting a scenario in which small UAVs form a part of a TEN, each of them can take control of the network by running a controller instance. However, unnecessary control processing leads to waste of energy. On the other hand, a single instance of an active controller for the whole net-

work will be unreachable for nodes located in different network segments. In this case, an SDN controller should be selected in each segment by running a leader election algorithm. When all the nodes return to the same segment, the leader election algorithm is run again and elects a single controller for the entire network. The elected controller might be one of the controllers previously active in one of the partitions, or even a new one running in a different node. In this case, the new controller is synchronized with the previous ones to ensure that consistency is kept with the already applied configurations. Stress should be laid on awareness of the overhead, or even the infeasibility of this election process due to the extreme dynamics of such networks. In these cases, the proposed architecture should be slightly changed and include, for instance, an out-of-band control channel. The assumption here is that the dynamics of the network is at a degree in which changes are expected to happen in minutes. This is plausible in these scenarios and ensures the feasibility of the approach.

The node that runs an active instance of the SDN controller is the master in that network segment, and responsible for its control. The other nodes in the same segment are called slaves. Figure 4.4 shows the internal architecture of the master and slave nodes on its left and right sides, respectively. Instead of an SDN controller, the slave nodes run an SDN daemon component (on the right side in Figure 4.4), which continuously surveys the current network segment to detect active SDN controllers. When an SDN controller is not found, the SDN daemon starts the leader election algorithm. When a slave node becomes a master (the new master elected node), the SDN daemon initializes the SDN controller and the DTN orchestrator components in that node (top right and top middle of the master node in Figure 4.4). The other slave nodes in the TEN segment will be notified and have to update their configurations to connect to the recently elected controller.

There may be a huge number of messages exchanged between the master (SDN controller) and slave nodes, thus increasing energy consumption. To address this concern, the SDN daemon manages the migration of the SDN controller process among the nodes forming the current network segment. This management system aims to avoid battery depletion of the master node, thus increasing the autonomy of the small battery-powered devices that are often used in last-mile TENs. It should be noted that an instance of the SDN controller will run on a node when it is isolated from the remaining network. This controller will only manage this node's network interface. Also, it will watch for an opportunity to rejoin a network segment, and when this occurs, it will run the leader election algorithm and forward the temporally stored data that it may have.

Figure 4.4: Combined SDN and DTN architecture



Source: The Author, 2018

The communication between the forwarding devices and the SDN controller uses the managed network, and therefore an in-band controller connection (KIM; FEAMSTER, 2013). The OpenFlow messages are routed to controlled devices through conventional network connections, and hence traverse the transport, network, and medium access control (MAC) layers, and finally the physical wireless interface (Figure 4.4, at the bottom of both the master and slave nodes).

The DTN orchestrator coordinates the functionality of the DTN nodes. This module exchanges link information with the SDN controller in the master node, as can be observed in Figure 4.4 (on the left side). The DTN orchestrator schedules data transmission between the DTN nodes on the basis of data collected by the SDN controller. The SDN controller has updated information and a "global view" of the network, that is, a global view of the network segment it is controlling. Data flows in TENs have different QoS and security requirements (NOBRE et al., 2016) and the SDN controller keeps track of them. The DTN orchestrator also gathers information from the DTN layer about the bundle storage state, buffer utilization, and DTN endpoints (register storage). The acquired information can be used to select a different DTN routing algorithm or forward the data so that it can be stored temporarily in another node, thus avoiding the overflow of the DTN buffer.

The DTN layer, represented in Figure 4.4 (in the middle of both master and slave nodes), follows the concepts of current DTN implementations. The bundle storage com-

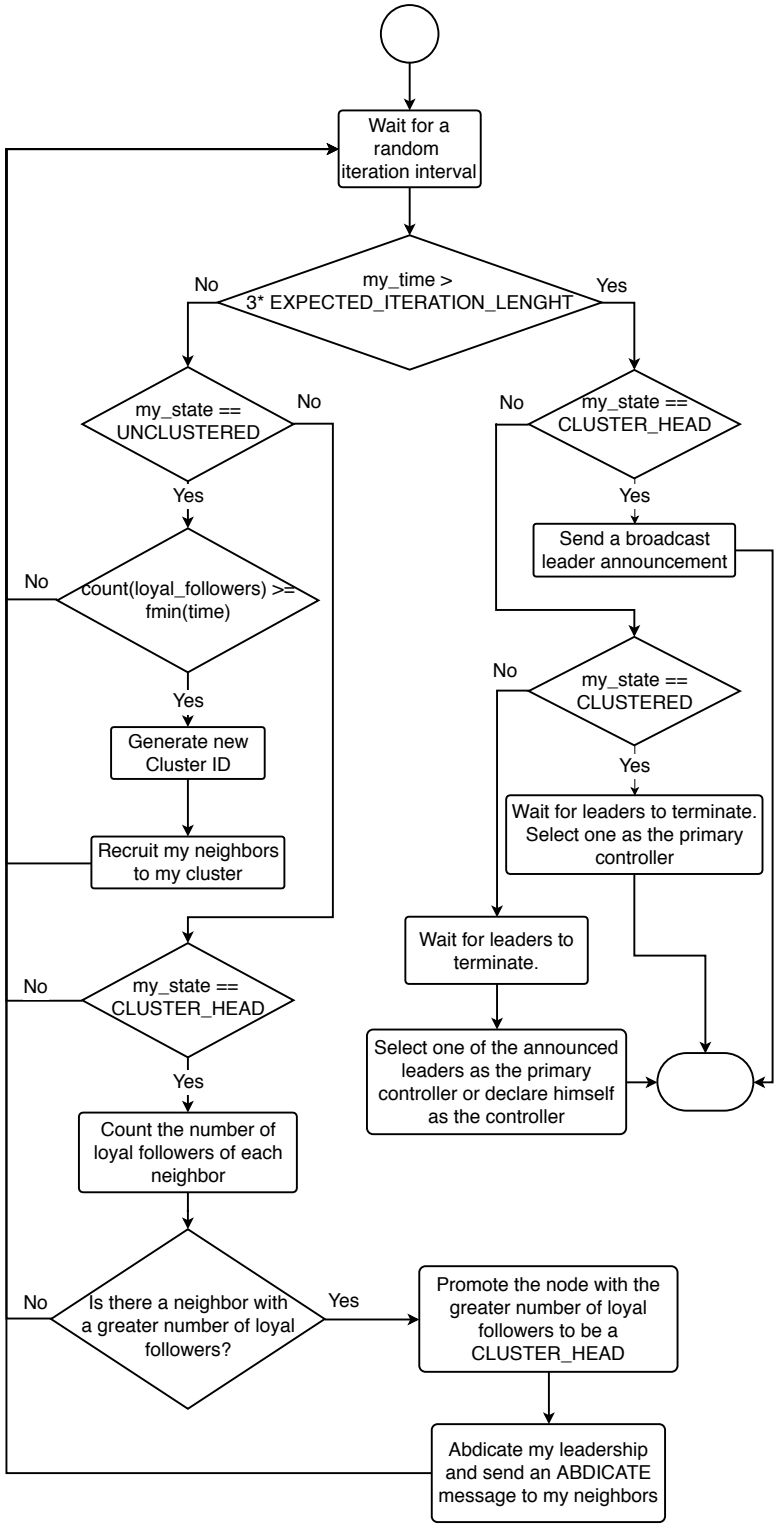
ponent is used to store the data collected by devices and applications when the destination node is unreachable (e.g., the destination node is in another network segment or is temporarily turned off). The convergence layer links the bundle storage to the transport layer. The data stored in the bundle storage buffer are sent to the destinations or neighbor nodes using the UDP or TCP transport layer protocols. The delay-tolerant data, which are temporarily stored in nodes, are routed by means of specific DTN algorithms. There are many routing algorithms for DTNs, and a particular set of routing algorithms can be employed to route the data between the DTN nodes in TENs, such as geo-routing algorithms. The DTN orchestrator manages the routing protocol switching, and it chooses the most suitable one in accordance with internal decision algorithms and the data exchanged with the SDN controller. The flexibility provided by the SDN makes it easier to change the routing protocol while the TEN remains running.

4.3 Choosing a Node to act as a Master in the TEN

Although any leader election algorithm can be chosen to elect the master node in the network partition, some characteristics are highly desirable, for example, the ability of elect himself as the master node (or leader) when the node is isolated from the network. The election process follows the algorithm depicted in Figure 4.5, and it is inspired in a leader election algorithm for WSN (CHAN; PERRIG, 2004). The ACE-based algorithm presents all the desired characteristics, the implementation is relatively simple, and the algorithm parameters can be tunned to reflect the field size and the number of nodes operating in the filed. Although the implementation of leader election is limited to one algorithm, the process could use another leader election algorithms that consider unreliable processors in a network like the Paxon Parliament protocol (LAMPORT, 1998).

After the SDN daemon starts the election, the node will wait for a random time interval. While the algorithm is under execution, the node also waits for a random time interval between iterations. The time interval is uniformly random distributed. Each node can assume one of three possible states, UNCLUSTERED, CLUSTERED and CLUSTER_HEAD. At the beginning of the algorithm, the node assumes the UNCLUSTERED state. The UNCLUSTERED state means that node is not a master node (*i.e.,* a leader) neither a slave node (*i.e.,* a follower of any leader). Assuming the CLUSTERED state, the node will act as a slave node, and finally, the CLUSTER_HEAD state indicates that node will act as a master node.

Figure 4.5: Flowchart representing the operation of the algorithm used for the election of the “master node”.



Source: The Author, 2018

Each node knows how many time has passed since the beginning of the algorithm. This time is represented by the variable *my_time* in the Figure 4.5. If the value of *my_time* is less than $3 \times EXPECTED\_ITERATION\_LENGHT$, the node execute an-

other iteration. If the node is UNCLUSTERED, it counts the number of loyal followers (*i.e.,* neighboring nodes that are UNCLUSTERED) that it will receive starting a new cluster. The node compares the number of loyal followers with the result of the function $fmin(my\_time)$. The $fmin(my\_time)$ function is represented in (4.1) and the value of the constants used in the algorithm are shown in Table 4.1.

Table 4.1: Parameter and values used for the leader election algorithm

Parameter	**Used Value**
k_1	2.3
k_2	0.08
cI	10
MAX_WAIT_TIME	2.5 s
$EXPECTED\_ROUNDS$	4
d	Estimated node degree (vary according to the number of nodes in the scenario
$EXPECTED\_ITERATION\_LENGHT$	1.5 s

Source: The Author, 2018

The values of k_1 and k_2 in (4.1) define the shape of the exponential graph representing the threshold of a node to become a leader in the case the node remains in the UNCLUSTERED state. If at a given time *my_time* the number of loyal followers – *i.e.,* the number of neighboring nodes in the UNCLUSTERED state – that node will get if it declares himself as a CLUSTER_HEAD is greater or equal to $fmin(my\_time)$, the node go ahead and will start a new "cluster" of nodes, declaring itself as a CLUSTER_HEAD and generating a new cluster ID to identify the new "cluster" or in other words, a group of nodes being controlled by the same leader.

$$fmin(my\_time) = \left(e^{-k_1 \times \frac{my\_time}{cI}} - k_2\right) d \tag{4.1}$$

The new CLUSTER_HEAD node will issue a message recruiting the neighbors to become followers of the new cluster. The *my_time* is the time since the node began the cluster election algorithm. The cI value is given by (4.2), where MAX_WAIT_TIME corresponds to the maximum waiting time between each algorithm iteration and the value of $EXPECTED\_ROUNDS$ is the expected number of the algorithm iterations. The used values are also shown in Table 4.1.

$$cI = (MAX\_WAIT\_TIME \times EXPECTED\_ROUNDS) \tag{4.2}$$

The value of the d constant in (4.1) is the average node degree, hence it may change depending on the number of node in the scenario, the field size, and the communications range of the nodes. The values used for each scenario, ranging from 5 to 30 nodes are shown in Table 4.2. As the scenarios were randomly generated, the values of d can vary. The minimum value used is 3.20 for scenarios with 5 nodes and the maximum value is 6.4 for scenarios with 30 nodes.

Table 4.2: Average node degree (d)

Node in the scenario	**Estimated average degree**
5	3.20
10	3.80
15	4.13
20	5.10
25	6.40
30	5.73

Source: The Author, 2018

If the node state is CLUSTER_HEAD when starting a new iteration, the node will ask its neighbors how many loyal followers each neighbor will get if that neighbor starts a new cluster. If the node finds a neighbor with a greater number of potential loyal followers, the node will promote the neighbor node to a CLUSTER_HEAD and become a follower of the promoted CLUSTER_HEAD. Following the execution, the node will abdicate its position as CLUSTER_HEAD. It the node does not find a neighbor with a greater number of potential loyal followers, the node does nothing and waits for the next iteration.

If the node state is CLUSTERED, then the node will do nothing. It does not matter the node state, at the end of the iteration, the nodes will wait for a random time interval before beginning the next iteration.

At the begging of each iteration, right after waiting for the time interval between iterations, the node checks whether the time passed since it started the leader election algorithm. If the value of my_time is greater than $3 \times EXPECTED\_ITERATION\_LENGHT$, the node can finish the election process. Depending on the actual state of the node, some additional actions are performed. If the node state is CLUSTER_HEAD, it will broadcast a message to the network, containing its number of loyal followers and is address. This information is used in the case of more than one CLUSTER_HEAD were elected. The node with the greater number of loyal followers will act as the master node. It the node state is CLUSTERED, it will select the CLUSTER_HEAD with the greater number

of followers to be its master node. It at the end of the leader algorithm a node remains UNCLUSTERED, it will select one of the cluster head nodes announced via a broadcast message. If the node does not have information about a cluster head, the node probably is isolated from other nodes; thus it will declare himself as a cluster head and act as a master node.

5 EXPERIMENTS AND RESULTS

This section presents the experiments carried out by means of emulation, and the obtained results. Initially, the selected tools, implementations and changes introduced in existent software to enable collection of measurements are reported. Next, a description of the parameters, the characteristics of the simulated scenario and the obtained results and discussions about their effects in the proposed application scenario are presented, for both, intra-partition and inter-partition experiments.

5.1 Simulation Tools

The experiments and evaluations were performed using the emulator for software-defined wireless networks Mininet-Wifi (FONTES et al., 2015). The Mininet-Wifi is a fork of the well-known Mininet emulator (LANTZ; HELLER; MCKEOWN, 2010). It adds wireless emulation features to the original Mininet emulator. Mininet-Wifi also integrates mobility models used to simulate the movement of the emulated network nodes corresponding to the UAVs in the simulated environment. The tool also enables Linux compatible programs to run on the simulated hosts using a real kernel (e.g. media server and media player applications).

The Mininet-Wifi supports many mobility models. The mobility models are used to provide movements to the access points and wireless stations in the simulated environment. Two mobility models, the Random Walk model and the Random Waypoint model, supported by Mininet-Wifi were selected for the experiments regarding video transmission on TEN and assessment of QoE perceived by the user on a dynamic environment.

The Random walk model was initially created to emulate the unpredictable movement of physical particles. It is believed that some nodes in mobile networks behave in the same fashion, with unpredictable movements, and thus the Random Walk came to be used to mimic their movement. At each predefined time interval, the nodes select a new direction to move toward. At each predefined time interval, the nodes select a new direction to move toward in the range $(0, 2\pi]$ and a new speed from limited range of values.

