Università
degli Studi
di Ferrara

DOTTORATO DI RICERCA IN
SCIENZE DELL'INGEGNERIA
CICLO XXXIV
COORDINATORE Prof. Stefano Trillo

Middleware Solutions for IoT-based Application in Disrupted Network Environments

Settore Scientifico Disciplinare ING-INF/05

Dottorando
Lorenzo Campioni

Tutore
Chiar.mo. Prof. Ing. Cesare Stefanelli

Cotutore
Prof. Ing. Mauro Tortonesi

Anni 2018-2021

Sommario

Grazie all'Internet of Things (IoT), sensori, attuatori, risorse computazionali e memorie si stanno sempre più integrando nella vita quotidiana, trasformando radicalmente il modo in cui gli esseri umani interagiscono e percepiscono ciò che li circonda. In particolare, IoT permette ai servizi informatici di sfruttare la capillare ed eterogenea rete di dispositivi per acquisire informazioni in tempo reale e interagire automaticamente con l'ambiente. A causa della natura distribuita della loro architettura, i servizi basati sull'IoT necessitano di una infrastruttura di rete in grado di supportare costantemente la condivisione di informazioni tra i servizi e i dispositivi. Tali caratteristiche non sono però riscontrabili in situazioni in cui la rete è inaffidabile, come avviene nel caso di zone urbane colpite da disastri e in reti tattiche militari che sono caratterizzate da reti instabili che impediscono ai nodi di condividere efficacemente informazioni. Tuttavia, i middleware possono rappresentare uno strumento efficace per abilitare servizi basati sull'IoT anche in questi complessi scenari. Infatti, grazie a soluzioni basate su middleware è possibile definire un layer ubiquo per il management di dispositivi in grado di adattarsi rapidamente alle caratteristiche di rete e riabilitare l'accesso a dispositivi IoT. Più precisamente, attraverso i middleware è possibile definire una soluzione federata in grado di scoprire autonomamente dispositivi IoT in reti frammentate, permettendo quindi ad applicazioni di trovare e connettersi con queste risorse. Inoltre, i middleware sono in grado di migliorare l'interazione tra device e applicazioni implementando meccanismi information-centric che sfruttano al meglio le scarse risorse di rete. Ad esempio, paradigmi come publish-subscribe e Information Centric Networking (ICN) disaccoppiano gli estremi di comunicazione permettendo scambi di informazioni meno sensibili alle condizioni di rete. Questa tesi studia l'efficacia di soluzioni middleware per il discovery proattivo e il paradigma di comunicazione information-centric per permettere l'uso di applicazioni IoT in reti inaffidabili. A questo scopo questa tesi introduce un middleware distribuito per il discovery di device IoT in zone urbane colpite da disastri e fornisce una valutazione sperimentale di diversi protocolli di comunicazione information-centric in reti tattiche militari. Le ricerche presentate in questa tesi sono frutto di una collaborazione con diversi istituti internazionali e un periodo di lavoro svolto al Florida Institute for Human and Machine Cognition (IHMC), FL, USA.

Abstract

Thanks to the Internet of Things (IoT), sensors, actuators, computational and storage resources, are getting more and more embedded in everyday life, radically transforming human interaction and perception of their surroundings. More specifically, IoT allows Information Technology (IT) services to exploit a capillary network of heterogeneous devices to acquire real time information and to autonomously interact with the environment. However, due to their distributed architecture, IoT-based services are highly dependent on a reliable network infrastructure that can effectively connect edge devices to IT services and users' devices. Therefore, disrupted network environments prevent the effective use of IoT. Post-disaster urban environments and tactical environments are characterized by an unreliable network infrastructure that negatively affects the capability of nodes to share information. However, middleware can provide an effective tool to enable IoT-based applications in these challenging environments. In fact, by means of a middleware based approach it is possible to define a ubiquitous IoT management layer that can rapidly adapt to the network characteristics and re-enable access to IoT devices. More specifically, middleware can form a federated solution capable of autonomously locating IoT devices in highly fragmented network environments, thus allowing applications to discover and directly connect to such resources. Moreover, middleware can also support the communication between applications and IoT devices by implementing information-centring communication paradigms that better exploit the scarce network resources. In particular, communication paradigms such as publish-subscribe and Information Centric Networking (ICN) decouple the communications' endpoints and enable more resilient information sharing in highly disrupted environments. This thesis investigates the effectiveness of middleware for proactive discovery and the information-centring communication paradigm to enable the use of IoT-based applications in disrupted network environments. To this end, this thesis also presents a distributed discovery middleware for disaster recovery environments and experimentally evaluates the capabilities of multiple information centric communication protocols within tactical networks. The research presented in this thesis is the result of the collaboration with international institutes and a research period at the Florida Institute for Human and Machine Cognition (IHMC), FL, USA.

Contents

Acronyms

ABE	Attribute-Based Encryption.
AFRL	Air Force Research Labs.
BLE	Bluetooth Low Energy.
C2	Command and Control.
C2PC	Corps Command and Control Personal Computer.
CCN	Context Centric Networking.
CIMIC	Civilian-Military Cooperation.
CoAP	Constrained Application Protocol.
COC	Combat Operations Center.
CP-ABE	Ciphertext Policy Attribute-Based Encryption.
D2D	Device-to-Device.
DDS	Data Distribution Service.
FDMA	Frequency Division Multiple Access.
GkMS	Group Key Management Service.
HADR	Humanitarian Assistance and Disaster Relief.
ICN	Information Centric Networking.
ICT	Information and Computing Technology.
IMS	Information Management System.
IoBT	Internet of Battlefield Things.
IoT	Internet of Things.
IT	Information Technology.

JTCW	joint Tactical Common Operational Picture Workstation.
LoRa	Long Range.
LPWAN	Low-Power Wide-Area Network.
MARGOT	Multi-Domain AsynchRonous Gateway Of Things.
MEC	Mobile Edge Computing.
MQTT	Message Queuing Telemetry Transport.
NB-IoT	NarrowBand Internet of Things.
NDN	Named Data Networking.
OFDM	Orthogonal Frequency-Division Multiplexing.
OLSR	Optimized Link State Routing Protocol.
P2P	Peer-to-peer.
QoS	Quality of Services.
REST	Representational State Transfer.
SATCOM	Satellite Communication.
TLS	Transport Layer Security.
TOC	Tactical Operations Center.
TSOA	Tactical Service Oriented Architecture.
UAV	Unmanned Aerial Vehicle.
UHF	Ultra High Frequency.
UV	Unmanned Vehicle.
XMPP	Extensible Messaging and Presence Protocol.

Chapter 1

Introduction

Since the concept of IoT was defined in the end of the '80s its adoption in everyday life has gained more and more momentum [1, 2]. Sensors, actuators, computational and storage resources, are getting more and more embedded in our everyday life impacting the interaction and perception of humans with their surroundings. Ranging from mobile devices, smart home assistants, and building sensors to industrial automation enablers, the scale and diversity of implementations of IoT-technologies has multiplied manyfold. Apart from the hardware aspects, there have been developments in IoT related communication technologies, data communication protocols, security and privacy mechanisms, and interoperability aspects between IoT technologies as well as between IoT and legacy technologies [3, 4].

One of the most important environments drastically changed by the application of IoT are Smart Cities. In fact, in Smart Cities, the cheap, yet powerful and innovative solutions enabled by IoT, have paved the way for new business opportunities that are attracting large numbers of investors and industry leaders to the sector. Hence, more and more stakeholders actively participate in the enrichment of urban network environments accelerating the flourishing of Smart Cities and the implementation of innovative ICT solutions that improve human life. For example, in Smart Cities services such as health, transportation, security, education, and others, can benefit from the use of a pervasive network of IoT devices by offering time-critical, context-, and location-aware responses to citizens [5, 6, 7, 8].

The accessibility of "Things" and their interoperability with other systems are major challenges that Smart Cities have to face, as they directly impact their capability to advance at a fast pace by introducing new services or extending the ones currently offered [9, 10]. To address this issue, nowadays urban networks rely on a multi-layered infrastructure composed by gateways, Edge, Fog, and Cloud services that orchestrate the growing number of sensing and computational resources available. For example, public and private organizations that work in the field of smart services and IoT have been deploying a multitude of new Cloud-based ser-

vices that allow applications to retrieve access information on smart Things and other connected devices managed by those organizations. However, in order to correctly function, these services rely on top of swarm of computational resources distributed between both Edge and Fog networks of the city that directly cooperate with the constrained devices to address communication heterogeneity, mitigate the shortcomings caused by the resources' limits and reduce bandwidth occupation.