In the Random Waypoint movement model, the nodes randomly select a destination point in the simulation field. The nodes travels towards the selected destination with constant velocity chosen uniformly and randomly from a predefined range of values. The destination point and the speed of each node is chosen individually. When the node ar-

rives at the destination point, the node stops for a defined pause time, and then repeats the process selecting a new destination. Because of its simplicity and availability, the Random Waypoint is a kind of benchmark mobility model used to evaluate the performance of routing protocols in “Mobile Ad-hoc Networks” (BAI; HELMY, 2004).

The video AppQoE measurements were performed using a modified version of the FFplay player. According to the software documentation, FFplay is a portable media player using the FFmpeg libraries (BELLARD, 2000). It is mainly written using the “C” programming language. Small changes were made in the source code of the FFplay, and a recompiled version of the player including the modifications was used in experiments. The first change was implemented to make it possible to compute the video initialization time. A timer starts when the player requests the video data to the server. Once the player receives enough data to display the first frame of the video, the video playback start time is computed. A modification that monitors the player video queue buffer computes each of the video stalls. The FFplay stores the received data in a video queue buffer. If the network throughput is lower then the required video bitrate, the buffer becomes empty, causing a video stall. The changes in the player source code can compute each video stall and the stall length. The acquired data is reported in the player log allowing a data analysis later.

Each of the UAV hosts, acting as video sources, run an instance of the FFServer media server. The FFserver is a streaming server, for both audio and video streams, and it is also a part of the FFmpeg software package. No changes were introduced in FFserver software to act as a video stream server, and it was configured to serve a video stream over the HTTP protocol. The data is transfered using the TCP protocol.

The algorithm for leader election proposed in Chapter 4.2 used in selecting the master node in each network segment is implemented using the Python programming language. Each node in the network acting as a leader run an instance of the script. The communication between nodes is implemented using network sockets. The nodes establish a TCP connection to exchange the algorithm messages. In a real scenario, it is highly desirable the use o broadcast messages or even a Wireless Access in Vehicular Environments (WAVE) architecture for node interaction. However, due to emulator limits, and to get more control over the communication between nodes, defining which specific nodes exchange messages.

External SDN controllers are used to handle the OpenFlow-capable switches and access points. The OpenFlow version 1.3 is used in the communication between network

elements and the SDN controller. For the Video QoE experiments described in the sub section 5.2, the controller was implemented using the Ryu Framework (Ryu Community, 2014). The Ryu Framework provides several libraries and functions available through a straightforward API, for instance, easing the topology discovery process and the flows installation on switches. For the experiments regarding the leader election algorithm, described in sub section 5.3, the Floodlight SDN controller was chosen. The Floodlight SDN controller is an open source Java-based application. Its highly modular design and the big collection of functions accessible through the controller API make the configuration of the OpenFlow-capable devices easy.

5.2 Intra-Partition Experiments

In this subsection, the experiments related to the architecture proposed in the Chapter 4.1 are described.

5.2.1 Selected Evaluation Metrics

Three objective metrics were elected to quantify the Application Quality of Experience (AppQoE) on the client side of the surveillance system using video over HTTP (MOK; CHAN; CHANG, 2011; JULURI; TAMARAPALLI; MEDHI, 2016). These metric are:

- *Video playback start time*: Corresponds to the time taken by the player to start the *playout*. As it was used a standalone video player application, the measured time corresponds to the request of the stream, the download of the initial part of the video filling the player buffer to a threshold set by the application (*Initial Buffering*), and the playout of the initial part o the video.
- *Number of interruptions*: When the playback is temporarily frozen a video interruption is computed. This event occurs when the throughput of the network is not enough to keep the video player reproducing the video. The player buffer decreases to a low value nearly zero, and the player waits for the buffer to be partially filled again to resume the video playout. This event is also referred as a *(re)buffering* event.
- *Total duration of interruptions*: This metric is a sum of the duration of all interruptions (*Buffering Time*) during video playout. The first buffer event is ignored

because it corresponds to *Initial Buffering* (JULURI; TAMARAPALLI; MEDHI, 2016).

The predictions of the MOS values were obtained from the AppQoE data collected. Recent studies (SEUFERT et al., 2015; HOSSFELD et al., 2012) relate the influences of the three selected metrics with the degradation of user experienced quality, represented by MOS score that would be assigned by the user. In (HOSSFELD et al., 2012), the authors relate the video playback start time with the MOS value using (5.1), where *Qini* is the MOS influenced value and t_0 is the video playback start time obtained in measurements.

$$Qini = -0.963 \times \log(t_0 + 5.381) + 5 \tag{5.1}$$

The MOS value influenced by video stalls (*Qstall*) can be obtained by applying (5.2) and (5.3) to the AppQoE data. The λ factor is the ratio of the total time that the video was stalled(σ) and the interval elapsed since the beginning of observation, given by the sum of σ with the effective video play time (ρ). The value of λ is used to determine the value of the a_i, b_i and c_i constants in (5.3) according to predefined values observed by Casas, Schatz and Hoßfeld (2013). The n variable corresponds to the number of video interruptions on the T time observation interval, considered as one minute in the present work. The values of "n" are mapped for up to 6 interruptions on the video playout, because with a value greater than this threshold, the MOS assumes the value $Qstall = 1$, that means a very bad quality experienced by the user.

$$\lambda = \begin{cases} \frac{\sigma}{\sigma+\rho}, & \text{if } \sigma + \rho < T \\ \frac{\sigma}{T}, & \text{otherwise} \end{cases} \tag{5.2}$$

$$Qstall = \begin{cases} 1, & \text{if } n > 6 \\ a_i \times e^{-b_i \times n} + c_i, & \text{if } n \leq 6 \end{cases} \tag{5.3}$$

Since *Qini* and *Qstall* were obtained independently, from different functions, the minimum value was assumed as the final video MOS, as can be seen in (5.4). Therefore, the final video MOS will be limited by the minimum value observed in *Qini* and *Qstall* (FILHO et al., 2016).

$$MOS = \min\{Qini, Qstall\} \tag{5.4}$$

5.2.2 Simulated Scenario and Parameters

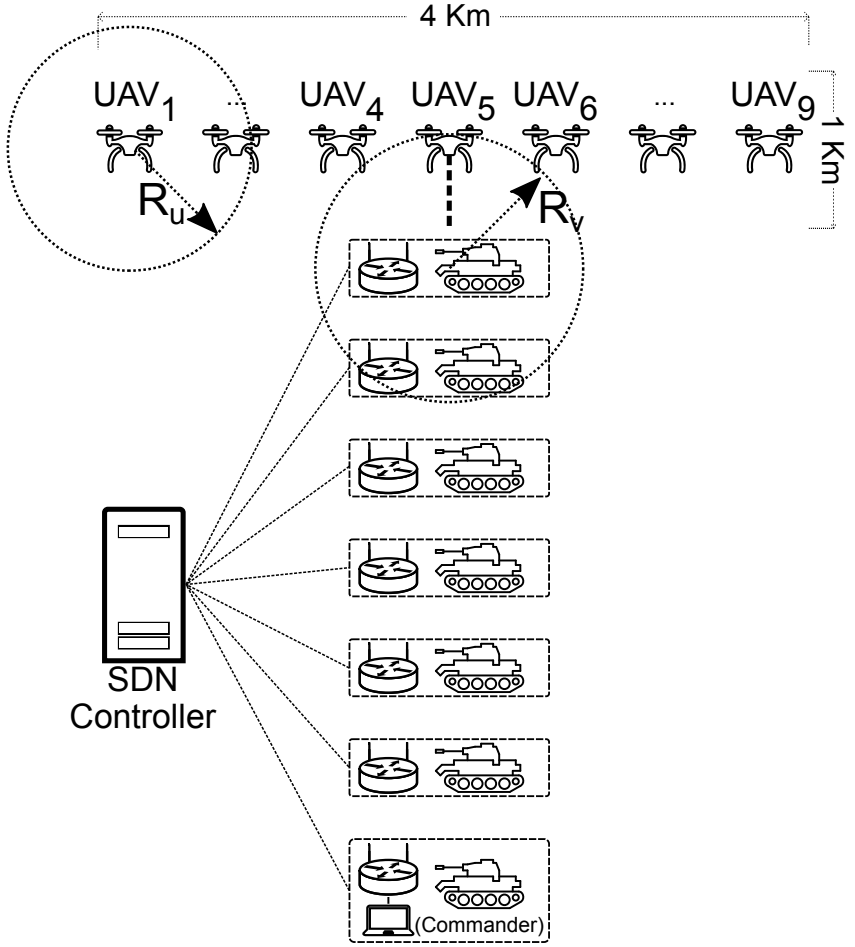
The following experiments focus on Scenario 1, described in Section 4.1 and depicted on the Figure 3.1, due to the bottleneck represented by the unique connection between the network composed of UAVs and the network of the ground vehicles. A software defined network was used to connect the ground vehicles in the platoon, according to the proposal described in Chapter 4.1. The connections among ground vehicles was considered as links with speeds of 100 Mbps between them and discarding interference that may occur in wireless connections links.

The ground vehicles are arranged in a row and the network that connects them form a kind of bus. UAVs connect to ground vehicle closest to the UAV-squad. There is a mesh network among UAVs to enable farthest UAVs connect to the ground vehicles' network. For example, in Figure 5.1 the UAV_6 can not connect directly to the ground vehicles' network. Thereby, the mesh network among UAVs allow the data packets to be sent from UAV_6 to reach the destination network using UAV_5 as a kind or "relay" node. The UAV_5 has a wireless link to the destination network and therefore it should route the packets sent by UAV_6 to the ground vehicles network.

In the Scenario 2, depicted in Figure 3.2, there is no need for the UAVs to relay the data through the mesh network, as in Scenario 1. Since UAVs can directly connect to the ground vehicles, all devices are on the same network. The SDN controller has more information about the network state, enabling better management and faster response to events, such as ping-pong effects and links overload.

In the simulated scenario, the experiments were performed with multiple simultaneous video streams, ranging from a one video stream at a time and increasing the number of streams up to nine videos streams being transmitted simultaneously. The used frame rate was 30 fps, the video stream length was 60 seconds with a codec H.264 (AGENCY, 2009). The UAVs acting as video stream server were uniformly and randomly selected. At the extreme case of nine video streams being transmitted, all UAVs in the scenario send their data. The software displaying the generated video streams is located in the farthest vehicle of UAVs squad (The vehicle at the bottom of the Figure 5.1). The node selection for the video playback is in accordance with the instructions of the army, considering that the mission commander will have access to video and that it usually takes one of the last vehicles of the convoy. Moreover, this choice represents the worst case for data transmission because the data flow should travel the entire ground vehicles network, before it

Figure 5.1: Ground vehicles and UAV disposition simulated in scenario 1.



Source: The Author, 2018

reaches the destination host. Table 5.1 summarizes the main simulation parameters.

Table 5.1: Parameters used in the simulations

Parameter	**Used Value**
Number of UAVs	9
Number of ground vehicles	9
Total simulation area	4 Km x 4 Km
UAV moving area	4 Km x 1 Km
UAV communication radius (R_u)	360m
Ground communication radius (R_v)	650m
Mobility models	Random Walk, Random Waypoint
Video size	960x540 pixels
Video frame rate	30 fps
Video codec	H.264
Video length	60 seconds
Number of runs per number of streams being served	33 runs

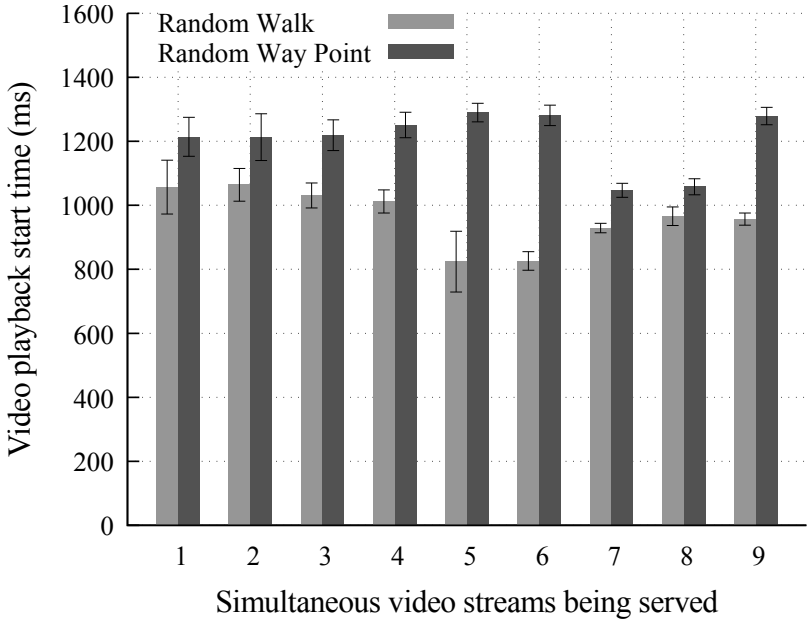
Source: The Author, 2018

5.2.3 Results Presentation and Discussion

Although the results in extreme cases present MOS notes below the expectations, positive facts can be listed. In all of the performed experiments, the video playback start time measurements hold at an acceptable value for a video surveillance application and its value is negligible. The values are in the range from 800 milliseconds to 1300 milliseconds as depicted in Figure 5.2. Regardless of video playback start time influence in the final MOS note, *i.e.,* the value slightly grater then 4 for the cases with up to 4 simultaneous streams, this value is still acceptable for the majority of video applications, both for entertainment and surveillance applications.

Revisiting the requirements presented in Chapter 3, Section 3.1, particularly examining the provided example in which a platoon moves at a speed of 60Km/h against enemies coming from the opposite direction at the same speed, the use of the UAV squad providing clearance in the area 1Km ahead sufficiently addresses the needs. The video playback starts within a acceptance time interval, even in the worst case, as can be observed in Figure 5.2.

Figure 5.2: Video playback start time measured in scenario 1.

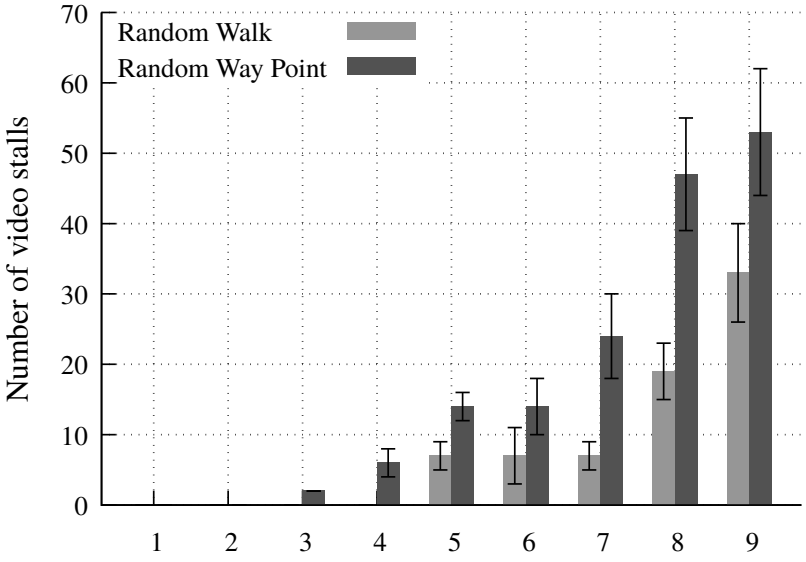


Source: The Author, 2018

The number of video interruptions, or video stalls, was strongly influenced by the number of simultaneous video flows. The various streams competing for available bandwidth generated resource contention, especially in the context of the wireless network environment.