While the sophisticated infrastructure and the pervasive network of sensors and IoT devices that permeates smart cities is mainly fostered by the design and evolution of smart city services, it can also represent an indispensable support for disaster recovery operations. More specifically, if exploited, the sensing capabilities offered by nowadays smart city networks can enable rescuers to rapidly acquire situational awareness in severely unstable and dynamic environments, greatly simplifying the phases of planning, organization, and decision making, necessary for effective recovery efforts. However, accessing the sensing network can be obstructed by the conditions that affect the city's network during and after a disaster [11, 12, 13].

In particular, large scale disasters, such as earthquakes, floods or human-caused disasters, can physically damage the smart cities' network infrastructure and consequently interrupt network links, broke routers, or destroy antennas. Moreover, the unavailability of network links can cause the traffic to be re-routed on undamaged links, resulting in potential situation of network congestion. Both physical damages on the network infrastructure and traffic congestion fragment the network in an unpredictable set of quasi-, or completely, disjointed sub-networks with limited capabilities in which rescuers do not have prior knowledge of the possible IoT assets that the environments may offer to acquire situational awareness [14].

Similarly to post-disaster urban networks, also mobile military networks require tailored solutions to enable the use of IoT during tactical missions. The wireless nature of these networks, and the fact that they have to operate in an unfriendly environment, results in unreliable and capacity limited performance. At the same time, in order to achieve and maintain information superiority, in these environments there is a need for solutions that enable the insertion of networked sensors and IoT technology (e.g., the Internet of Battlefield Things (IoBT) program [15]), the deployment of autonomous nodes and swarms, the adoption of new communication patterns where available information might be attractive for more receivers in an ad-hoc manner, and the adoption of Joint Intelligence, Surveillance and Reconnaissance (JIRS) as well as joint firing and targeting.

In both of these uniquely challenging environments middleware can be an effective tool to enable IoT-based next generation emergency and military applications. In fact, by means of a middleware based approach it is possible to define a ubiquitous layer that responds to application needs and implement specific mechanisms to

restore access to IoT resources.

In particular, properly designed, proactive discovery procedures can autonomously locate and register surviving segments of the IoT infrastructure and provide access to rescuers. Moreover, the sensing network environment after a disaster is extremely dynamic, as rescue organizations can deploy new resources and military forces can bring network-enabled equipment within militarized sub-networks that can connect to the public network. Proactive discovery can autonomously enable access to resources that resisted the disaster just as well as new resources that different rescue teams and organizations deploy during emergency operation.

Although proactive discovery represents the core mechanism of the IoT accessing process, complementary federation mechanisms are crucial to effectively respond to disaster recovery and tactical environments needs. In fact, since both environments present fragmented network environments, proactive discovery can solely restore accessibility of devices within their area of influence. On the other hand, by integrating federation mechanisms, it is possible to support the entirety of the network by relying on a federation of autonomous discovery services that can cover each fragment of the network. Moreover, military and disaster recovery operations typically involve multiple agencies (e.g. military missions might be carried out by forces deployed by diverse nations with a common goal) each adopting different applications and introducing proprietary resources for the operations [16, 17].

In such cases, federation mechanisms can be enabled by defining specific secure communication schemes that foster the cooperation and resources sharing among different partners. For example, secure group communication schemes can adopt attribute-based mechanisms that allows users to define information access policy with ease and thus assuring information privacy and security even in cooperating federated networks.

While proactive discovery and federation mechanisms represent effective tools to rapidly restore an IoT registry that application can exploit, in disrupted network environments it is also necessary to specifically design communication protocols to ensure reliable information exchange. To exploit the scarce network resources multiple communication protocols have been proposed to address the issues of tactical environments, leveraging many different paradigms: from connection-oriented solutions that support session mobility and are more resilient to channel fluctuations and disconnections [18], to delay tolerant solutions based on store and forward [19], to adaptation solutions that can run with good performance and reliability levels COTS TCP-based applications without requiring any modification [20]. While typically these type of communication protocols are built on top of the TCP/IP stack, several proposals have started to also include different addressing and transport level solutions to introduce new communication paradigms.

Typically, these types of communication protocols adopt an evolutionary design approach and thus have to address quite a few complications in order to provide applications with a network programming model suited for tactical edge environments while building on top of the TCP/IP stack [21]. However, the newer Information Centric Networking (ICN), instead, represents an interesting clean slate approach that aims at replacing the entire TCP/IP stack with a new communication paradigm. ICN provides several tools to tailor the robustness of data retrieval connections through different mechanisms for signaling redundancy and intermediate data storage (caching). Within an ICN architecture data producers, data location and transport means become transparent. This enables it to seamlessly switch between different traffic patterns, for data meant for single receivers and data meant for a group of recipients, or even to adapt traffic from fairly stable networks to intermittently connected ones.

Several ICN proposals exist with varying degrees of maturity [22]. The most mature, popular, and relevant solution for the purpose of supporting tactical communications is arguably Named Data Networking (NDN) [23, 24, 25], which builds on the concepts developed by the Context Centric Networking (CCN) proposal [26]. In fact, NDN-based solutions have recently started attracting considerable attention from the perspective of military applications [27, 28].

In this thesis, the problem of accessing the IoT resources in disruptive environments is tackled by 3 main points of view: the discovery of resources in absence of Cloud-based registers, the network domains federation and secure information sharing among different authorities, and what are the main characteristics that communication protocols should present to effectively share information in the addressed environments. To this end, the thesis also proposes the project developed to in-depth investigate the 3 points of view and experimentally evaluates the effectiveness of the mechanisms discussed in the various sections. Moreover, on the aspect of evaluating solutions for effective information sharing, also commercial solutions typically adopted for enterprise environments are validated in disruptive network contexts.

The thesis is organized as follows, in chapter 2 the nowadays IoT enabling architecture of urban environment will be unraveled, chapter 3 discuss how disasters affect urban environments and how they relate to tactical networks. Following, in chapter 4 the challenges and requirements to enable IoT in a disrupted environment will be discussed and then, in chapter 5, a proof-of-concept middleware to enable IoT applications in the target environments is presented. chapter 6, will tackle the problematic related to data exchange and investigate the effectiveness of several communication protocols. Chapter 7, further extends the investigation on communication protocols by describing and experimentally evaluating NDN in comparison

with protocols previously introduced. Finally, chapter 7, will conclude the thesis's topics and discuss the discoveries achieved with the work.

Chapter 2

IoT Nowadays Technologies and Infrastructures

The term IoT identifies the collection of communication solutions, protocols, and hardware components that enables the interconnection of a multitude of various physical devices via the internet. Sensors, actuators, computational nodes, and the Cloud cooperate with one another defining a capillary frame of resources that virtually countless diverse type applications such as healthcare, smart cities, smart farming, industry 4.0, intelligent transport systems, smart homes, smart monitoring, smart metering, power grids, and so on [29, 30].

Thanks to its versatility, IoT attracted a multitude of stakeholders that invested and widely adapted this technology to enhance their business or provide new innovative products to their client [31]. For example, metering services have started to rely more and more on IoT for the sensing process and to enable sophisticated leak or waste detection mechanisms [32]. This led to an exponential growth of the number of connected devices since the definition of IoT and it is estimated that the number of IoT devices will surpass 27 billion by 2025.

However, in order to access the internet and share data with other components or users, IoT devices are supported by a plethora of different communicative solutions and protocols designed to fit specific requirements and characteristics that these devices exhibit [33, 34]. More specifically, the "Things", especially sensors and actuators, tend to have several constraints that limit their capability to adopt highly requesting communicative solutions and protocols. For example, IoT devices tend to have limited computational capability and power supply, or in certain cases they must rely on batteries which may cause up and down time cycles, and so on.