The simplest mobility model, *i.e.,* the Random Walk, resulted in more stable wireless associations, both among the UAVs and between them and the access points. Stable and lasting wireless associations produce better results in the video assessments because they allow a higher data rate transfer, avoiding the starvation of the player buffer (*i.e., rebuffering events*). As a consequence, the number of video stalls and the total stall length depicted in Figure 5.3 and Figure 5.4 for the Random Walk mobility model present slightly better values compared to Random Waypoint mobility model. Even in the simulations using the Random Waypoint mobility model, the total video stall time is satisfactory up to eight simultaneous video streams. Still considering the example presented in Chapter 3, Section 3.1, the stall lengths are also within an acceptable interval, being in the worst case 10s, which is still a short time considered the time to reach the line of contact.

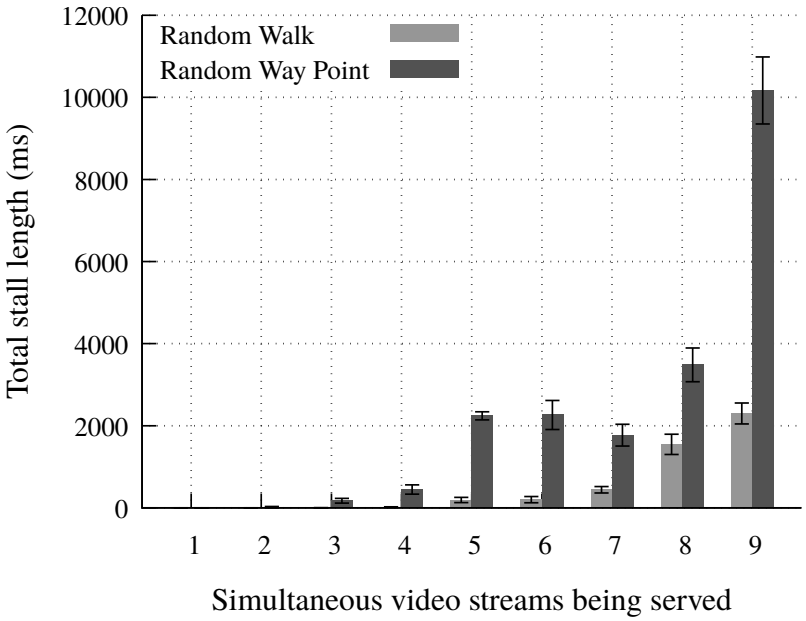
Figure 5.3: Number of interruptions measured in scenario 1.



Source: The Author, 2018

From the user perspective, according to the predicted MOS values in Figure 5.5, it was observed good results using up to 4 simultaneous video streams with the Random Walk mobility model, which represents approximately a half of the UAV platoon sending video traffic simultaneously. It is noteworthy to observe that the video frame rate used in the simulation ($\sim$ 30 fps). Normally, video surveillance does not require this higher frame rate, and with smaller video frame rate the streams should be lesser resource intensive. Thus a better user experience is expected, resulting in a higher MOS. The same applies to the video resolutions selected for the simulations. The selected video size is in the range

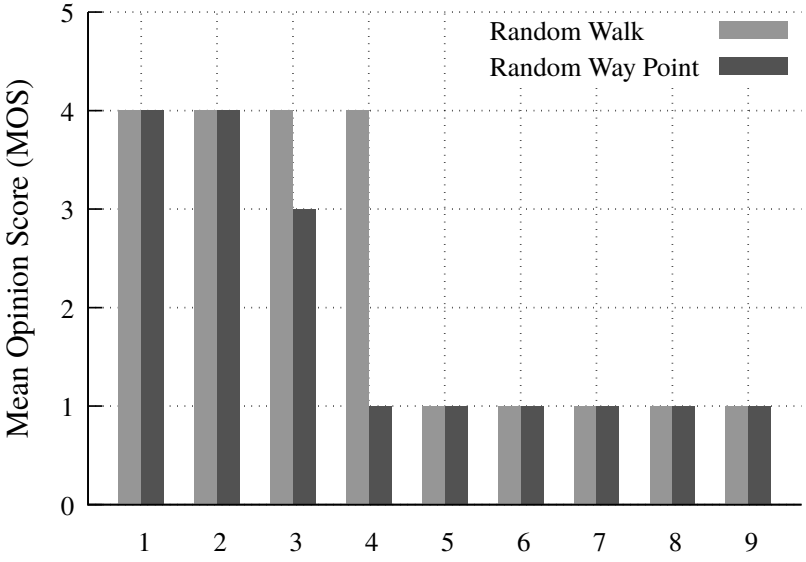
Figure 5.4: Total duration of interruptions in scenario 1.



Source: The Author, 2018

of values used in the military context. However, many military applications make use of smaller video size. Finally the case with all UAVs generating video traffic was being simulated. However, it is expected that a smaller number of video transmissions occur simultaneously in surveillance and reconnaissance applications.

Figure 5.5: Predicted Mean Opinion Score (MOS) in scenario 1.



Source: The Author, 2018

5.3 Inter-Partition Experiments

In this subsection, the experiments related to the architecture proposed in Section 4.2 are described. The control plane, which is in charge of the network logic and instructs the forwarding devices about the actions that should be applied in incoming packets, is an essential element introduced by the SDN paradigm. Therefore, the existence of at least one SDN controller per network partition is crucial. The following experiments explore the election of nodes to act as the SDN controller in each network partition of the TEN.

5.3.1 Selected Evaluation Metrics

The election of the node to act as the master node as described in Section 4.2 is an important process. If a network partition lost its master node, and consequently its SDN controller, the network configuration cannot be updated. The leader election algorithm was not defined in the proposed architecture. For a proof of concept, we have implemented a distributed leader election algorithm. Using this distributed algorithm the nodes in TEN can negotiate which node is more suitable to be the master node and act as the SDN controller. The node selection will be directly influenced by the leader election algorithm chosen.

The selected metrics to access the election process will change according to the chosen algorithm. Three metrics were selected to estimate the performance of the implemented algorithm. These metrics are:

- *Number of iterations*: This metric corresponds to the maximum number of iteration of the algorithm on a node until a leader is elected on the network partition. A node executing the algorithm can have three possible states: UNCLUSTERED, CLUSTERED and CLUSTER_HEAD. Depending on the node state, each iteration means getting information from the neighbor nodes, which can be associated with energy consumption.
- *Election time*: The required time executing the leader election algorithm for the nodes to agree on the node to be elected as the master node. The computed time corresponds to the interval from the SDN controller becomes unreachable in the network segment until the first elected node announces himself as a master node.

- *Data loss during the election*: This metric is the amount of data (in bytes) lost when a leader election process occurs. To check the total data loss a data transmission was started right before the leader election process. The data transmission traverses an SDN controlled device and the device was configured to discard all data when the SDN Controller is unreachable.

5.3.2 Simulated Scenario and Parameters

This use case scenario deals with communication among heterogeneous nodes in the TEN spread on different network partitions. Observing Figure 3.3, the situation considered here is the transmission of data acquired by WSN nodes located in Network Partition I to a C2 application running in the vehicle in which the operation commander is following how the battle is unfolding, located in Network Partition III (rightmost network partition in the figure).

The WSN is initially isolated from the others partitions, covering a distant or difficult to access area, collecting data about enemies' movements. The sensor nodes store the collected data in a temporary memory or forward them to a particular sensor node which is capable of store larger amount of data.

An UAV performing a task nearby is assigned to collect the WSN data. When the UAV establishes a connection with WSN nodes, the leader election chooses a node to run the SDN Controller and DTN Orchestrator instances. The SDN Controller updates the topology data concerning the actual network partition. Flow rules applied to forwarding devices classify the data being transmitted among WSN nodes and the UAV. The WSN acquired data belong to an elastic application, thus DTN support is used. The SDN Controller notifies the DTN Orchestrator about the data being received and optimizes the data transfer among DTN nodes and the UAV by selecting links capable of proceeding with the data transfer. These links connect the UAV to other UAVs and/or ground military vehicles that can act as data forwarding nodes inside a given partition and between nearby partitions that are joined for a while.

As the UAV continues its movement around the area, a connection with another partition eventually occurs. Again the leader election algorithm selects a node to host the SDN Controller and DTN Orchestrator instances. The DTN Orchestrator is aware of data acquired from WSN and interacts with the SDN Controller to choose the best route to forward the data stored by the DTN Layer toward the destination node. The SDN Con-

troller install flow rules in the forwarding devices to allow the data to be transmitted via Partition II until it reaches the destination in Partition III. In a high dynamic network environment, such a network composed of UAV, ground vehicles and WSN monitoring and exploring a wide area, the leader election is a frequent and essential task. The presence of a master node in each network segment is required for the correct operation of the software-defined network. The results of the experiments regarding the leader election algorithm are addressed in the next subsection.

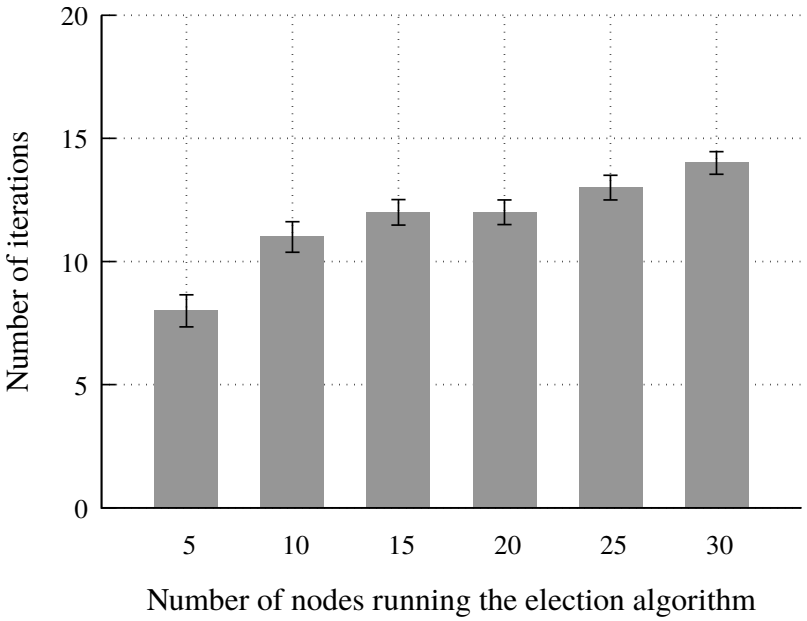
5.3.3 Results Presentation and Discussion

It is important to notice that the setup proposed in Section 4.2 does not specify the leader algorithm to be used for the election of the master node. The network administrator should choose the algorithm that best fits the requirements of the scenario in witch the TEN is running. Therefore, the experiment results may vary depending on the chosen leader election algorithm. The following results were obtained using the algorithm described in Section 4.3, and depicted in Figure 4.5.

The implemented algorithm is distributed, so nodes are expected to present slightly different values for the observed measurements. The algorithm finishes the election of the master node in few iterations. As the node only interacts with a limited set of nodes (neighbor nodes), the number of nodes in the network partition does not have a strong influence over the number of iterations of the algorithm in each node. The Figure 5.6 shows the average maximum number of iterations of a node to finish the election process, for scenarios ranging from 5 to 30 nodes. In other words, these values describe the "worst" performance of a node to finish the election in their network partition.

The time each node takes to finish the leader election algorithm can also differ; it is due to a waiting time selected randomly by each node. If a node is in the course of a task, for example communicating to its neighbors to count the number of followers it will gain in the case it becomes a master node (*i.e.,* promoting himself a CLUSTER_HEAD), the node should finish the algorithm only after finishing the task under execution. As the number of nodes in the scenario increases and the dimensions of the area where nodes are deployed remains the same, more nodes lie in the transmission range of other nodes. Thus, the degree of nodes increase and the node should exchange messages with a greater number of neighboring nodes, leading to an increase in the total time. This behavior can be observed in Figure 5.7 where it is possible to notice a slight increase in the total

Figure 5.6: Maximum number of iterations for choosing a master node.



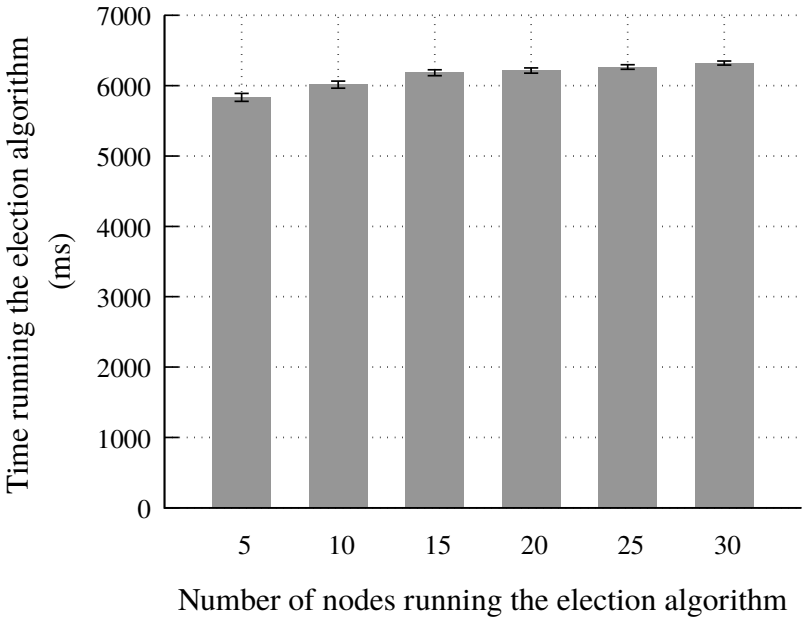
Source: The Author, 2018

time running the election algorithm as the number of nodes in the network partition also increases. Therefore, it is plausible to infer that the algorithm easily scales for scenarios with a higher number of nodes in the network partition.

An important point to be considered is the data loss during the election of the new master node (*i.e.,* the amount of data loss while an SDN controller cannot be reached in the network partition). Taking as an example a surveillance video application running on the tactical edge, this data loss is perceived as video transmission interruptions. In the scenario with 10 nodes running the algorithm, which is the worst value observed, the average data loss is approximately 6750 KBytes sending 30 data flows at a rate of 10 Mbits/s, transferring 1.19 MBytes per flow and totalizing 35.8 MBytes. Following standards from the Agency (2009) and results from empirical evaluations conducted by Nightingale et al. (2016), the observed values are likely to produce freezes of 1s or less in a video surveillance transmission, in a TEN environment with similar characteristics.