To address such constraints IoT stakeholders defined and developed a multitude of solutions and approaches capable of responding to the diverse requirements that constrained devices and IoT applications might present [35]. By following the ISO/OSI stack, these solutions can be categorized in 3 different groups: communica-

tions technologies, which represent the low level communicative solutions that enable connectivity between devices, communication protocols, that enhance the basic connectivity provided by the communication technologies and enable specific communication model and evolved information sharing mechanisms, and IoT management infrastructures. On contrary of the first two, which address the challenges related to connectivity and information exchange between applications and constrained devices, IoT management infrastructures represent comprehensive set of tools that assists and simplify the development of IoT-based applications [36, 37, 38, 39]. For example, IoT management infrastructures mitigate the challenges related to IoT devices heterogeneity and by providing a common interface that applications can rely on to access diverse types of resources.

2.1 Communication Technologies

IoT communication technologies are the plethora of solutions that enable basic connectivity between constrained devices. These solutions emerged as both extension of well known technologies for Device-to-Device communication (e.g. Bluetooth), or newly defined mechanism tailored specifically on IoT requirements which, thanks to their characteristics, became de-facto standards solutions to connect constrained devices to one another.

In particular, since communication technologies represent physical and link layer solutions, they are designed to enable an effective throughput to support timeliness information sharing while maintaining a low profile in terms of power and computational resources consumption. Such trade-off is achieved by balancing diverse properties that the connection might present. For example, wireless solutions might achieve high throughput but limit the connectivity range to reduce the power consumption while others might use more simple information coding, which might limit the throughput, but achieve low power long range communication since the signal emitted is less affected by noises.

Moreover, among these communication technologies some solutions achieve a low profile by avoiding the support of the internet stack and relying on distributed architecture. More specifically, some devices might not have a computational support that enable the devices to directly participate in communication with consumer devices using the full internet stack (e.g. TCP/IP) and thus rely on other nodes that act as mediators between them and applications. In this case, since the constrained devices can expend less computational resources to share information, communication technologies can exploit more devices' resources to enable fairly high throughput and connectivity range.

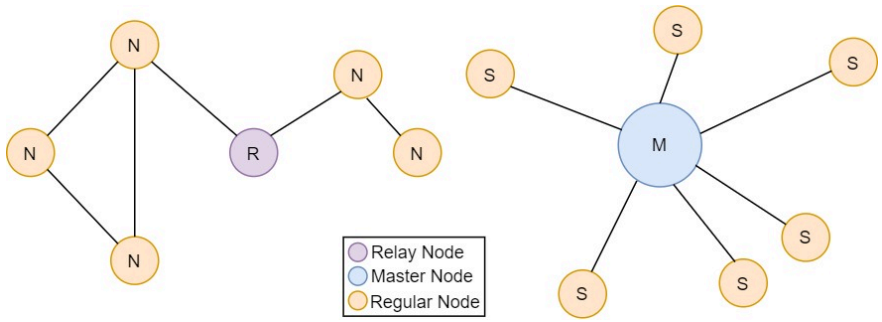

Figure 2.1: BLE Mesh and Master-Slave topology

2.1.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE) [40], also known as Bluetooth Smart, is a low range wireless technology defined as an independent and not compatible extension of the classic Bluetooth. The core difference of BLE from its predecessor is noticeably lower power consumption, however it also supports diverse types of communication topologies, ranging from classic master-slave, broadcast, and mesh, which allow this technology to support multiple network environments. In Fig.2.2 an example of mesh and master-slave topology is given.

Moreover, BLE extends the data transport model of the classic by providing connection-less communication (CL-mode) mechanisms. CL-mode allows devices to send messages similarly to datagram protocols and thus decoupling the two end-points of communications. This type of mechanisms are particularly relevant in IoT network in which devices might have up-down time cycles that might temporarily disconnect them from the network, and enable more idempotent communications that better suit the producer-consumer communication model.

One of the most interesting capabilities that BLE provide for IoT devices is the support of a beacon infrastructure [41]. BLE beacons uniquely identifiable small hardware components that can be connected via BLE. Thanks to these characteristics, these devices allow the definition of a capillary network of beacons that enable various mechanisms such as indoor localization and proximity based notification capable of enhancing the effectiveness of context-based applications.

However, while the BLE effectively enable network of constrained devices, it is not directly compatible with the internet stack, since it adopts its own addressing and packet formats, and require the support of a gateway node between the BLE network and the internet in order to enable the traffic.

2.1.2 ZigBee

Zigbee [42] is a wireless communication technology built on top of the n IEEE 802.15.4 that enables low-cost and low-power wireless mesh networks for constrained

devices. To achieve this, ZigBee defines two main typologies of resources: Full Function Devices (FFD) and Reduced Function Devices (RFD). FFDs represent the subset of resources in a ZigBee network with sufficient computational and power resources that enable them to directly interact with other devices. On other hand, RFDs model devices wiht limited capabilities and incapable to support the full stack of functions enabled by ZigBee and can only be associated with FFD devices. Thanks to this distinction, ZigBee allows a low-energy profile for very constrained devices while enabling a full participation to the network via the support of FFD.

Moreover, ZigBee distincts devices on three more categories: coordinators, routers, and end-devices. Coordinators are nodes responsible for the definition of a ZigBee network and they involve several core mechanisms such as routing, identification of the network, and selecting the operating channel for communications. Each ZigBee network must count a unique coordinator. Routers involved in a ZigBee network comprehend the implementation of the functions necessary to define routing data paths and forward packets in order to extend the range of the network. Finally, end-devices are nodes responsible to send and receive data. Typically, these nodes are connected to sensors and actuators and enable such resources to participate in a ZigBee network.

ZigBee also offers 2 interesting features to automatize the integration of IoT devices and thus be very beneficial for IoT applications. In particular, ZigBee defines precise mechanisms to implement discovery and joining which allows new devices to be integrated and associated in an already defined network. Such capabilities are crucial for IoT network where new devices can be constantly deployed and integrated in a network. For example, a user might dispose of a ZigBee network for a smart home application and integrate new sensors and actuators over time [43, 44].

2.1.3 Z-Wave

Z-Wave [45] is a proprietary wireless communication technology developed by Zensys and later acquired by Sigma Designs that is typically adopted for smart home applications and other types of home related automation. This solution achieves medium-range energy efficient radio communications (90 meters outdoor and 25 circa indoor) at the expense of a low throughput if compared with solutions such as BLE. Since Z-wave defines a mesh network topology nodes can share information from sources to destinations by directly connecting nodes to one another, if they are in range, or by relying on nodes between the two communication's endpoints that can seamlessly forward packets.

Similarly to ZigBee, Z-wave divides the resources in the network into two diverse types of resources: controllers and controllable devices. Controllers are devices that

act as masters for other other devices thus defining a sort of gateway for applications to access the sensors and actuators available, each network must count a primary controller and, contrary to ZigBee, Z-wave enable the presence of multiple controller within the same network as long as there is a unique primary controller. Moreover, controllers implement a pairing procedure to enable the integration of new resources in a Z-wave network. On other hand, controllable devices instead model the edge-resources in the network, which comprehend both sensors and actuators that can connect to the network controllers and send and receive messages.

2.1.4 LoRa and LoRaWAN

Long Range (LoRa) [46] is a modulation technique developed by the Cycleo of Granoble that achieves long range communication while maintaining a low profile in terms of energy consumption. As a trade-off modulation techniques do not offer high throughput communication, however, represent an extremely relevant solution for certain IoT applications. For example, simple sensing devices that can be adopted for smart metering do not require high throughput since certain measurements do not require data streams and can be summarized in relatively small size messages that are then periodically forwarded to consumer services.

Since LoRa is a physical level approach, it has been adopted for a more complete stack of technology for IoT. One of the most successful Low Power Wide Area Network (LPWAN) technologies that adopt LoRa modulation is LoRa Wide Area Network (LoRaWAN). LoRaWAN is Media Access Control (MAC) protocol based on ALOHA which provides a large range communication with low data throughput for constrained devices. In order to connect to each other, edge devices must rely on a network of networks that act as mediator. More specifically, LoRaWAN defines a network topology in which each device sends messages to its local gateway via wireless channel enabled by LoRa, then if the receiver is connected to the same gateway the messages are directly forwarded back to the destination, otherwise, the local gateway forwards the messages to the gateway connected to the receiver via wired channel. As a result, LoRaWAN topology forms a wireless-wired stars-of-stars hybrid network.

Similarly to other technologies, LoRaWAN defines different class of nodes to model different devices and uplink and downlink arrangement: class A, class B, and class C. Class A, represent the most basic type of resources which present limited capabilities, discontinuous availability (e.g. sleep cycles). This class is specifically designed to model the most constrained resources and thus also identifies the minimum requirements that a device must support to participate in the LoRaWAN network. Class B extends the model defined by class A by describing less limited

resources that provide configurable up/down time cycles. In particular, while a class A device's downtime periods are scheduled by the device's hardware characteristics and requirements, a class B device allows network admin to configure its downtimes to match more effective traffic schedules. Finally, class C models devices that do not present any sleep period and thus are always available.

2.1.5 WiFi

WiFi is a wireless communication technology commonly used to connect users' devices, such as desktops computers, laptops, smartphones, tablets, to the internet, it is based of IEEE 802.11 standards, thus making it part of the communication technology defined for Local Area Network [47, 48, 49]. Moreover, thanks to the 802 enabled connectivity, WiFi is also compatible with wired technology that are part of the same family (Ethernet). This allows devices that adopt WiFi, to seamlessly interact with other wired devices in the network, which enable the definition of a highly interoperable network of devices.

In terms of enabled topology, WiFi supports 2 distinct modes: infrastructure mode and ad-hoc mode. Infrastructure mode enables the definition of standard start network topology in which a centralized node, also known as access point, mediates the communications between the nodes connected to the same network. This mode is the most commonly used since it simplifies the protocol. The ad-hoc mode instead allows the nodes to directly communicate with one another. Since the ad-hoc mode poses numerous challenges especially when communication channels require multi-hop communication it is not very common. However, in order to include the possibility of Peer-to-peer (P2P) communication via WiFi, the organization WiFi alliance, has standardized and promotes WiFi-Direct which defines how two devices can share information without the support of an access point but limits the interactions by supporting solely single-hop communications.

Since WiFi has been designed for LAN applications and for devices that typically do not present severe constraints in terms of power consumption nor computational capabilities, this technology is not very suited for limited IoT devices. In fact, this technology is highly demanding in terms of energy consumption and, due to the frequency band adopted, it is not suited for environments in which devices are not in line of sight. However, since WiFi is the de-facto standard for devices that directly interface to users and allow them to access IT services, less constrained IoT devices has started to adopt this technology more often or are supported by infrastructure that enable IoT access to this communication technology

2.1.6 4G, 5G and Beyond

4G is nowadays the most preeminent communication technology to provide internet access via mobile networks. More specifically, 4G is the fourth generation of technologies adopted for mobile networking designed to achieve high throughput (100Mbps in a high mobile network environment) and presents a higher system spectral efficiency, which enables more simultaneous communications on the same channel compared to the previous generation [50, 51, 52].

Similar to its predecessor 4G does not allow edge devices to directly connect to one another and communication must rely on the base stations that provide network coverage. More specifically, cellular networks are formed by dividing the geographical area of coverage in small sections also known as cells. Within each cell a radio base station is deployed which are then connected to one another via a wired technology to form a unique network for devices. Hence if two devices want to share information to one another they must be both connected to the network formed by the base stations which forward messages from the information source to the destination.

4G like WiFi is not a technology specifically designed for IoT devices. However, due to its success for mobile communications, the 4G can result in a solid technology also to support IoT [53, 54]. More specifically, the hardware required to adopt this technology has been tested and optimized by the researches made to improve mobile devices and the cellular base stations form a ubiquitous and already available network. Hence, IoT devices can be enabled by integrating the same hardware components used for smartphones (SIM card, antennae, etc).

While 4G is a solid technology for mobile communication and also for IoT, the fifth generation (5G) plans to directly support IoT applications by introducing a set of features that directly address the requirement of this type of applications. In fact, 5G design is not limited to an overall improvement of network capabilities in term of low latency communications, higher throughput and bandwidth and so on, but also supports IoT by design and introduce new specific mechanisms to support IoT services and applications [55, 56, 57].

More specifically, 5G focuses on 3 macro areas: Internet for mobile via cellular network, like its predecessor, D2D communication and ultra reliable low latency networking (URLL). D2D communications enables constrained devices to directly interact with one another which further reduces latency and power consumption during message exchange. Moreover, due to the radio frequency adopted, 5G organizes the network in a composition of macroassisted small size cells that allows devices to achieve high-throughput low-latency communications while further reducing the power required to transmit packets.

For example, due to the radio frequency it adopts, 5G organizes the network in a composition of macroassisted small size cells that allows devices to achieve high throughput and low latency communications while also reducing the power required to transmit packets. Moreover, 5G is designed to enable D2D communications thus enabling constrained devices to directly interact with one another which further reduce latency and power consumption. Finally, thanks to 5G IoT devices and applications will also benefit from network slicing.

Network slicing is a network architecture that allows to define multiple independent, end-to-end isolated, logical networks, also known as slices, on the same physical network infrastructure. In 5G networks network slicing is further enhanced by enabling different and contrasting Quality of Services (QoS) to coexist on the same wireless link. For example, high throughput best effort communications and low latency ultra reliable communications can be supported simultaneously on the same wireless link. Thanks to this characteristic, 5G is able to seamlessly support different IoT services with highly heterogeneous communication requirements [58, 59, 60].

Even if 5G is still involved in the process of standardization, the sixth generation of cellular network is already being studied and discussed in terms of novel features and requirements that the technology will present in the first phase of definition in terms of requirements. More specifically, this novel approach will certainly improve the common characteristics of the network, higher throughput, low latency, higher energy efficiency and reliability, and so on, in comparison to previous generations but also integrate new features that will enable more pervasive and revolutionary IT services. Like 5G, 6G will include IoT enabling feature that will further extend the applications and connectivity of IoT devices.

2.1.7 NB-IoT

NarrowBand Internet of Things (NB-IoT) is a Low-Power Wide-Area Network (LPWAN) cellular communication technology defined and launched by 3GPP to enable the definition of high-density, low power network for D2D communication [61, 62]. To achieve this NB-IoT adopt a subset of the LTE standard and enables traffic via a single narrow-band of 200kHz which correspond to one single block Global Mobile in LTE transmission [63].

This technology design been to also achieve longer range than typical cellular range coverage radius. However, in comparison to other long range technologies, such as LoRa, NB-IoT does not achieve similar results. More specifically, even if specifically designed for constrained devices, NB-IoT still require constant synchronizations and uses Orthogonal Frequency-Division Multiplexing (OFDM) and

Frequency Division Multiple Access (FDMA) to transmit information, which can be highly demanding in term of power resources. As a result, NB-IoT is not typically adopted for very wide networks and instead is adopted for indoor coverage.

While might not extremely effective in certain scenarios, theses design choices enable the rapid development of a full specification of the technology. Moreover, the reuse of LTE mechanisms also enable the rapid reconfiguration of several applications and devices to this technology.

2.2 Communication Protocols

While communication technologies enable the primary connectivity to IoT devices, higher level approaches are necessary to effectively support the participation of constrained devices to the Internet. More specifically, communication protocols implement mechanisms that address requirements from transport, for low level protocols, to presentation and application level thus enabling more management, coordination and interaction between constrained resources [64, 65, 66, 64]. For example, communication protocols allow to define hierarchical communication schemes that allow better equipped devices to actively support highly constrained devices by exploiting their resources (e.g. data caching).

Moreover, via communication protocols it is possible to define sophisticated communication paradigms capable of decouple applications or services from IoT devices and simplify the interaction between the two. In fact, connection oriented communication might not always be suitable for IoT due to the periodical unavailability of devices and the high computational demand that might be forced onto the communications endpoints. For example, secure information sharing via connection oriented protocols require an initialization phase in which the two endpoints negotiate the encryption mechanism that is most suited to ensure data security. However, this phase is typically based on asymmetric algorithms which can be taxing in terms of computational resources and thus not suited for constrained devices. By decoupling applications and services from IoT devices it is instead possible to avoid these situations while also enabling the integration of other components that mediate the interaction between the endpoints and further support the introduction of highly constrained devices within the Internet.

To achieve this communicative model communication protocols, thanks to their middleware level approach, can implement and take advantage of several ICN concepts. ICN is a communication paradigm that aims to evolve the host-centric paradigm by basing core network mechanisms that allows data exchange, such as addressing and routing, on information related to contents rather than to the communication endpoints. Thanks to this paradigm, the information sharing process

becomes fully independent from where data is located or stored, it can be supported also by intermittent links and can effectively exploit network caching since information consumers can rely on virtually any node in the network [67, 68, 69, 70]. By adopting fully or partially ICN concepts, communication protocols can define sets of characteristics that can effectively respond to IoT requirements.