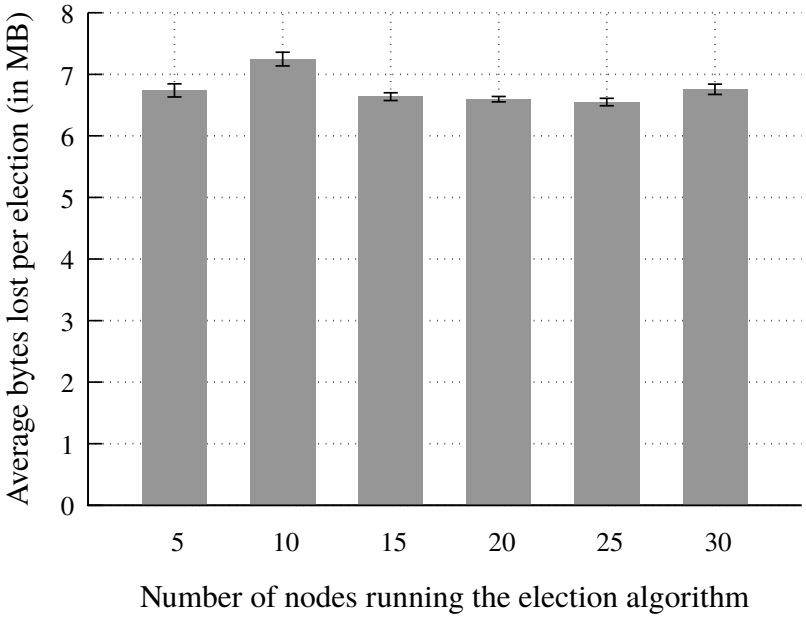
For the experiments measuring the data loss during the master node election process, a isolated network was created. Therefore, the traffic regarding the election process does not interfere in the measurements of data loss, and the data was sent using the UDP transmission protocol. Also, it is important to mention that a temporary master node is elected before all nodes end up the algorithm. Thus, Figure 5.7 does not represent the total time the network operated without a controller, but the total time for the election of a final master node. In all results in this section, bars represent average values, while error

Figure 5.7: Time needed for nodes to elect a master node.



Source: The Author, 2018

Figure 5.8: Average data loss during the election process.



Source: The Author, 2018

bars represent confidence intervals for 95% of confidence from 200 runs for each network scenario generated randomly.

6 CONCLUSION

The BN is a challenging communication environment in which very different concerns have to be considered. Attempts to address these concerns have involved exploring approaches based on SDN or DTN. Despite the features that these two paradigms provide, they cannot address all these concerns on an individual basis. Moreover, few proposals have examined them in the last-mile TEN, which is the most challenging part of the BN due to its high mobility and constrained resources.

Observing advances in both the SDN and DTN paradigms, a promising approach to combine the best of these to address last-mile TEN is proposed in this work. The architecture shows features capable of tackling this difficult operational scenario. The proposed architecture was evaluated in an emulator for SDN. Additionally, it was demonstrated an application of SDN observing the quality of experience in video streaming in the context of military mobile networks. Objective measurements were collected using a modified version of a popular media player. The collected measurements were used to predict the subjective QoE indicator: the Mean Opinion Score (MOS). The results were promising and demonstrate that programmable networks can be successfully applied to heterogeneous networks, disruptive networks, and networks with opportunistic connections.

The proposed approach aims at linking a heterogeneous network composed of UAVs and ground vehicles focusing on surveillance and recognition applications in the military operations context. Although, it should be stressed that other domains may benefit from this architecture, such as networks designed to assist disaster relief operations, which face similar harsh operational conditions.

Future work should address the evaluation of the DTN features, that was not considered in the present study. The implementation of DTN protocols that could use the global knowledge of the SDN controller about the network to optimize the transmission and routing of packets on opportunistic links should also be considered in future works. Position-based routing can be used to optimize data routing along the partitions. Finally, the proposed architecture and the leader election algorithm could be deployed in real UAVs networks, for more accurate assessment, and testing of real scenarios.

REFERENCES

ABUTEIR, R. M.; FLADENMULLER, A.; FOURMAUX, O. SDN based architecture to improve video streaming in home networks. In: **International Conference on Advanced Information Networking and Applications (AINA)**. Crans-Montana: IEEE, 2016. p. 220–226.

AGENCY, N. S. **Standardisation Agreement: NATO digital motion imagery standard**. 3. ed. Brussels: [s.n.], 2009. 73 p.

AMIN, R. et al. Design considerations in applying disruption tolerant networking to tactical edge networks. **IEEE Communications Magazine**, v. 53, n. 10, p. 32–38, October 2015.

ARMY, D. of the. **Field Manual No. 34-02-1: Tactics, Techniques, and Procedures for Reconnaissance and Surveillance and Intelligence Support to Counterreconnaissance**. Washington, DC: [s.n.], 1991. 162 p.

ARMY, D. of the. **Field Manual No. 3-04.155: Army Unmanned Aircraft System Operations**. Washington, DC: [s.n.], 2006. 183 p.

ARMY, D. of the. **Field Manual No. 3-55.93: Long Range Unit Surveillance Operations**. Washington, DC: [s.n.], 2009. 386 p.

BAI, F.; HELMY, A. A survey of mobility models in wireless adhoc networks. In: ____. **Wireless Ad Hoc and Sensor Networks**. [S.l.]: Springer International Publishing, 2004. p. 1–30.

BEKMEZCI, İ.; SAHINGOZ, O. K.; TEMEL, Ş. Flying Ad-Hoc networks (FANETs): A survey. **Ad Hoc Networks**, v. 11, n. 3, p. 1254–1270, May 2013.

BELLARD, F. **FFMpeg**. 2000. Retrieved 19 July, 2018. Available from Internet: <http://ffmpeg.org/>.

BLOEBAUM, T. H. et al. Recommendations for realizing SOAP publish/subscribe in tactical networks. In: **International Conference on Military Communications and Information Systems (ICMCIS)**. Brussels: IEEE, 2016. p. 1–8.

BRANNSTEN, M. R. et al. Toward federated mission networking in the tactical domain. **IEEE Communications Magazine**, v. 53, n. 10, p. 52–58, October 2015.

CASAS, P.; SCHATZ, R.; HOSSFELD, T. Monitoring youtube QoE: Is your mobile network delivering the right experience to your customers? In: **IEEE Wireless Communications and Networking Conference (WCNC)**. Shanghai: IEEE, 2013. p. 1609–1614.

CHAN, H.; PERRIG, A. ACE: An emergent algorithm for highly uniform cluster formation. In: ____. **Wireless Sensor Networks**. [S.l.]: Springer Berlin Heidelberg, 2004. p. 154–171.

DALAMAGKIDIS, K. Classification of UAVs. In: ____. **Handbook of Unmanned Aerial Vehicles**. [S.l.]: Springer Netherlands, 2015. p. 83–91.

FALL, K.; FARRELL, S. DTN: an architectural retrospective. **IEEE Journal on Selected Areas in Communications**, v. 26, n. 5, p. 828–836, June 2008.

FERNANDES, R.; HIEB, M. R.; COSTA, P. Levels of autonomy: Command and control of hybrid forces. In: **C2 in a Complex Connected Battlespace (ICCRTS)**. London: IC2I, 2016. p. 1–16.

FILHO, R. I. T. da C. et al. Network fortune cookie: Using network measurements to predict video streaming performance and QoE. In: **IEEE Global Communications Conference (GLOBECOM)**. Washington, DC: IEEE, 2016. p. 1–6.

FONTES, R. R. et al. Mininet-WiFi: Emulating software-defined wireless networks. In: **International Conference on Network and Service Management (CNSM)**. Barcelona: IEEE, 2015. p. 384–389.

GUPTA, L.; JAIN, R.; VASZKUN, G. Survey of important issues in UAV communication networks. **IEEE Communications Surveys and Tutorials**, v. 18, n. 2, p. 1123–1152, April 2016.

HOSSFELD, T. et al. Initial delay vs. interruptions: Between the devil and the deep blue sea. In: **International Workshop on Quality of Multimedia Experience (QoMEX)**. Melbourne: IEEE, 2012. p. 1–6.

ITU-T. Recommendations of the ITU, **Recommendation P.800.1: Mean opinion score (MOS) terminology**. 2016.

JOSEPH, V.; BORST, S.; REIMAN, M. I. Optimal rate allocation for video streaming in wireless networks with user dynamics. **IEEE/ACM Transactions on Networking**, v. 24, n. 2, p. 820–835, April 2016.

JULURI, P.; TAMARAPALLI, V.; MEDHI, D. Measurement of quality of experience of video-on-demand services: A survey. **IEEE Communications Surveys and Tutorials**, v. 18, n. 1, p. 401–418, January 2016.

KIM, H.; FEAMSTER, N. Improving network management with software defined networking. **IEEE Communications Magazine**, v. 51, n. 2, p. 114–119, February 2013.

KLEINROUWELER, J. W.; CABRERO, S.; CESAR, P. Delivering stable high-quality video: An SDN architecture with DASH assisting network elements. In: **International Conference on Multimedia Systems (MMSys)**. Klagenfurt, Austria: ACM, 2016. p. 1–10.

KREUTZ, D. et al. Software-defined networking: A comprehensive survey. **Proceedings of the IEEE**, v. 103, n. 1, p. 14–76, January 2015.

KUROSE, J. F.; ROSS, K. W. **Computer Networking: A Top-down Approach**. 6th. ed. [S.l.]: Pearson, 2013. 588–593 p. (Always learning).

L-3 WESCAM. **WESCAM's MX-15: EO IR ISR Systems**. 2015. Retrieved 19 July, 2018. Available from Internet: <http://www.wescam.com/wp-content/uploads/PDS-MX-15-April2015.pdf>.

LAMPORT, L. The part-time parliament. **ACM Trans. Comput. Syst.**, v. 16, n. 2, p. 133–169, may 1998.

LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop. In: **ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets)**. Monterey, CA: ACM, 2010. p. 1–6.

LIU, Y. et al. Choquet integral based QoS-to-QoE mapping for mobile vod applications. In: **International Symposium on Quality of Service (IWQoS)**. Beijing: IEEE, 2016. p. 1–6.

MCKEOWN, N. et al. OpenFlow: Enabling innovation in campus networks. **ACM SIGCOMM Computer Communication Review**, v. 38, n. 2, p. 69–74, March 2008.

MOK, R. K. P.; CHAN, E. W. W.; CHANG, R. K. C. Measuring the quality of experience of HTTP video streaming. In: **IFIP/IEEE International Symposium on Integrated Network Management and Workshops (IM)**. Dublin: IEEE, 2011. p. 485–492.

MOTLAGH, N. H.; BAGAA, M.; TALEB, T. UAV-Based IoT platform: A crowd surveillance use case. **IEEE Communications Magazine**, v. 55, n. 2, p. 128–134, February 2017.

NAM, H. et al. Towards QoE-aware video streaming using SDN. In: **IEEE Global Communications Conference (GLOBECOM)**. Austin, TX: IEEE, 2014. p. 1317–1322.

NIGHTINGALE, J. et al. Reliable full motion video services in disadvantaged tactical radio networks. In: **International Conference on Military Communications and Information Systems (ICMCIS)**. Brussels: IEEE, 2016. p. 1–9.

NOBRE, J. et al. Toward software-defined battlefield networking. **IEEE Communications Magazine**, v. 54, n. 10, p. 152–157, October 2016.

NUNES, B. A. A. et al. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. **IEEE Communications Surveys and Tutorials**, v. 16, n. 3, p. 1617–1634, July 2014.

ORFANUS, D.; FREITAS, E. P. d.; ELIASSEN, F. Self-organization as a supporting paradigm for military UAV relay networks. **IEEE Communications Letters**, v. 20, n. 4, p. 804–807, April 2016.

Ryu Community. **Ryu SDN Framework**. 2014. Retrieved 19 July, 2018. Available from Internet: <http://osrg.github.io/ryu>.

SEUFERT, M. et al. A survey on quality of experience of HTTP adaptive streaming. **IEEE Communications Surveys and Tutorials**, v. 17, n. 1, p. 469–492, January 2015.

SINGH, S. K.; DAS, T.; JUKAN, A. A survey on internet multipath routing and provisioning. **IEEE Communications Surveys and Tutorials**, v. 17, n. 4, p. 2157–2175, October 2015.

Stark Aerospace Inc. **POP200 Specs**. 2015. Retrieved 19 July, 2018. Available from Internet: <http://starkaerospace.com/products-services/sensors/pop200/>.

TORTONESI, M. et al. Enabling the deployment of COTS applications in tactical edge networks. **IEEE Communications Magazine**, v. 51, n. 10, p. 66–73, October 2013.

TORTONESI, M. et al. Multiple-UAV coordination and communications in tactical edge networks. **IEEE Communications Magazine**, v. 50, n. 10, p. 48–55, October 2012.

WICKBOLDT, J. et al. Software-defined networking: Management requirements and challenges. **IEEE Communications Magazine**, v. 53, n. 1, p. 278–285, January 2015.

APPENDIX A — PUBLISHED JOURNAL (IEEE COMM. MAGAZINE)

Iulisloi Zacarias, Luciano P. Gaspary, Andersonn Kohl, Ricardo Q. A. Fernandes, Jorgito M. Stocchero, Edison P. de Freitas. **Combining Software-Defined and Delay-Tolerant Approaches in Last-Mile Tactical Edge Networking**. In IEEE Communications Magazine, Volume 55, Number 10, pp. 22–29, October 2017. DOI: 10.1109/MCOM.2017.1700239.

- **Title**: Combining Software-Defined and Delay-Tolerant Approaches in Last-Mile Tactical Edge Networking.
- **Abstract**: Network-centric warfare is a no-way-back trend in modern military operations. The application of this concept ranges from upper-level decision making echelons to troop guidance on the battlefield, and many studies have been carried out in this area. However, most of these are concerned with either the higher-level strategic networks, that is, the networks linking the higher echelons with abundant resources, satellite communications, or even a whole network infrastructure, or high-end TEN, representing resource-rich troops in the field, with military aircraft, battleships, or ground vehicles equipped with powerful wireless communication devices and (almost) unrestricted energy resources for communication. However, these studies fail to take into account the "last-mile TEN," which comprises resource constrained communication devices carried by troopers, equipping sensor nodes deployed in the field or small unmanned aerial vehicles. In an attempt to fill this gap in the studies on battlefield networking, this article seeks to combine software-defined and delay-tolerant approaches to support the diverse range of strict requirements for applications in the last-mile TEN.
- **Status**: Published.
- **Qualis / CAPES**: A1.
- **Journal**: IEEE Communications Magazine.
- **Date**: October 2017.
- **Local**: New York.
- **URL**: <https://www.comsoc.org/commag>.
- **DOI**: <https://doi.org/10.1109/MCOM.2017.1700239>.

Combining Software-Defined and Delay-Tolerant Approaches in Last-Mile Tactical Edge Networking

Iulisloi Zacarias, Luciano P. Gaspary, Andersonn Kohl, Ricardo Q. A. Fernandes, Jorgito M. Stocchero, and Edison P. de Freitas

Observing a gap in the literature about handling problems arising from the last mile TEN, and taking into account the resource-constrained devices used by troops in the field, the authors propose the joint exploration of SDN and DTN concepts to address the needs of tactical-operational networks.