However, communication protocols for IoT have not solely followed the ICN trend. In fact, introducing a diverse communication paradigm is not always beneficial since it increases the heterogeneity of the IoT communicative stack which poses several challenges during the implementation of services and applications that rely on IoT. In particular, communication protocols largely adopted in nowadays distributed applications, such as HyperText Transfer Protocol (HTTP), have also started to become a common approach for IoT applications. While these protocols present conflicting design choices compared to IoT characteristics, similar to the WiFi case for communication technologies, they represent an extremely effective protocol for devices that interact with multiple services or users' applications since commonly adopted protocols are typically built-in in consumer nodes.

2.2.1 HTTP/REST

HTTP is an application layer communication protocol designed for distributed systems to cooperate and share hypermedia information [71, 72]. HTTP and TCP/IP form the stack of protocol used Web applications. Even if it is typically paired with TCP/IP, HTTP is a stateless communication protocol and follows the request-response (or client-server) communication paradigm and does not support long-session interaction between client and server. More specifically, once a response has been successfully returned to the client, following requests will not have memory of previous exchanges. However, to address session dependent interactions HTTP supports a token based mechanism (Cookie) or rely on step-by-step specifically designed parameters.

While widely adopted, HTTP is not an highly efficient mechanism since its packets are characterized by a case-sensitive textual format which forces the two endpoints to constantly parse and compare sequences of character in order to interact with each other [73, 74, 75]. Moreover, this protocol does not present any constraints in the packet header which allows applications to produce considerably large overheads which require long parsing phases and might not be suited for IoT devices. However, the success of this protocol and the definition of the Representational State Transfer (REST) have made HTTP an interesting and effective protocol also for constrained devices.

REST is an architectural style based design to improve and simplify the develop-

URL	Method	Result
example-api.com/resources	GET	Acquisition of the entire collection of resources
	POST	Insert a new resource in the collection
	PATCH	Replace the entire collection of resources with the new given
	DELETE	Delete the entire collection of resources
example-api.com/resources/{id}	GET	acquisition of the specified resource
	POST	Generally not implemented (405)
	PATCH	Modify the specified resource
	DELETE	Delete the specified resource

Table 2.1: REST compliant API provided by a generic web application

ment of Web applications [76]. REST is based on a few key principles: statelessness, cacheability, and uniformed interface. More specifically, models every information as a resource, each resource is uniquely identified by an URI, and the interaction between clients and servers is limited to few operations: request, submit, update and delete a resource. For example, if a client request for a specific information it can simply specify to the server the resource (via URI) and the specify the appropriate operation it wants to do which, since REST is based on HTTP, can be done by properly using the method defined by the HTTP protocol (GET, PUT, PATCH, DELETE). Table 2.1 presents in detail which result is expected from a REST compliant interface for each combination of HTTP method and URL. Thanks to this design client-server interactions become completely independent from one another, since each possible action on a resource is achieved by a single client request, and a server's interface results uniformed with interfaces of other servers, since each the actions that clients can perform on resources are limited and predefined. Moreover, thanks to the resource-oriented model, caching server results are extremely advantageous with REST. In fact, since contents are uniquely identifiable within the network, caching servers can easily store the resources transmitted, which is a process further enhanced by the stateless nature of the client-server interactions, and simply respond to other clients when the same resource is later requested.

As a result, RESTarchitectural style allows the design extremely effective and simple interaction between devices and thus can result beneficial when adopting HTTP in IoT applications. In particular, the REST/HTTP client results are extremely simple and easily deploy-able in highly constrained devices, meanwhile, on the server side more computational resources are required which results more suited for more complex devices, such as gateways or services. However, the statelessness

nature of HTTP/REST enables their deployment even in edge devices [77, 78, 79].

2.2.2 CoAP

Constrained Application Protocol (CoAP) a request-response communication protocol specifically designed for constrained devices in low-power and lossy-networks [80, 81, 80]. In particular, the main design goal of CoAP is to enable constrained devices incapable to effectively support the HTTP, due to computational or power constraints, to realize RESTlike architecture that can be easily traduced and integrated within web applications and services.

To achieve this, CoAP re-implement the core HTTP mechanisms and characteristics necessary to follow the RESTarchitectural style, in order to obtain a quasi 1-to-1 match with modern web architecture, via a low-overhead binary protocol which require less computational resources. To further reduce protocol complexity and increase communication efficiency CoAP adopts UPD or UDP analogue transmission protocols which are also more suited for unreliable network environments. Moreover, contrary to HTTP, CoAP is designed to specifically address D2D communications offering non-web features such as built-in device discovery, asynchronous message exchange, and multicast support.

Another aspect that differentiates CoAP from HTTP is how the caching mechanism is implemented. In fact, HTTP base caching is enabled by a comparison of the HTTP method adopted to receive information. For example, the acquisition of an information, which in RESTarchitecture is achieved by when a request is submitted using the HTTP method GET, allows nodes to easily recognize the response message as carrier of cacheable content since the method implies that the operation should not affect the server database and thus immediate identical request should produce the same result. Contrary to HTTP, the CoAP trigger for content caching process depends on the code embedded in the response. This strategy allows to define a more tailored caching mechanism since CoAP, like HTTP, describes a wider variety of possible responses in comparison to the limited request types available.

2.2.3 MQTT

Message Queuing Telemetry Transport (MQTT) is a lightweight communication protocol that adopts the publish-subscribe communication paradigm to transfer messages between devices [82, 83, 82]. In particular, the publish subscribe paradigm divides the communication endpoints in two categories: publishers and subscribers. Publishers represent the producer of a certain data while subscribers represent the consumers that will utilize the data produced by publishers. With this distinction publish-subscribe paradigms define a specific information flow and provide an

architecture where consumers and producers are loosely coupled to one another.

However, the main defining characteristic of the publish-subscribe paradigm is the topic based information exchange. More specifically, topics are named logical communication busses in which publisher and subscriber connect to respectively send and receive messages which completely decouple communication endpoints. For example, different publishers can send data via the same topic which may or may not have any subscriber attached and vice versa. Hence, thanks to topics nodes can share information with one another while remaining completely unaware of the network composition. Moreover, pub-sub protocols enable more information centric communications than client-server architecture. In fact, topics can be defined based on the type of content shared, nodes' context, or application specific information thus hiding hosts specific details which are irrelevant for the communications (e.g. IP address).

Publish-subscribe paradigm can be achieved via 2 main topologies: star-topology and P2P. The first topology type, which represents the most common topology among publish-subscribe protocols, introduces a 3 communication participant, named broker, which mediates the message exchange between all nodes as also depicted in Fig.**??**. Thanks to the broker, publisher and subscriber results are extremely simplified. In fact, the broker offers a reliable component that allows nodes to define, discover and connect to their preferred topics to exchange messages. Moreover, since the broker is typically deployed on highly reliable and computational capable nodes, it also implements a series of features that simplify both publishers and subscribers. For example, broker nodes are commonly responsible for information caching and historing, implementation of fail safe procedure, node authorization and secure information exchange. Hence, broker-based publish-subscribe represents an extremely valuable solution for IoT since constrained devices require only to implement the basic mechanisms to transmit or receive messages from the broker.

On other hand, P2P publish-subscribe architecture instead does not rely on a broker and offers a fully decentralized communicative approach. In particular, this type of topology does not present any centralization point that all publisher and subscriber must connect to in order to share data, thus resulting in a more suited approach for a disrupted network environment. In fact, in broker-based publish-subscribe, if the broker node is unavailable or unreachable all nodes connected will not be able to publish nor receive any message. However, the absence of a centralized broker results in a much more sophisticated implementation of both publisher and subscriber which absorbs more computational resources. More specifically, nodes must implement specific mechanisms to discover other participants, which typically involve broadcasting strategies, locate topics of interest, in case they are subscribers, or publicize the locally defined topics, in case they are publishers. Moreover, node

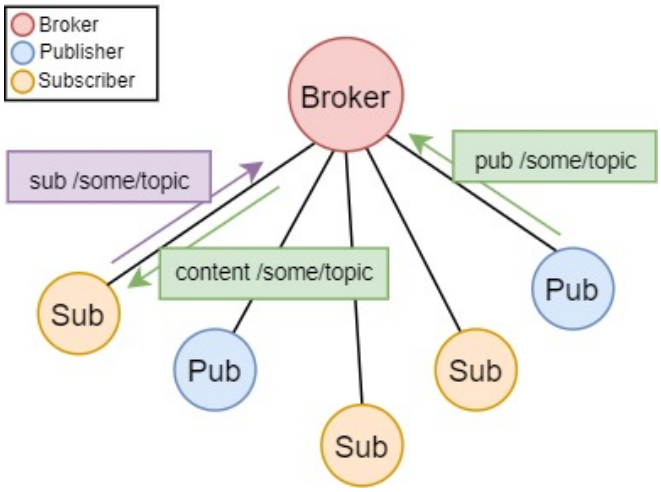

Figure 2.2: Broker Based Publish-Subscribe topology

authorization is an extremely delicate process in a decentralized topology. In fact, while centralized architecture can simply implement in the broker the authorization mechanism required to allow new nodes to send and receive messages, decentralized approaches require more sophisticated procedures, for example, more reliable nodes are able to admit or remove other nodes.

Since MQTT is designed with the objective of reducing both network and resources footprint, it adopts the broker-based architecture with a binary packets format, which reduces both overhead and computational requirements. Via broker MQTT provides three simple and effective QoS levels. The three levels provided are: best effort, which does not offer any mechanism to recover messages lost, at least once, which introduces a simple acknowledgement mechanism, and exactly once, which improves the previous mechanism by discarding retrasmission of an already received message. Moreover, MQTT also offers diverse types of data retention which allows messages to be retrieved also in case of publisher disconnection. More specifically, the MQTT broker cache, if required, messages received from the publishers to allow subscribers to receive specific content even if the original producer is unavailable.

2.2.4 XMPP

Extensible Messaging and Presence Protocol (XMPP) is an application layer communication protocol based on XML designed to enable instant messaging via a decentralized Client-Server architecture [84, 85, 86]. More specifically, XMPP clients, which represent the communication endpoints, interact with one another via a federation of XMPP servers. However, via this configuration XMPP does not define a precise communication paradigm and can support both client server and publish-

subscribe interactions. This protocol does not enable direct D2D communication. More precisely, in client-server mode different clients require the support of a server to share information with one another, and in publish-subscribe mode the publisher or subscriber must be connected to a XMPP server that act as a broker for the nodes.

However, while not providing a fully distributed approach, XMPP does not define any restriction on the introduction of a new XMPP server in the network. Hence, virtually anyone can deploy a server and federate it with other XMPP servers. This results in a federated open system where all nodes can interoperate with one another. This is further supported by the application layer addressing defined by the protocol. More specifically, all nodes can be uniquely identified via JID among all servers in a XMPP network.

Contrary to many application layer protocols, XMPP does not natively support the QoS and does not provide any mechanisms that grant reliable communication between different nodes. However, XMPP is designed to be easily extendable thus allowing application specific requirements to be easily designed and implemented within the protocol. This open design allows the overcoming of other protocol design constraints such as the absence of support for D2D. In fact, protocol extensions such as Jingle allow nodes to interact in a P2P fashion.

2.2.5 DDS

Data Distribution Service (DDS) is a D2D communication protocol designed by Object Management Group (OMG) that provides high-performance, real-time, and interoperable communications for highly dynamic and scaling networks environments [87, 88]. To achieve these characteristics DDS provide nodes that adopt a fully distributed publish-subscribe paradigm that enable the dissemination of data, commands and events.

Moreover, this protocol is not limited to the definition of communicative paradigm but provides applications of a sophisticated middleware that hides the complexity of distributed networks. In fact, DDS manages the discovery of other nodes, implements ad-hoc communication mechanisms which allows node to route and formward messages to other nodes, and presents a multitude of configurable parameters that allows the definition of QoS tailored to each applications specific requirements. Moreover, DDS vendors also exploit its extendable architecture to also provide plug-ins that further specialize this middleware to each specific application and network environment. For example, RTI has developed several plug-ins to improve the protocol effectiveness in a highly disrupted network environment.

DDS is widely adopted in a variety of modern applications such as aerospace

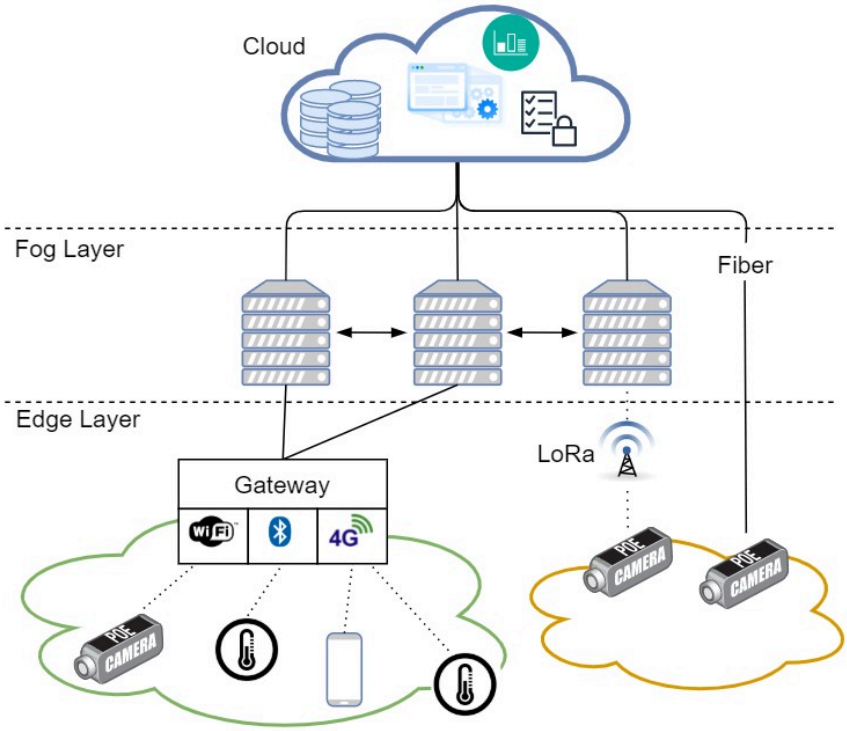

Figure 2.3: Schematic Representation of IoT multi-layer architecture based on heterogeneous communicative technologies

and defense, air-traffic control, autonomous vehicles, healthcare, smart grid management, transportation systems, and others which rely on highly reliable and time critical communications [89, 90].

2.3 IoT Multi-layer Management Infrastructure

Enabling the connectivity of constrained devices to one another and services, which is achieved by the stack of communication technologies and protocols, does not solely enable the effective definition and development of IoT applications. In fact, let's consider for example a smart metering service for a wide urban area. From a top-bottom view, the service represents a sort of centralized state; it can be composed by a distributed cooperation of service components, an element which is accessible by citizens and provides. In the underlying layer, the service is connected on a multitude of IoT sensors, potentially more per each served citizen, that constantly gather physical data and send it to the service via their preferred and supported stack of communication technology and protocol.

However, with this extremely basic architecture, the service must adapt to each possible different composition of communicative solution that each sensor adopts. More specifically, since devices might differentiate on various aspects such as device

type, type of data transmitted, message format, communication technology, communication protocol, and so on, the service will be able to support each possible combination of these characteristics. As a result, in a large environment this approach is not suitable since it requires a virtually limitless amount of resources on the service side. Moreover, in case the service is deployed in a geographical distant location, also the network result is negatively affected. In fact, all devices disseminated in the whole covered area must always send the information directly to the service which congest the network link and obstruct other distributed applications.

In order to tame these negative effects related to the adoption of IoT devices, nowadays network environments adopt a multi-layer architecture that optimize the usage of network resources and simplify the development IoT based services and applications. In particular, IoT is enabled by a 3 layer architecture formed by the edge computing layer, which represent the closest layer to the IoT devices, the fog computing layer, which orchestrate and mediate the interaction between the edge layer and the services, and finally the cloud computing layer, where typically IoT based services are deployed since it offer a virtually limitless amount of resources for computation and caching of the data acquired by the IoT devices [91, 92, 93].