ABSTRACT

Network-centric warfare is a no-way-back trend in modern military operations. The application of this concept ranges from upper-level decision making echelons to troop guidance on the battlefield, and many studies have been carried out in this area. However, most of these are concerned with either the higher-level strategic networks, that is, the networks linking the higher echelons with abundant resources, satellite communications, or even a whole network infrastructure, or high-end TEN, representing resource-rich troops in the field, with military aircraft, battleships, or ground vehicles equipped with powerful wireless communication devices and (almost) unrestricted energy resources for communication. However, these studies fail to take into account the "last-mile TEN," which comprises resource constrained communication devices carried by troopers, equipping sensor nodes deployed in the field or small unmanned aerial vehicles. In an attempt to fill this gap in the studies on battlefield networking, this article seeks to combine software-defined and delay-tolerant approaches to support the diverse range of strict requirements for applications in the last-mile TEN.

INTRODUCTION

Currently, military operations are controlled by interconnected units, from a strategic level (i.e., the higher-level centers for decision making) to the tactical-operational level involving the units on the battlefield. Battlefield networks (BNs) establish connections among these different nodes through various communication technologies, from short-range wireless to satellite links. At a tactical-operational level, the involved networks are referred to as tactical edge networks (TENs) [1], which vary depending on the capabilities of the nodes that form them. At one end, there are networks made up of high-end nodes, which are resource-rich in regard to communications technologies, high bandwidth, and (almost) no restrictions on energy consumption [2]. At the other end, networks of resource-constrained nodes, also called low-end nodes, cover the following: the last-mile BN consisting of wireless sensor nodes deployed in the field, radios carried by special force units, and small unmanned aerial vehicles (UAV) for image acquisition, among other areas [3]. These different networks support a variety of applications, ranging from elastic ones, such as file and text message transfer, to non-elastic real-time applications, such as video streaming. Applications are affected by different problems and have to meet different requirements depending on their degree of elasticity. To tackle these problems, the overall communication system that forms a part of the battlefield scenario can employ different concepts, such as software-defined networks (SDNs) and delay/disruption-tolerant networks (DTNs).

SDN is a promising paradigm, which provides flexibility to network management by separating the network infrastructure into distinct planes [4]. Each plane can be programmed to meet particular application requirements; for example, to support legacy or commercial off-the-shelf (COTS) applications without having to make intensive changes to the behavior of an application. These network adjustments may range from quality of service (QoS) configurations to the deployment of new protocols and policy enforcement in a running network. The application programming interface (API) provided by an SDN controller enables the technology to be integrated with high-level non-network systems. These systems can deploy security, media, or vendor-specific features using the SDN controller API regardless of the network protocols. The use of SDN in the BN scenario was previously explored in [2]. It aimed to improve communications between devices in a dynamic and heterogeneous military environment. The authors adopted a high-level approach to apply SDN concepts to military networks where the communication nodes are resource-rich (e.g., battleships and airplanes) using links provided by satellites, among other methods. Their main concern was to strengthen a military network at a strategic level through the flexibility provided by SDN.

DTN is a network architecture approach that aims to address the problem of lack of continuous connectivity in dynamic networks; that is, there are extended periods of link unavailability [5]. TENs often employ relay nodes (RNs) to forward data from source to destination nodes and thus acquire a larger coverage area (including rugged terrain and harsh environments), where they are usually deployed. However, RNs may be unavailable as a result of energy depletion, tech-

Digital Object Identifier:
10.1109/MCOM.2017.1700239

Iulisloi Zacarias, Luciano P. Gaspary, and Edison P. de Freitas are with the Federal University of Rio Grande do Sul; Andersonn Kohl, Ricardo Q. A. Fernandes, and Jorgito M. Stocchero are with the Brazilian Army.

nical failures, or attacks by enemy forces. In addition, due to the high mobility of some nodes (e.g., UAVs and ground vehicles), the data routes often have to be recalculated, and sometimes end-to-end connection between a given source and destination nodes does not exist, or only exists for a very short time. Long link outages lead to loss of connectivity, transmission timeouts, and routing failures. DTN tackles this problem by temporarily storing the data in RNs. These data are then forwarded to the destination when opportunistic connections show up. As outlined in [6], DTN is a technology widely explored in military networks due to their need to augment the coverage of communication networks by using aerial high-capacity backbone systems comprising manned and unmanned aircraft. Links between aircraft display more frequent intervals of outage than ground or satellite networks. However, like other approaches such as SDN usage in BNs, the DTN design described in [6] is also restricted to networks at the strategic level, which rely on resource-rich nodes, and does not cover the last-mile TEN.

Observing this gap in the literature about handling problems arising from the last-mile TEN, and taking into account the resource-constrained devices used by troops in the field, this work proposes the joint exploration of SDN and DTN concepts to address the needs of these tactical-operational networks. The proposed approach benefits from the programmability offered by SDN and the ability of DTN to handle link outages. These features are used in the context of network-centric military operations in the field, and primarily to fulfill the hard-to-meet requirements of low-end nodes. Thus, the main contribution of this article is to define an architectural solution to support the last-mile TEN by adapting and combining SDN and DTN technology.

Reviewing SDN and DTN Concepts

The SDN paradigm was initially planned as a form of technology that could be applied in wired networks. It was rapidly adopted by the research community and industry, which extended it for other types of (wireless) networks, such as satellite networks [6] and wireless sensor networks (WSNs) [7]. The paradigm proposes the separation of the data forwarding plane (or data plane) from the network control logic (control plane). The network equipment in the forwarding plane consists of simple packet forwarding devices, while the control logic of the network is implemented in a centralized entity, called the SDN controller [8]. It is noteworthy that centralized control does not imply having a physically centralized entity. Indeed, single points of failure (SPOFs) should be avoided, and redundancy in the control plane is highly desirable [4].

The clear separation between the network planes and the abstraction of the network control logic from the distributed hardware makes it easier to implement new network management directives. An external application can interact with the network through the SDN controller. Examples of external applications are load balancers, network orchestration frameworks, and business functions. The SDN architecture allows interaction between planes by defining protocol-neutral interfaces. The southbound interface allows the control plane to exchange data with the forwarding plane. The current well-known de facto standard protocol adopted by industry and used in the southbound interface is OpenFlow [7]. Communication between external applications and the control plane occurs through the northbound interface. External applications use this interface to communicate their requirements to the SDN controller or to get information about the overall state of the network. The SDN controller translates the high-level specifications from the application plane through a northbound API to low-level specifications and applies them to the data plane by means of the southbound API [4].

DTNs were originally designed to cope with problems of an interplanetary Internet (IPN). This environment is subject to problems such as large round-trip times (RTTs), limited bandwidth, errors in communication links, possible link disruptions for long intervals, and intermittent connectivity. Researchers also recommend using DTN solutions in satellite applications because satellite links share some of the difficulties found in IPN. Despite the completely different scenario, terrestrial applications also make use of DTN solutions. Examples of terrestrial applications that make use of DTN solutions are BNs (including TENs), low-density and underwater wireless sensor networks, and UAV communication systems [6].

As a means of overcoming the problems caused by link disruption and large delays in data transmission, the DTN nodes act in a store-carry-forward way [5]. Link interruptions can range from a few seconds to several hours depending on the type of application employed. The DTN nodes must temporarily store a potentially large amount of data, and the intermediary DTN nodes are responsible for it until it can be forwarded to the next DTN-capable node or the destination node [6]. The Internet Research Task Force (IRTF) proposes an architecture and protocols that address data exchange by means of DTNs. An additional layer is placed between the transport and application layers to provide the special features required by DTNs [5].

Application Scenario

TENs are used to integrate communication devices in harsh battlefield environments. Scenarios in which TENs can be applied include, but are not limited to, borderline surveillance and the exploration of enemy-occupied areas. Access to the environment may be limited or restricted, and often there is no pre-existing communication infrastructure like cellular fourth/fifth generation (4G/5G) networks. The communication network in these scenarios is inherently very dynamic, due to the presence of mobile nodes such as small UAVs, ground vehicles, and the devices carried by soldiers [3, 9]. Small and energy-constrained devices are also combined with the TEN, such as piezoelectric sensors used to detect enemy forces [3, 10]. To benefit from the acquired data, there is a clear need for data exchange and forwarding. The importance of this data exchange and forwarding is to provide data to support commanders in their decisions, assisting in the management of the military operations [11].

The data exchanged between the TEN nodes are very diverse, ranging from simple and sin-

The SDN paradigm was initially planned as a form of technology that could be applied in wired networks. It was rapidly adopted by the research community and industry, which extended it for other types of (wireless) networks, such as satellite networks and wireless sensor networks.

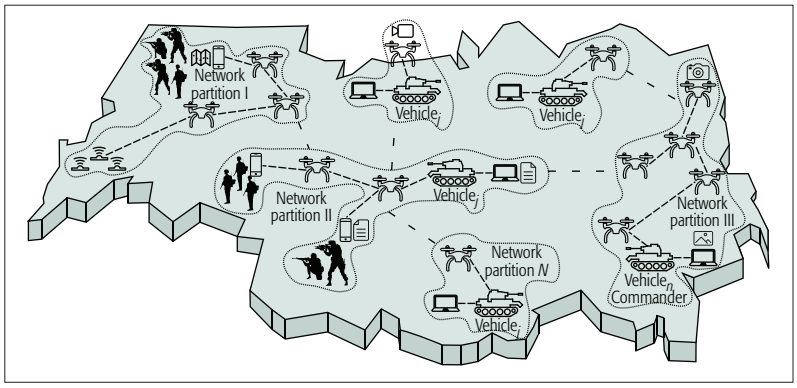


Figure 1. Last-mile TEN application scenario.

gle-valued data collected by small sensors [10] to a large amount of data generated by video capture devices (e.g., visible light and infrared cameras) and file transfers [3]. These data flows have different QoS requirements, depending on the applications [9]. For example, a file transfer application has flexible requirements related to delay, delay variation (jitter), and bandwidth. This is in contrast with a video surveillance application, which requires near-real-time transfers. A delayed video stream transmission of a surveillance application may affect the decision making and therefore hamper military operations. In this context, small UAVs can be used to fly ahead of the troops to gather information about hostile threats and enemy positioning.

Examples of the above mentioned applications in last-mile TEN are shown in Fig. 1. It is important to notice that there is not a single (big) network, but several small networks that occasionally and opportunistically interconnect with each other. For instance, a group of small UAVs launched from a ground military vehicle to video survey the terrain ahead forms its own network. This network may not be connected to the launching vehicle for a given period of time due to interference or range limits. However, this connection may eventually be restored, and a UAV closer to the ground vehicle can deliver the video that one of them has acquired. Another example, illustrated on the left side of Fig. 1, is a WSN for motion detection, which may be completely disconnected from the other nodes outside the WSN. These sensors acquire and process information about enemy movements in the area, and when a friendly node comes within range (e.g., a UAV), they supply the acquired data. Following this example, this UAV can, in turn, forward the data through a UAV-relay network to command and control (C2) applications running on mobile devices carried by troops in the field. As these applications have different requirements, they can be systematically classified so that a careful analysis can be conducted to address these specific requirements.

The dissemination of information is an essential part of C2 systems, since they directly affect the battlefield scenario [10, 11]. The use of information grids is a popular approach in network-centric warfare, because it allows situation awareness sharing by exchanging data between different applications and systems [1]. Legacy and COTS applications are also often used in C2 systems, and messages exchanged between them usually benefit from service-oriented architectures (SOAs) in TENs [12]. The SOA concept is a means of building distributed systems and a fundamental principle for federated military systems [9]. The SOA approach makes it easier to carry out data sharing between different applications and offers a flexible mechanism for reuse of already existing services [12]. TEN nodes can exchange SOA messages with each other by means of SOA-based software. For example, a C2 application handles maps running on handheld devices, which must subscribe to receive updated data from a WSN to be displayed. The use of intermediary nodes for caching the information and forwarding it over opportunistic wireless links causes delays to the communication. However, these delays do not represent a significant problem in elastic applications. For example, file transfers of photos and topographic maps among the TEN nodes belong to an elastic type of application, which can thus tolerate the store-carry-forward approach [13].

Applications that make use of a video to survey or secure an area should be able to meet stricter network requirements [14]. They are used when rapid decision making is required, which means that they are not as tolerant to network service degradation as the elastic applications. By nature, video transmission requires higher throughput, and the application cannot support a higher degree of network instability, like throughput limitations and significant latency variation (jitter). Bandwidth and latency variations result in inefficient video data transfers and low-quality video playout, with freezes and degradation of displayed images. The result is a loss of important details, which may lead to wrong decisions. The quality of images and video is crucial to the decision making process and cannot be disregarded [11]. In light of these requirements, the group of applications dealing with video transmission and continuous data flows having strict QoS network requirements can be classified as *non-elastic applications* [13].

Applications using bidirectional data transfers, such as video conferencing and voice over IP, are also valuable in networked warfare systems [14]. The network requirements imposed by these applications are very strict because the communication is an interactive process. The network should support a minimum throughput by allowing the transmission of video and/or audio and additionally meet strict requirements regarding latency variation (jitter). In these applications, end-to-end connections must exist between the points that wish to communicate, which means that temporary data storage at intermediate nodes using DTN solutions is inappropriate. Applications that have these restrictive features are classified as *interactive non-elastic applications*.

Table 1 summarizes the main characteristics of the aforementioned application classes, comparing their network requirements and suitability for making use of DTN intermediary storage.

In addition to network resource requirements, other important factors that need to be observed are the following:

- The deployment of new devices should meet strict requirements with regard to timing. Ease of configuration and fast device replacements are also important.

- In view of its high dynamicity, the network has to be prepared to promptly respond to changes in the topology, and even be able to provide self-healing properties.
- The different data flows require different QoS levels according to application classes.
- Security is a first-order requirement for military networks. Thus, the deployment of security mechanisms is essential.
- The nodes operating in TENs are often subject to energy constraints. Optimized routing algorithms and energy-aware protocols are desirable to extend the operating time of battery-powered devices.
- There are plenty of legacy applications already being deployed in military networks. The need for significant changes in these applications should be avoided. Existing applications should be included and seamlessly integrated.
- There is a need to avoid SPOFs in military systems, and for this reason link redundancy mechanisms are essential components in military networks [9].

The Proposed SDN-DTN Military Network Architecture

The last-mile TEN represents a challenging network environment that can be explained by the high mobility of the nodes, device heterogeneity, resource constraints, and specific (and hard to meet) requirements of military applications. This network connects applications with different requirements with regard to QoS, security, and reliability. When the concepts of SDN are merged with those provided by DTN, they offer the means to tackle this challenge. This section describes the objectives of the proposed architecture together with the best features offered by each approach.

By following the SDN paradigm, the forwarding devices are remotely managed by a centralized SDN controller, although this does not imply that it is a physically centralized entity. TENs must be robust and tolerate attacks so that they are able to continue operating even with damaged components. Redundancy mechanisms for the SDN controller are required to achieve the desired robustness.

The use of multiple controllers in the TEN overcomes the SPOF problem by allowing the network to continue operating when it is disrupted. By selecting a scenario in which small UAVs form a part of a TEN, each of them can take control of the network by running a controller instance. However, unnecessary control processing leads to waste of energy. On the other hand, a single instance of an active controller for the whole network will be unreachable for nodes located in different network segments. In this case, an SDN controller should be selected in each segment by running a leader election algorithm. When all the nodes return to the same segment, the leader election algorithm is run again and elects a single controller for the entire network. The elected controller might be one of the controllers previously active in one of the partitions, or even a new one running in a different node. In this case, the new controller is synchronized with the previous ones to ensure that consistency is kept with the already applied configurations. Stress should be laid on awareness of the overhead, or even the infeasibility of this election process due to the extreme dynamics of such networks. In these cases, the proposed architecture should be slightly changed and include, for instance, an out-of-band control channel. The assumption here is that the dynamics of the network is at a degree in which changes are expected to happen in minutes. This is plausible in these scenarios and ensures the feasibility of the approach.

Parameter	Application class		
	Elastic	Non-elastic	Interactive non-elastic
Network throughput usage	Low	High	Medium–high
Network latency tolerance	High	Low	Low
Jitter tolerance	High	Medium	Low
Packet loss tolerance	Low	Medium–high	Medium
DTN storage allowed	Yes	No	No
Example applications	File transfer, messaging, SOA-based applications	Video surveillance	Video and/or audio conference

Table 1. Applications classes and network requirements.

The node that runs an active instance of the SDN controller is the master in that network segment, and responsible for its control. The other nodes in the same segment are called slaves. Figure 2 shows the internal architecture of the master and slave nodes on its left and right sides, respectively. Instead of an SDN controller, the slave nodes run an SDN daemon component (on the right side in Fig. 2), which continuously surveys the current network segment to detect active SDN controllers. When an SDN controller is not found, the SDN daemon starts the leader election algorithm. When a slave node becomes a master (the new master elected node), the SDN daemon initializes the SDN controller and the DTN orchestrator components in that node (top right and top middle of the master node in Fig. 2). The other slave nodes in the TEN segment will be notified and have to update their configurations to connect to the recently elected controller.

There may be a huge number of messages exchanged between the master (SDN controller) and slave nodes, thus increasing energy consumption. To address this concern, the SDN daemon manages the migration of the SDN controller process among the nodes forming the current network segment. This management system aims to avoid battery depletion of the master node, thus increasing the autonomy of the small battery-powered devices that are often used in last-mile TENs. It should be noted that an instance of the SDN controller will run on a node when it is isolated from the remaining network. This controller will only manage this node's network interface. Also, it will watch for an opportunity to rejoin a network segment, and when this occurs, it will run the leader election algorithm and forward the temporally stored data that it may have.

The communication between the forwarding devices and the SDN controller uses the managed

The UAV on the top collects video to be sent to the commander's vehicle, located at the bottom. This task is valuable in reconnaissance missions, when a platoon arrives at a region of interest and UAVs are used to search for possible threats. The video transmission is a non-elastic application and must meet strict QoS requirements.

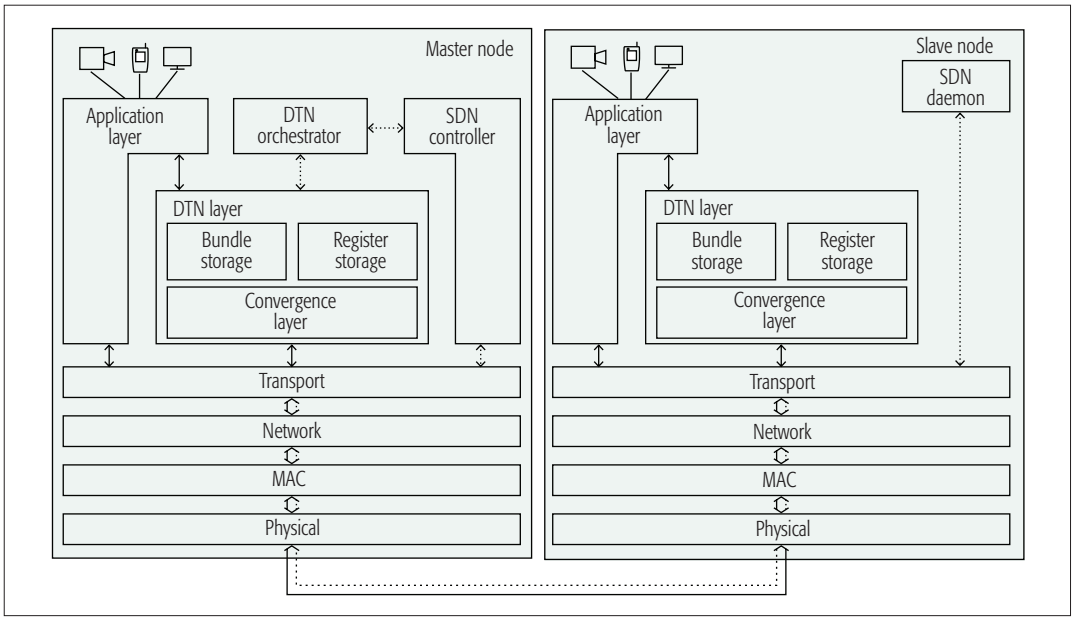


Figure 2. Combined SDN and DTN architecture.

network, and therefore an in-band controller connection [8]. The OpenFlow messages are routed to controlled devices through conventional network connections, and hence traverse the transport, network, and medium access control (MAC) layers, and finally the physical wireless interface (Fig. 2, at the bottom of both the master and slave nodes).

The DTN orchestrator coordinates the functionality of the DTN nodes. This module exchanges link information with the SDN controller in the master node, as can be observed in Fig. 2 (on the left side). The DTN orchestrator schedules data transmission between the DTN nodes on the basis of data collected by the SDN controller. The SDN controller has updated information and a "global view" of the network, that is, a global view of the network segment it is controlling. Data flows in TENs have different QoS and security requirements [2], and the SDN controller keeps track of them. The DTN orchestrator also gathers information from the DTN layer about the bundle storage state, buffer utilization, and DTN endpoints (register storage). The acquired information can be used to select a different DTN routing algorithm or forward the data so that it can be stored temporarily in another node, thus avoiding the overflow of the DTN buffer.

The DTN layer, represented in Fig. 2 (in the middle of both master and slave nodes), follows the concepts of current DTN implementations. The bundle storage component is used to store the data collected by devices and applications when the destination node is unreachable (e.g., the destination node is in another network segment or is temporarily turned off). The convergence layer links the bundle storage to the transport layer. The data stored in the bundle storage buffer are sent to the destinations or neighbor nodes using the UDP or TCP transport layer protocols. The delay-tolerant data, which are temporarily stored in nodes, are routed by means of specific DTN algorithms. There are many routing algorithms for DTNs, and a particular set of routing algorithms can be employed to route the data between the DTN nodes in TENs, such as geo-routing algorithms. The DTN orchestrator manages the routing protocol switching, and it chooses the most suitable one in accordance with internal decision algorithms and the data exchanged with the SDN controller. The flexibility provided by the SDN makes it easier to change the routing protocol while the TEN remains running.

SDN-DTN Architecture in Action

Two use cases are examined to illustrate the practical application of the proposed architecture. The first sets out a scenario that highlights some of the SDN features within network partitions supporting a non-elastic application. The second complements the first by giving a description of an alternative situation, that is, a scenario in which two different network partitions are connected, supporting an elastic application.

Use Case #1: Intra-Network Partition

The first use case handles simultaneous video streaming in an intra-network partition scenario. This situation is depicted in Fig. 1, partition III. The UAV on the top collects video to be sent to the commander's vehicle, located at the bottom. This task is valuable in reconnaissance missions, when a platoon arrives at a region of interest and UAVs are used to search for possible threats [11]. The video transmission is a non-elastic application and must meet strict QoS requirements, as shown in Table 1, or there is a risk that the user may lose significant events.

Although all the devices are in the same partition, due to the distance between the source and destination nodes, RNs must be employed. The bandwidth of an RN may be shared with many data flows from different video sources. The SDN controller has global and near-real-time information about the link usage, and thus can select appropriate paths to forward video flows that comply with the requirements. Additionally, the controller employs mechanisms to ensure fairness among concurrent video flows or evenly distribute the data streams among redundant links. For example, in an area of 4 km $\times$ 4 km in

which nine UAVs and nine ground vehicles form a mesh network, from one to nine UAVs may provide video streams toward the commander's vehicle. Figure 3 shows the video playback start time (Fig. 3a), the number of video stalls (Fig. 3b) and the lengths of the stalls (Fig. 3c) for simulations performed according to the parameters provided in Table 2. The videos are compatible with military application needs [9, 14]. The video metrics were collected from the player reflecting the user experience. According to [9, 11], it is possible to infer that for TEN operating environments and the way a military operation in this level unfolds, a small number of short video stalls (around 10–15 stalls of 1–2 s each) do not harm the usage of the video. By the presented results, with more than six streams the system becomes visibly degraded. However, this degradation is due to the inherent limits of the used physical layer, as the SDN controller correctly behaved, selecting the best routes. Improvements can address this problem by changing the physical layer or using adaptive protocols to diminish the image resolution or the frame rate. Without using the proposed architecture, even with just one stream, the results are poor and, for many simultaneous streams, completely unacceptable.

Use Case #2: Inter-Network Partitions

The second use case deals with communication between heterogeneous nodes in the TEN, spread across different partitions. In Fig. 1, there is an illustration of the transmission of data acquired by WSN nodes located in partition I (on the left of Fig. 1) to a C2 application running in the vehicle in which the operational commander is following the battle unfold, in partition III (on the right in Fig. 1).

The WSN is initially isolated from the other partitions, and covers an area that is far away or difficult to access, where it collects data about enemy movements. The sensor nodes store the collected data in temporary memory or forward them to a particular sensor node that is capable of storing larger amounts of data.

A UAV performing a task nearby is assigned to collect the WSN data. When it establishes a connection with the WSN nodes, the leader election chooses a node to run the SDN controller and DTN orchestrator instances. The SDN controller updates the topology data concerning the current network partition. The flow rules applied to the forwarding devices classify the data being transmitted between the WSN nodes and the UAV. The WSN data belong to an elastic application, and thus DTN support is used. The SDN controller notifies the DTN orchestrator about the data being received and optimizes the data transfer between the DTN nodes and the UAV, by selecting links that are capable of proceeding with the data transfer. These links connect the UAV to other UAVs and/or military ground vehicles that can act as data forwarding nodes inside a given partition and between nearby partitions that are opportunistically joined.

Parameter	Used Value
Number of UAVs	9
Number of ground vehicles	9
Total simulation area	4 Km × 4 Km
UAV moving area	4 Km × 1 Km
UAV communication radius (R_u)	360m
Ground communication radius (R_v)	650m
Video size	960 × 540 pixels
Video frame rate	30 f/s
Video codec	H.264
Video length	60 seconds
Number of runs per number of streams being served	33 runs
Wireless standard	802.11g
Frequency range	2.4 – 2.485 GHz
Data rate	Up to 54Mb/s
Simulation environment	Mininet-WiFi
Mobility model	Random Waypoint

Table 2. Parameters used in the simulations.

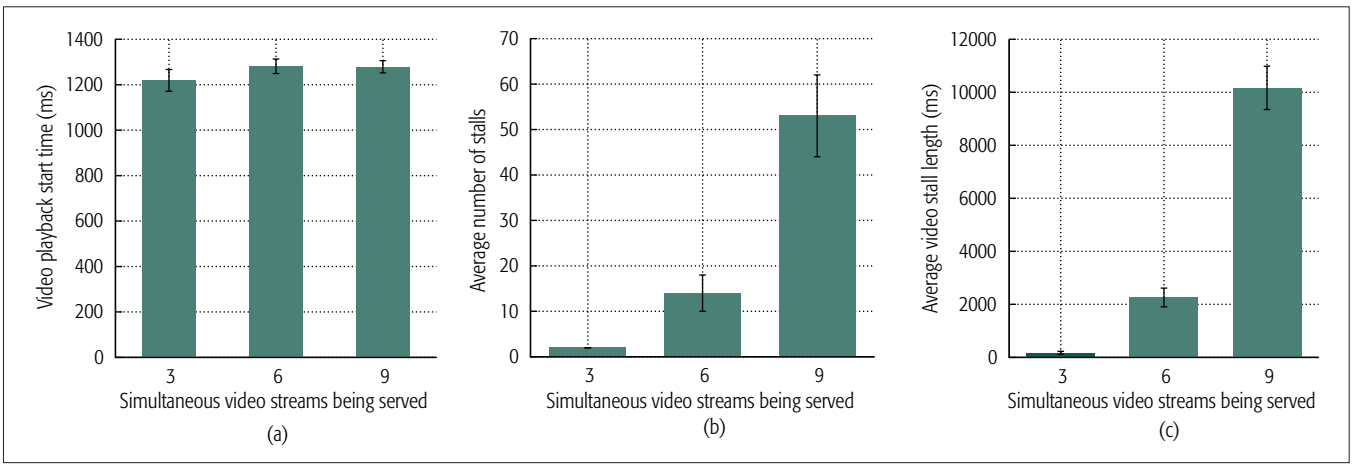


Figure 3. Simulations employing simultaneous video transmissions: a) playback start time; b) average number of stalls per video stream; c) average video stall length per video stream.

Observing advances in both SDN and DTN paradigm, a promising approach to combine the best of these to address last-mile TEN is proposed in this work. The architecture shows features capable of tackling this difficult operational scenario. It should also be stressed that other domains may benefit from this architecture.

As the UAV continues its movement around the area, a connection is eventually made with another partition. Again, the leader election algorithm selects a node to host the SDN controller and DTN orchestrator instances. The DTN orchestrator is aware of the data acquired from the WSN and interacts with the SDN controller to choose the best route to forward the data stored by the DTN layer toward the destination node. The SDN controller installs flow rules in the forwarding devices to allow the data to be transmitted via partition II until it reaches the destination in partition III. Preliminary results from the experiments within this scenario show that the proposed approach is able to successfully deliver 100 percent of the data sent by the WSN. On the other hand, the same network without the proposed SDN-DTN approach has results with a success rate for delivered data ranging from 44 to 55 percent.

Another possible way of using the WSN data in this scenario would be similar to mobile microclouds [15], in which the data could be processed by services placed on nodes inside partition I, after the UAV that gathered the data from the WSN rejoins this partition.

Conclusion

The BN is a challenging communication environment in which very different concerns have to be considered. Attempts to address these concerns have involved exploring approaches based on SDN or DTN. Despite the features that these two paradigms provide, they cannot address all these concerns on an individual basis. Moreover, few proposals have examined them in the last-mile TEN, which is the most challenging part of the BN due to its high mobility and constrained resources.

Observing advances in both the SDN and DTN paradigms, a promising approach to combine the best of these to address last-mile TEN is proposed in this work. The architecture shows features capable of tackling this difficult operational scenario. It should also be stressed that other domains may benefit from this architecture, such as networks designed to assist disaster relief operations, which face similar harsh operational conditions.

Prospective Directions

Despite the contribution presented in this work, many challenges remain to be solved by academia and industry. Solutions to keep network topology consistency when facing highly dynamic situations are under development, and mechanisms to migrate the SDN controller between nodes connected by restricted bandwidth wireless links require further work. Network signaling in low-end nodes needs improvements to efficiently apply the SDN paradigm regarding the concern about the energy consumption of these nodes. The development of adaptive protocols to change the video rate and resolution according to network conditions is an interesting topic that also requires effort to evolve. Also, the integration of the proposed architecture with existing military network solutions needs further research, mainly related to network security.