2.3.1 Edge Computing Layer

Edge Computing is a distributed system paradigm which defines architecture in which the computational resources for data management, storing and analysis, are deployed close to the data sources and potentially also consumers. Even if this paradigm was initially introduced for the Web, IoT, which can be already considered an example of Edge Computing, takes advantage of this architecture in order to mitigate the data deluge, device heterogeneity and reduce the overall overhead to transmit sensed data [94, 95, 96, 97].

More specifically, within the edge network the swarms of IoT devices are typically connected and share data to computational resources that act as a gateway between them and the internet. Due to their strategic position, gateways provide a first level of approach in terms of mitigation of communication heterogeneity. For example, if a gateway provides an interface between a public network and a set of IoT devices within a LAN, in order to support all resources it must implement all the communication technologies and protocols adopted within the LAN. Alongside the support for IoT, gateways also offer a support for the applications and services by also supporting communicative stack commonly used by non-constrained devices such as HTTP over WiFi. In this way, applications and services that want to acquire information from the devices can simply rely on the gateway without directly adapting to each possible device of inteRESTavailable in the network. Moreover,

gateways can also provide more high level functionality that further simplify IoT applications or services. In particular, they can implement information caching and forwarding which result in effective mechanisms for devices that present down-time cycles, or basic authorization and authentication mechanisms to provide a first level of security for both devices and applications.

However, since gateways represent an edge computing approach, they cannot solely support all the IoT services. More specifically, gateways tend to be implemented in expendable devices such as Raspberry Pi which cannot offer the computational resources to implement effective data analysis or the storage of large quantities of data. These requirements are instead met in the other layers of the IoT architecture: Fog and Cloud.

2.3.2 Fog Computing Layer

Fog Computing represents a sort of evolution of the Edge Computing approach and sometimes they are considered as synonyms. However, Fog Computing can be described as a more sophisticated application of Edge Computing in which edge computational resources are orchestrated and federated to form a management layer between the edge devices and the Cloud [34, 98]. In this way, Fog approaches can rely on the distributed network of computational resources and optimally dispatch computational tasks in order to effectively implement services.

In fact, Fog Computing effectively supports two core aspects of IoT: the acquisition and dispatching of data to edge devices, and the infrastructure which can host even computational demanding services and applications [99]. More specifically, Fog, which represents an extension and improvement of the Edge Computing paradigm, is able to support constrained edge devices by enabling the deployment of management components, such as gateways, in close proximity of these devices. Moreover, Fog federated approach can scale more effectively in large and dynamic environments since via federated approaches it is possible to easily extend and adapt a management component to the environment [100, 101]. In similar fashion Fog Computing also offers support to terms to IoT services and applications. In fact, the characterizing scaling that Fog Computing can achieve by federating new computational resources allows services to implement highly computational demanding tasks even in close proximity to the edge environment.

While Fog presents advantageous characteristics even in comparison with Cloud Computing, such as the possibility to deploy tasks geographically close to devices and consumers, this latter paradigm is still crucial for IoT. In fact, the fully distributed Fog approach requires extremely sophisticated mechanisms to efficiently coordinate and manage all the resources available. Moreover, Cloud Computing presents several

benefits in terms of cost, reliability, and availability.

2.3.3 Cloud Computing Layer

Cloud Computing is an on-demand computing paradigm in which computing system resources, infrastructure or even applications available in remote data centers are provided to consumers via the Internet. This approach presents several interesting characteristics which make it one of the most prevalent paradigms of the past few decades. More specifically, the concept behind Cloud Computing is to enable the sharing of resources to enable economies of scale and a more efficient usage of network resources in Information Technology (IT) applications. In fact, Cloud Computing allows to define "pay-as-you-go" service model for both applications and hardware infrastructure. For example, a user can "rent" hardware computational resources to support its application as well as use applications running on remote data centers and using its local devices as an interface.

In this way, Cloud Computing offers a scaling business model which is able to constantly meet the applications' and users' requirements. Moreover, since it exploits large data centers designed to offer a virtually unlimited amount of computational resources to clients, Cloud Computing is able to also support highly computational demanding services which receive, analyze and store large amounts of data. This characteristic is crucial for IoT. In fact, thanks to the unlimited computational capability and the scaling service model enable by the Cloud, Cloud-based IoT management service can effectively manage and adapt to the increasing number of IoT devices that populate nowadays networks [39, 102, 103, 104].

Furthermore, IoT management service vendors are also exploiting the cloud to offer ready-to-use solutions for IoT applications via the Internet. In particular, vendors, such as Amazon, Microsoft, SIEMENS and son on, support the development of new IoT applications by providing comprehensive IoT services, named IoT platform, capable to analyze, store, and display data, manage multitude of different devices, support the cooperation between different stakeholders, directly as a Cloud services. In this way, IoT applications only require to deploy and register their own devices and implement, possibly directly on the Cloud, their application specific features without considering any of the constraints and challenges that IoT devices might present.

Finally, one of most relevant characteristics of Cloud-based approaches is the availability offered by this type of solutions. In fact, Cloud computing providers typically exploit the large amount of computational resources of their data center to also implement sophisticated information and application replication mechanisms to rapidly recover in case of crashes or hardware malfunctions. For example, data

and applications status are cached and replicated in different locations so, in case that the application hosting server fails, it can be rapidly re-instantiated in another node and all application functionalities recovered.

Chapter 3

IoT Applications and Disrupted Network Environments

3.1 Smart Cities Architecture

While IoT requires a sophisticated and highly heterogeneous composition of communicative and infrastructural solutions, it also enables the definition of a new era of time-critical, context-, and location-aware applications. One of the environments in which this type of applications has thrived are Smart Cities [105, 106, 107, 108]. Smart Cities, exploited IoT to respond to the pressure that the migration of populations around the world to city-spheres has exerted on city administrations to come up with new solutions to respond to the needs of the people.

In fact, IoT allows urban services to access in real time to a multitude of diverse device, including webcams, weather sensors and stations, different kinds of sensors (pollution, noise, light, traffic, etc.), vehicles and other transportation-related things, and so on, and exploit them to implement featured tailored to each possible context or even scenarios. As introduced in the previous section the connection between urban services and IoT devices is enabled by several different technology and architectural approaches.

In particular, in order to learn about IoT devices, services typically rely on Cloud-based services that offer access and other information about IoT devices within their domain, enabling clients to search for such devices and retrieve their address and other data. Such information is encoded in both machine-, and human-readable format, e.g., JSON, and often includes some form of device identification string, the MIME type of the generated content, the device location, a URL to access the data, and other fields. In this way, both humans via their preferred interface and applications can locate their devices of interest. Moreover, these Cloud-based approaches do not offer direct access to the managed devices, but provide an interface

to query and retrieve the data generated by them.

For example, let's consider an application for police authorities that makes use of traffic camera feeds and image recognition software in order to track down a criminal during a chase. The application requires access (IP addresses, protocols used, and so forth) and other information (e.g., cameras' locations) to be able to connect to cameras located in positions from which they can record the target's movements. Without other solutions, the application must have prior knowledge of all cameras' locations and access information, an approach that is not really suited for highly dynamic environments such as Smart Cities, where sensors can move, sleep, or fail for a number of possible reasons. In fact, this approach would require that either all these events are notified to all applications or applications are designed to handle failures nicely and fallback to other data sources (for instance, if the desired camera is offline, there might still be a lower resolution camera that can record the same area, or other cameras nearby that might still provide useful streams for the purpose of the application). Furthermore, the Smart City infrastructure is typically composed of many different domains where resources are managed and maintained by different organizations, each with potentially different security and access policies.

Hence, Smart Cities services rely on a sophisticated Cloud-based approach that creates a separation between constrained devices and IoT-based applications which hides the complexity and challenges related to the IoT devices and simplifies development and applications. Examples of these services are: Windy (www.windy.com), OpenWeather (www.openweathermap.org), City Bikes (www.citybik.es), Thingful (www.thingful.net), Digitraffic from the Finnish Transport Agency (www.digitraffic.fi), the New York State's 511 Traveler Information System (511NY) (www.511ny.org), LookCAM (www.lookcam.com), and Airly (www.airly.eu).

3.2 IoT-based Services in Disaster Recovery Environments

Even if the primary role of Smart City IoT capabilities is to provide valuable services to its citizens, they also present valuable assets for Humanitarian Assistance and Disaster Relief (HADR) operations. HADR operations take place after a disaster has severely damaged parts of the environment and/or put human lives at risk, which requires local authorities to immediately enact safety protocols to assist the victims and prevent aggravating the current situation.