The development of a suite of protocols specifically designed for the proposed architecture and envisaged applications might be considered as a promising future work direction. Also, improvement of the leader election mechanisms to cope with the network resource constraints and the recurrent changes in network topology is still a relevant topic to be explored. Further, the adaptation of this approach to different contexts, such as emergency networks, and the exploration of edge computing and micro-clouds to complement and extend the proposal also deserve examination.

Acknowledgments

The authors thank Marcelo A. Marotta for the valuable discussions, Ramon. R. Fontes for developing the Mininet-Wifi emulator and the support for its usage, and the State of Rio Grande do Sul Research Foundation (FAPERGS Proj. no. 2240-2551/14-2SIAFEM) for the financial support to perform this work.

References

[1] M. Tortonesi *et al.*, "Enabling the Deployment of COTS Applications in Tactical Edge Networks," *IEEE Commun. Mag.*, vol. 51, no. 10, Oct. 2013, pp. 66–73.
[2] J. Nobre *et al.*, "Toward Software-Defined Battlefield Networking," *IEEE Commun. Mag.*, vol. 54, no. 10, Oct. 2016, pp. 152–57.
[3] D. Orfanus, E. P. De Freitas, and F. Eliassen, "Self-Organization as a Supporting Paradigm for Military UAV Relay Networks," *IEEE Commun. Lett.*, vol. 20, no. 4, Apr. 2016, pp. 804–07.
[4] J. Wickboldt *et al.*, "Software-Defined Networking: Management Requirements and Challenges," *IEEE Commun. Mag.*, vol. 53, no. 1, Jan. 2015, pp. 278–85.
[5] K. Fall and S. Farrell, "DTN: An Architectural Retrospective," *IEEE JSAC*, vol. 26, no. 5, June 2008, pp. 828–36.
[6] R. Amin *et al.*, "Design Considerations in Applying Disruption Tolerant Networking to Tactical Edge Networks," *IEEE Commun. Mag.*, vol. 53, no. 10, Oct. 2015, pp. 32–38.
[7] T. Luo, H. P. Tan, and T. Q. S. Quek, "Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 16, no. 11, Nov. 2012, pp. 1896–99.
[8] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking," *IEEE Commun. Mag.*, vol. 51, no. 2, Feb. 2013, pp. 114–19.
[9] P. Bartolomasi *et al.*, "NATO Network Enabled Capability Feasibility Study," NATO Consultation, Brussels, Belgium, tech. rep. 2.0, Oct. 2005.
[10] A. Kott, A. Swami, and B. J. West, "The Internet of Battle Things," *Computer*, vol. 49, no. 12, Dec. 2016, pp. 70–75.
[11] R. Fernandes, M. R. Hieb, and P. Costa, "Levels of Autonomy: Command and Control of Hybrid Forces," *Proc. 21th C2 in a Complex Connected Battlespace*, Sept. 2016, pp. 1–16.
[12] M. R. Brannsten *et al.*, "Toward Federated Mission Networking in the Tactical Domain," *IEEE Commun. Mag.*, vol. 53, no. 10, Oct. 2015, pp. 52–58.
[13] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 6th ed., ser. Always Learning, Pearson, 2013.
[14] J. Nightingale *et al.*, "Reliable Full Motion Video Services in Disadvantaged Tactical Radio Networks," *Proc. 2016 Int'l. Conf. Military Commun. and Info. Systems*, May 2016, pp. 1–9.
[15] S. Wang *et al.*, "Emulation-Based Study of Dynamic Service Placement in Mobile Micro-Clouds," *Proc. 2015 IEEE MILCOM*, Oct. 2015, pp. 1046–51.

Biographies

Iulisloi Zacarias (izacarias@inf.ufrgs.br) is an M.Sc. student in computer networks at the Federal University of Rio Grande do Sul (UFRGS), Brazil. He achieved his Bachelor's degree in Information Systems at the Federal University of Santa Maria (UFSM), Brazil, 2015. Currently, his research interests are related to wireless networks, drone networks, software-defined networks, ad hoc networks, and the Internet of Things.

Luciano Paschoal Gaspary (paschoal@inf.ufrgs.br) holds a Ph.D. in computer science (UFRGS, 2002). He is deputy dean and associate professor at the Institute of Informatics, UFRGS. He has been involved in various research areas, mainly computer networks, network management and computer system

security. He is an author of more than 120 full papers published in leading peer-reviewed publications. In 2016, he has been appointed as a Publications Committee member of the IEEE SDN initiative.

Andersonn Kohl (kohl@cds.eb.mil.br) has a postgraduate degree in military sciences from EsAO (1998), a communication engineering degree from IME (1996), and a Bachelor's in military sciences from AMAN (1990). He is a Colonel Military Engineer in the Brazilian Army with much experience in the development and deployment of communication and software defense systems, and is currently the director and senior researcher of the C2 Division at CDS in transition to a new position as assistant to the Army vice-chief of Information Technology and Communications.

Ricardo Queiroz de Araujo Fernandes (ricardo@cds.eb.mil.br) attended a postdoctoral program in command and control at GMU (2016) and has a D.Sc. degree in Informatics from PUC-Rio, Brazil (2012), an M.Sc. in systems and computing from IME (2009), an M.Sc. in pure mathematics from UFRGS (2007), a postgraduate degree in military sciences from EsAO (2009), and a systems and computing engineering degree from IME (2002). Currently he is responsible for the research on command and control at CDS.

Jorgito Matiuzzi Stocchero (stocchero.jorgito@eb.mil.br) has an M.Sc. degree in electrical engineering by COPPE/UFRJ (2004), an M.B.A. in politics and strategy by FGV/RJ (2016), a postgraduate degree in military sciences from ECEME (2016 and 2012), a communication engineering degree from IME (1996) and a Bachelor's in military sciences from AMAN (1990). He has worked on important projects during his career in the Brazilian Army, such as SIVAM and SisTEx. Currently, he is a Colonel assigned as vice-commander of the Telematics Integrated Center.

Edison Pignaton de Freitas (epfreitas@inf.ufrgs.br) has a Ph.D. in computer science and engineering from Halmstad University, Sweden (2011), an M.Sc. in computer science from UFRGS (2007), and a computer engineering degree from the Military Institute of Engineering (2003). Currently he holds an associate professor position at UFRGS, acting in the Graduate Programs on Computer Science (PPGC) and Electrical Engineering (PPGEE) developing research mainly in the following areas: computer networks, real-time systems, and unmanned aerial vehicles.

APPENDIX B — PUBLISHED PAPER (NCA 2017)

Iulisloi Zacarias, Janaína Schwarzrock, Luciano P. Gaspary, Anderson Kohl, Ricardo Q. A. Fernandes, Jorgito M. Stocchero, Edison P. de Freitas. **Employing SDN to Control Video Streaming Applications in Military Mobile Networks**. In 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), Cambridge, 2017, pp. 1–4. DOI: 10.1109/NCA.2017.8171390.

- **Title**: Employing SDN to Control Video Streaming Applications in Military Mobile Networks.
- **Abstract**: Video streaming is an important service provided by surveillance systems to enhance situation awareness. However, in military systems, data acquisition heavily depends on the network infrastructure. In this application domain, units are spread and the distance between the sources of data and the decision makers may be very large. In the case of video streaming, the demand for high network throughput poses some extra requirements on the network. Considering the mobility patterns of the military units and the diversity of the new generations of sensors, especially those used by Unmanned Aerial Vehicles (UAV), the configuration and the management of the network must be so dynamic and so sensitive to data flow parameters that manual configuration is not acceptable. For this reason, the capability of the network to configure itself to offer the necessary Quality of Service is a must. Using principles of Software Defined Networks (SDN), this paper presents an analysis of video streaming for military surveillance in which multiple UAVs are employed as data providers through an SDN-enabled network, with promising results.
- **Status**: Published.
- **Qualis / CAPES**: B1.
- **Conference**: 16th International Symposium on Network Computing and Applications (NCA).
- **Date**: Oct 30 – Nov 1, 2017.
- **Local**: Cambridge, MA.
- **URL**: <http://www.ieee-nca.org/2017/>.
- **DOI**: <https://doi.org/10.1109/NCA.2017.8171390>.

Employing SDN to Control Video Streaming Applications in Military Mobile Networks

Iulisloi Zacarias*, Janaína Schwarzrock*, Luciano P. Gaspary*, Anderson Kohl†,
Ricardo Q. A. Fernandes†, Jorgito M. Stocchero† and Edison P. de Freitas*
*Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil
†Brazilian Army
{izacarias,jschwarzrock,paschoal,epfreitas}@inf.ufrgs.br; {kohl,ricardo}@cds.eb.mil.br; celstocchero@dct.eb.mil.br

***Abstract*—Video streaming is an important service provided by surveillance systems to enhance situation awareness. However, in military systems, data acquisition heavily depends on the network infrastructure. In this application domain, units are spread and the distance between the sources of data and the decision makers may be very large. In the case of video streaming, the demand for high network throughput poses some extra requirements on the network. Considering the mobility patterns of the military units and the diversity of the new generations of sensors, especially those used by Unmanned Aerial Vehicles (UAV), the configuration and the management of the network must be so dynamic and so sensitive to data flow parameters that manual configuration is not acceptable. For this reason, the capability of the network to configure itself to offer the necessary Quality of Service is a must. Using principles of Software Defined Networks (SDN), this paper presents an analysis of video streaming for military surveillance in which multiple UAVs are employed as data providers through an SDN-enabled network, with promising results.**

I. Introduction

Nowadays, Unmanned Aerial Vehicles (UAVs) became a widely used technology. Applications have emerged in many fields, such as precision agriculture and communication support. Also, UAVs are widely employed in military applications as border surveillance and reconnaissance missions [1].

Ground surveillance and reconnaissance missions make use of imaging and video resources. Wireless transmission of those media must meet rigorous requirements concerning latency, latency variation (jitter) and throughput. Although the requirements for a useful video stream transmission in military applications differ from entertainment applications, they can be measured similarly. Measurements of video stream playback quality can be classified into objective and subjective metrics [2]. Objective measurements can be collected in the user video player. Subjective metrics like Mean Opinion Score (MOS) are based on the user feedback.

As demonstrated in [3], it is possible to infer the user subjective evaluation based on the objective measurements. The video quality assessment can also be used as a feedback to adjust network settings and policy enforcements [4]. Software-Defined Networks (SDNs) [5] enables configuration changes and policy enforcement in a dynamic and practical manner. The network control is centralized in the SDN controller, which provides a programmatic interface to the network.

It is a major challenge to keep video quality requirements at acceptable levels, considering a highly dynamic environment such as multi UAV-based surveillance setups. Most of the solutions proposed to UAV surveillance networks (commercial or military) make use of ad hoc solutions based on wireless mesh networks [6]. The use of conventional network solutions (non-SDN) in these applications makes the device reconfiguration process difficult, or dependent on proprietary solutions [7].

In [8], Nam *et al.* propose an architecture and an SDN controller that can adjust the network parameters to deliver video traffic to the final user with better Quality of Experience (QoE). In [9] the authors explore the overall view of the network, provided by the SDN controller, to assist the Dynamic Adaptive Media Players (DASH) to select the best video resolution supported by the network. The work presented in [4] proposes a solution to cope with network bandwidth competition among concurrent video flows in the network.

In light of this discussion, this paper proposes an SDN approach to improve the quality of video stream transmissions. An assessment of video quality is conducted employing objective measurements, with MOS assessment. The major contributions of this work are: i) application of SDN in UAV-based military surveillance scenarios; ii) demonstration of the feasibility of using an OpenFlow [5] compatible solution, enabling the efficient usage of Commercial Off-The-Shelf (COTS) equipment in military networked applications; and iii) an analysis based on quality of experience indicators and experimental results supporting the proposed approach.

II. Military Surveillance Application Scenario

To carry out military surveillance, a suitable proposal is to use a fleet of UAVs to obtain video of the monitored site. The UAV sends the captured data to a command center, where the video will be examined to support decision making. The UAVs and the Command and Control (C2) systems form a network which may be partially disconnected or disrupted due to the possibility of wide UAV movements to cover larger areas. This trade-off (coverage X connectivity) was studied in [6].

Focusing on military reconnaissance missions, the combined use of small UAVs and conventional ground military vehicles is a promising setup. This combination can provide awareness of threats expected ahead of the troop's line of sight. In this kind of reconnaissance missions, the military vehicles move along an axis of advance in the direction of the enemy as shown in Fig. 1. In this situation, the goal is to acquire visual information about the clearance of the area ahead. Using this information the platoon commander can decide about the

978-1-5386-1465-5/17/$31.00

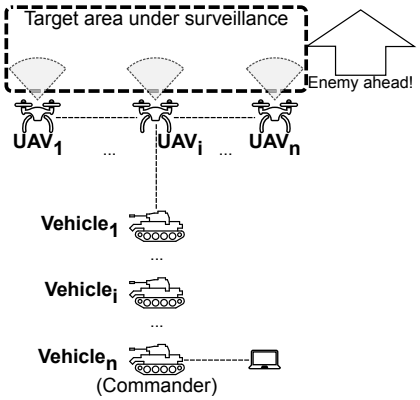


Fig. 1. Schematic scenario in which UAVs provide clearance to the ground military vehicles advance along the axis.

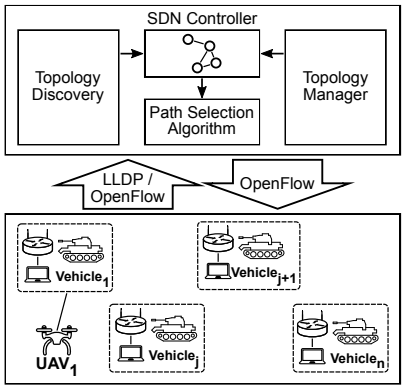


Fig. 2. Schematic SDN Controller Architecture

advancement of his troop. The vehicle occupied by the platoon commander is in general one of the last vehicles in the row. The video acquired by the UAVs should be delivered to this vehicle and played in the commander's C2 terminal.

The video transmission in the described scenario must meet restrict requirements, under the risk of losing significant events about the enemy movements. A critical requirement relates to the duration of each possible video interruption. For example, a long interruption in the monitoring video allows an undetected enemy to come close the platoon, posing security risks. For video transmissions triggered by some event of interest, the video playback start time is also important.

III. An SDN Architecture to Enhance Video Streaming in Dynamic Networks

The expansion of the ground coverage area maintaining connectivity among UAVs and the ground platoon is one of the problems to be addressed and can be acquired using intermediate UAVs as relay nodes. These nodes act forwarding the data until it reaches the destination node. The use of relay nodes has a drawback: the overload of the intermediate nodes. Intermediate nodes send data collected by their sensors (e.g. video cameras) and additionally, the data sent by the neighbor nodes. Thus, the throughput of the communication channels is shared among different video streams.

The concurrency of the throughput by many video flows affects the video QoE. The negative feedback of a video application is expressed by a low MOS score. The selected path to forward the video stream and the number of hops needed to reach the destination has a significant impact on this assessment. A management entity with a global view of the network can select optimal links to route the data, avoiding congestions and choosing paths with fewer hops. An SDN based on the OpenFlow [5] architecture enables the controller to gather information about the global state of the network (e.g., detect congested and deteriorated links).

Video streams used in military surveillance applications show slightly different characteristics to those used for entertainment. The purpose of the captured video in military missions is the rapid detection of threats, gather information about the enemy, or assist in the organization of the units. The features of the Remote Video Terminals (RVT) on embedded systems also influence the choice of the videos resolutions. The images are usually obtained by cameras, synthetic aperture radar (SAR), and infrared devices. The videos used in the military context range from resolutions of few Kilopixels to HD images [10]. Focusing on reconnaissance missions, like that presented in Fig. 1, videos with low resolution (e.g. 640x512 pixels) are enough to support decisions.

The proposed network architecture provides a software-defined environment to manage the data and the SDN controller to handle the network of ground vehicles. Each of the ground vehicles is equipped with a data forwarding device enabling the creation of a programmable network. All forwarding devices are connected to the same SDN controller. Thus, in this proposed architecture each ground vehicle is considered a network switch controlled by the SDN controller.

The UAVs move around the area where they are conducting the surveillance mission, and therefore, the connection to the network composed of ground vehicles is not stable. During the mission, the UAVs can often reconnect to the same ground vehicle or connect to a different one. The SDN controller must optimize the network to make the reconnection process as smooth as possible, with minimal impact on the video quality.

When starting the network operation, a topology discovery process is performed, and it allows the controller to select the most suitable path for the transmission of the video streams. Once an OpenFlow-enabled device connects to the SDN controller, it starts a handshaking process that allows the controller to be aware of all forwarding devices in the SDN data plane. The LLDP protocol performs the discovery of the links among the discovered devices as depicted in Fig. 2.

In the SDN controller, the topology data gathered is represented by an undirected graph. The controller computes the routes among the devices based on the information contained in this network representation. As the topology of the network changes, the graph needs to be updated to correspond to the new network topology. The controller detects OpenFlow-enabled devices that are joining or leaving the data plane through the OpenFlow Channel. The controller also periodically checks for updates in the links using the LLDP protocol.

After the initial topology discovery of the OpenFlow-

enabled devices, the SDN controller is ready to process the requests. The SDN controller identifies the UAVs in the TEN based on a Layer 2 learning process. When a UAV joins the TEN, the OpenFlow-enabled access point does not have the rule to forward the packets from the new UAV. Following the behavior of OpenFlow specification, the forwarding device sends the data packet to the SDN controller. Key components of the SDN Controller that perform these tasks and their relations to each other are illustrated in Fig. 2.

Each data packet that arrives at the SDN controller triggers the *Packet_in* event. As a UAV joins the TEN, the SDN controller updates the network graph representation. Next, the SDN controller searches for the destination address host *dst_address* in the network graph. If the target host specified by the *dst_address* field in the data packet is found, the network controller installs an OpenFlow entry in the forwarding device that originates the *Packet_in* event and forwards the packet to the next hop, or to the destination host. Otherwise, if the network graph representation does not have information about the host specified by the *dst_address*, the controller floods the network with an ARP Request packet until the destination host answers the request. To prevent packets from being forwarded in network loops, the controller gets information about all ports in the *datapath* (logical representation of the forwarding device), compute the ports that may cause loops in the network using a minimum spanning tree (MST) algorithm and sends the ARP Request package only to loop-free ports.

IV. Experiments and Results

A. Selected Evaluation Metrics

Three objective metrics were elected to quantify the Application Quality of Experience (AppQoE) on the client side using video over HTTP [2], [11]. These metrics are:

1) Video playback start time: The time taken by the player to start the *playout*, from the moment the stream is requested.

2) Number of interruptions: When the playback is temporarily frozen a video interruption is computed.

3) Total duration of interruptions: The sum of the duration of all interruptions (*Buffering Time*) during video playout. The *Initial Buffering* event is ignored [2].

The predictions of the MOS values were obtained from the AppQoE data collected. Studies relate the influences of the selected metrics with the degradation of QoE, represented by MOS score that would be assigned by the user. In [3], the authors relate the video playback start time with the MOS value using (1), where $Qini$ is the MOS influenced value and t_0 is the video playback start time.

The MOS value influenced by video stalls ($Qstall$) can be obtained by applying (2) and (3) to the AppQoE data. The λ factor is the ratio of the total time that the video was stalled(σ) and the interval elapsed since the beginning of observation, given by the sum of σ with the effective video play time (ρ). The value of λ is used to determine the value of the a_i, b_i and c_i constants in (3) according to predefined values observed by Casas *et al.* [12]. The n variable corresponds to the number of video interruptions on the T time interval, considered as one minute in the present work. Since $Qini$ and $Qstall$ were independently obtained, the minimum value is the final MOS.

TABLE I
Parameters used in the simulations

Parameter	Used Value	Parameter	Used Value
Number of UAVs	9	Mobility models	Random Walk,
Number of ground vehicles	9		Random Waypoint
Total simulation area	4 Km x 4 Km	Video size	960x540 pixels
UAV moving area	4 Km x 1 Km	Video frame rate	30 fps
UAV communication radius	360m	Video codec	H.264
Ground communication radius	650m	Video length	60 seconds
Number of runs per number of streams being served	33 runs		

$$Qini = -0.963 \times \log\left(t_0 + 5.381\right) + 5 \tag{1}$$

$$\lambda = \begin{cases} \frac{\sigma}{\sigma+\rho}, & \text{if } \sigma + \rho < T \\ \frac{\sigma}{T}, & \text{otherwise} \end{cases} \tag{2}$$

$$Qstall = \begin{cases} 1, & \text{if } n > 6 \\ a_i \times e^{-b_i \times n} + c_i, & \text{if } n \leq 6 \end{cases} \tag{3}$$

B. Simulated Scenario and Parameters

The experiments were performed considering the scenario described in Section II using Mininet-WiFi[1], Ryu-SDN Framework[2] and FFmpeg[3] player. An SDN was used to connect the ground vehicles, according to the description in Section III. Links between ground vehicles were considered of 100 Mbps.

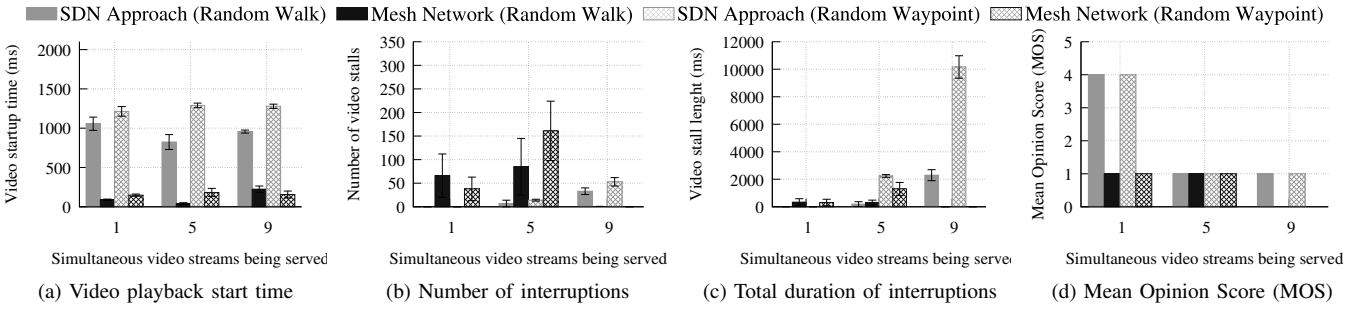
The network that connects ground vehicles forms a kind of bus. UAVs connect to ground vehicle closest to the UAV-squad. There is a mesh network among UAVs to enable farthest UAVs to connect to the ground vehicles' network. For example, in Fig. 1 the UAV_n cannot connect directly to the ground vehicles' network. The mesh network among UAVs allows it's data packets reach the destination network using UAV_i.

The experiments were performed with multiple simultaneous video streams, ranging from a one to nine video streams being transmitted simultaneously. The video characteristics are described in the Table I. The UAVs acting as video stream servers were uniformly and randomly selected. The software displaying the generated video streams is located in the farthest vehicle of UAVs squad (the Vehicle_n in the Fig. 1). This setup follows the military doctrine and also represents the worst case for data transmission.

C. Results Presentation and Discussion

In all of the performed experiments, the measurements of video playback start time hold at an acceptable value for a video surveillance application. The values are in the range from 800 milliseconds to 1300 milliseconds as depicted in Fig. 3a. The SDN approach presented values larger than the mesh network due to the communication of the forwarding devices to the SDN controller. Regardless of video playback start time influence in the MOS note, for the cases with up to 4 simultaneous streams this value is still acceptable.

[1]https://github.com/intrig-unicamp/mininet-wifi
[2]https://osrg.github.io/ryu/
[3]https://ffmpeg.org/



(a) Video playback start time (b) Number of interruptions (c) Total duration of interruptions (d) Mean Opinion Score (MOS)

Fig. 3. Simulation results for Random Walk and Random Waypoint Mobility Models.

Examining the example in Section II, a platoon moves at a speed of 60Km/h against enemies coming from the opposite direction at the same speed. The use of the UAV squad providing clearance in the area 1Km ahead sufficiently addresses the needs. The video playback starts within an acceptance time interval, even in the worst case. The number of video interruptions was strongly influenced by the number of simultaneous video flows. The various streams competing for available bandwidth generated resource contention.

The mobility model Random Walk resulted in more stable wireless associations, providing better results because it allows a higher data rate transfer, thus avoiding the starvation of the player buffer. As a consequence, the number of video stalls and the total stall length depicted in Fig. 3b and Fig. 3c present slightly better values for the Random Walk mobility model. Even in the simulations using the Random Waypoint mobility model, the total video stall time is satisfactory up to eight simultaneous video streams. The stall lengths in the SDN approach are also within an acceptable interval, being in the worst case 10s. It was not possible to measure the number of video stalls, the total stall length, and the MOS note from 6 to 9 simultaneous streams in the mesh network due to connections errors.

According to the predicted MOS values, it was observed good results using up to 4 simultaneous video streams with the Random Walk mobility model as depicted in Fig. 3d, which represents approximately a half of the UAV platoon sending videos. The SDN approach overcomes the mesh network that was not able to present MOS values greater than 1 as observed in Fig. 3d. It is noteworthy the video frame rate used in the simulation ($\sim$ 30fps), normally video surveillance applications require slow rates. With smaller video frame rate the streams should be lesser resource intensive. Thus a better user experience is expected. The same applies to the video resolutions selected for the simulations.

V. Conclusion

This work demonstrates the use of SDN in the context of military mobile networks. The work considers military applications, however the described scenarios can also be used in the civilian domain. The proposal was evaluated in an emulator. By collecting and analyzing objective QoE measurements, the MOS indicator was predicted. The results were promising and demonstrate that SDN can be successfully applied to heterogeneous and networks with opportunistic connections. As future work, it can be suggested the application of SDN in UAV relay network, allowing greater control of packet routing. Moreover, DTN protocols could also be used in conjunction with the SDN-based solution.

Acknowledgments

The authors thank to State of Rio Grande do Sul Research Foundation (FAPERGS – Proj. nr. 2240-2551/14-2SIAFEM) and Brazilian National Council for Scientific and Technological Development (CNPq) for the financial support.

References

[1] L. Gupta, R. Jain, and G. Vaszkun, "Survey of Important Issues in UAV Communication Networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, Apr. 2016.

[2] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 401–418, Jan. 2016.

[3] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *4th Int. Workshop on Quality of Multimedia Experience*, Jul. 2012, pp. 1–6.

[4] R. M. Abuteir, A. Fladenmuller, and O. Fourmaux, "SDN Based Architecture to Improve Video Streaming in Home Networks," in *30th Int. Conf. on Advanced Inform. Networking and Applicat.* IEEE, Mar. 2016, pp. 220–226.

[5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Comput. Commun. Review*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[6] D. Orfanus, E. P. De Freitas, and F. Eliassen, "Self-Organization as a Supporting Paradigm for Military UAV Relay Networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 804–807, Apr. 2016.

[7] P. Patil, A. Hakiri, Y. Barve, and A. Gokhale, "Enabling software-defined networking for wireless mesh networks in smart environments," in *15th Int. Symp. on Network Computing and Applicat. (NCA)*. IEEE, Oct. 2016, pp. 153–157.

[8] H. Nam, K. H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware video streaming using SDN," in *IEEE Global Commun. Conf.*, Dec. 2014, pp. 1317–1322.

[9] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering Stable High-quality Video: An SDN Architecture with DASH Assisting Network Elements," in *7th Int. Conf. on Multimedia Syst.*, May 2016, pp. 4:1–4:10.

[10] J. Nightingale, Q. Wang, J. M. A. Calero, I. Owens, F. T. Johnsen, T. H. Bloebaum, and M. Manso, "Reliable full motion video services in disadvantaged tactical radio networks," in *Int. Conf. on Military Commun. and Inform. Syst.*, May 2016, pp. 1–9.

[11] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," *IFIP/IEEE Int. Symp. on Integrated Network Manage. and Workshops*, vol. 1, pp. 485–492, May 2011.

[12] P. Casas, R. Schatz, and T. Hoßfeld, "Monitoring YouTube QoE: Is Your Mobile Network Delivering the Right Experience to your Customers?" Apr. 2013, pp. 1609–1614.

APPENDIX C — OTHER SUBMISSIONS AND COLLABORATIONS

Iulisloi Zacarias, Carlos E.T. Leite, Janaína Schwarzrock, Edison P. de Freitas. **Control Platform for Multiple Unmanned Aerial Vehicles**. In IFAC-PapersOnLine, Volume 49, Issue 30, pp. 36–41, November 2016. DOI: 10.1016/j.ifacol.2016.11.119.

- **Qualis / CAPES**: N/A.
- **Status**: Published.

Janaína Schwarzrock, **Iulisloi Zacarias**, Ana L.C. Bazzan, Ricardo Queiroz de Araujo Fernandes, Leonardo Henrique Moreira, Edison Pignaton de Freitas. **Solving task allocation problem in multi Unmanned Aerial Vehicles systems using Swarm intelligence**. In Engineering Applications of Artificial Intelligence, Volume 72, pp. 10–20, June 2018. DOI: 10.1016/j.engappai.2018.03.008.

- **Qualis / CAPES**: A2.
- **Status**: Published.

Iulisloi Zacarias, Janaína Schwarzrock, Luciano Paschoal Gaspary, Kohl Anderson, Ricardo Q. A. Fernandes, Jorgito M. Stocchero, Edison Pignaton de Freitas. **Enhancing Mobile Military Surveillance based on Video Streaming by Employing Software Defined Networks**. In Wireless Communications and Mobile Computing. pp. 1–20.

- **Qualis / CAPES**: A2.
- **Status**: Under Review.

Gabriel Martins Leal, **Iulisloi Zacarias**, Jorgito Matiuzzi Stocchero, Edison Pignaton de Freitas. **Empowering Command and Control by using Combined Information Centric and Software Defined Networking**. In IEEE Communications Magazine. pp. 1–7.

- **Qualis / CAPES**: A1.
- **Status**: Under Review.