In these scenarios the emergency responders require constant real-time updates of their surroundings, and, more in general, acquire situational awareness of all the areas affected by disaster. In fact, Major natural disasters occur unpredictably and

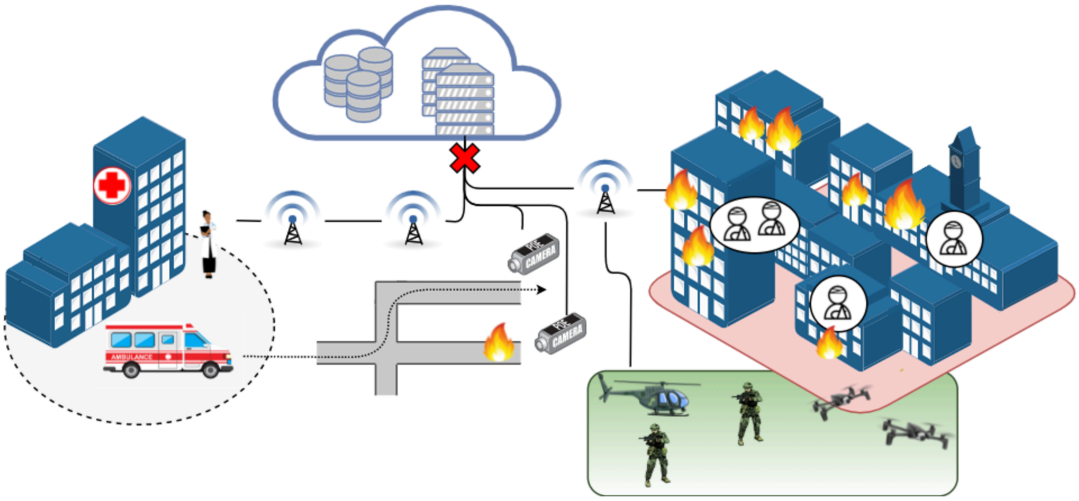

Figure 3.1: CIMIC disaster recovery operation in a urban environment

can wreak havoc in high density urban environments and radically reshape several aspects of a city which can negatively affect the activities of rescuers involved in the HADR operations. For example, earthquakes or floods might inflict damage to the city road infrastructure, thus changing the physical connections between areas of the city. In fact, roads could be interrupted by water floodings, by debris, or by direct damage on the road surface that obstructs the passage of rescue vehicles. Furthermore, the remaining intact roads can be congested due to human movement. More specifically, obstructed roads cause all the city traffic to re-route to the available transportation links, thus overloading any such connection and creating a situation of traffic congestion that further limits the ability of passage of local rescuers toward the areas in danger.

Hence, the ability to exploit local sensors and other IoT assets plays a key role in increasing the effectiveness of HADR operations by improving the situational awareness of responder teams. In particular, the pervasive network of sensors and IoT devices that permeates smart cities can prove to be critically advantageous by enabling rescuers to rapidly acquire situational awareness in severely unstable and dynamic environments, greatly simplifying the phases of planning, organization, and decision making, necessary for effective recovery efforts.

Nowadays emergency response teams increasingly rely on IT services for assistance in the execution of response activities. Hence, Smart cities represent an extraordinarily valuable source of information thanks to an extensive deployment of IoT devices [109, 110]. For example, traffic and security cameras can provide visual information of critical areas, while air and temperature sensors can be used to identify gas leaks and high temperature or fires. Moreover, crowdsourcing and crowdsensing also represents an interesting source of information, e.g., the analysis of social media could also provide valuable situational insights [111]. Thanks to

these resources, rescuers can deploy and rely on *information-centric* emergency IT services that, based on live feeds and analytics obtained from the smart city sensing network, can provide context-specific information and situational awareness. For instance, first responders will be able to leverage IoT-empowered supply chain services to find the closest source of food, clean water, medicines, clothes, or any other basic necessity. Likewise, emergency responders could take advantage of the information generated by medical wearables to locate people that require immediate help. We also imagine that responder squads will carry IoT devices with them to the emergency site, such as drones equipped with cameras and sensors of diverse nature, to help monitor the situation and build a more accurate picture of the event.

However, similarly the smart city physical environment, also the smart city network will almost surely be affected by the natural disaster which can limit and in extreme case prevent emergency services from accessing IoT devices.

3.3 HADR Network Characteristics

The characteristics of the urban network after a large-scale disaster occurs, which is schematically represented in Fig 3.1, can be summarized in two macro aspects: physical fragmentation and administrative fragmentation. This dual nature of the network fragmentation in disaster recovery scenarios reduces the capability of IoT to support the disaster recovery operations and requires tailored solutions to enable the effective organization of the rescue mission and the acquisition of situational awareness.

The network's physical fragmentation comprehends all the communication obstructions caused by the direct damages done by the disaster on the urban network infrastructure. In fact, network infrastructure hardware can be easily affected by physical events. For example, floods, earthquakes, fires, as well as human-made disasters, can destroy 4G antennas, network routers or optical fibers which represent the backbones of urban network communication. Moreover, also the power supply chain can be damaged by the disaster resulting in the unavailability of unaffected core networks nodes which are necessary for the interaction between edge devices and services. Another aspect that further afflicts network connectivity is the situation of network congestion caused by the network conditions. More specifically, when a part of the communication links fails, the remaining links will be often consequently overloaded by the messages that have been re-routed to reach their destination. In fact, internet protocols mechanisms such as Optimized Link State Routing Protocol (OLSR) [112], allow routers to dynamically redefine traffic paths in case of link unavailability. Furthermore, in the hours immediately following a disaster, the network might register traffic spikes, which are caused by survivors

and their families that try to communicate with each other, thus further absorbing the limited network resources.

The network's administrative fragmentation is a consequence of the multi-partnered nature of HADR operations. In fact, IoT operations cannot be carried out by a sole civilian authority (e.g. firefighters) since large scale disasters can affect the urban population on diverse aspects. More specifically, civilians might require assistance from firefighters squads if trapped in under fallen buildings or blocked by fires. However, once free from the dangerous situations, civilians cannot solely rely on firefighters but will also require assistance from medical teams. Furthermore, in case of large-scale disaster, the civilian authorities might not be sufficient to successfully carry out IoT operation. In particular, when disasters overwhelm local authorities, specific branches of federal agencies such as the Coast and National Guard can participate in the relief operations and assist civilian forces by bringing assets and expertise generally only available to armed forces. Military units are well prepared to work in challenging environments and can provideCommand and Control (C2) and Logistics assets adequate to support activities in damaged territories. This type of scenario is also referred to as Civilian-Military Cooperation (CIMIC).

However, the presence of diverse authorities within the same network environment can lead to the definition of a logical private network, defined by each authority involved in the mission, that presents administrative boundaries that limit information sharing between devices. More specifically, these private networks, in which rescuers nodes are connected, might not share information with external nodes due to administrative constraints defined by owners in order to protect from users with malicious intents. Typically, these constraints are implemented by defining a private overlay network that exposes a gateway node which regulates the traffic between the protected and the public network. In CIMIC operations, this situation is further emphasized by the restrictive access policies that military partners specify to grant access to their sub-networks.

These two types of fragmentation negatively affect the whole IoT infrastructure. As previously described, IoT in urban environments often rely on connectivity to the Cloud in order to both collect data published by sensors and to enable applications to send control to the physical environment via actuators. However, IoT network environments cannot reliably enable Cloud-based solutions since this type of approaches rely on reliable WAN to transfer information from the edge to the cloud and vice versa. In these conditions, IoT services and applications that run within the edge network would offer a more robust solution. More specifically, IoT resources might remain available to clients within the same sub-network and thus allowing emergency responders to acquire situational awareness during their missions.

Another important aspect that must be considered to further improve the effec-

tiveness of emergency IT services is the deployment of application specific resources during IoT operations by rescuers. In particular, Even if the environment can provide resources that can support emergency IT services, rescuers might deploy proprietary assets that better respond to their mission-specific requirements and can further improve the capability to acquire situational awareness. For example, Army teams can dispose Unmanned Vehicle (UV), both terrestrial and aerial, equipped with cameras that can autonomously monitor the specified areas. However, also in this case, administrative constraints and the lack of support from centralized common solutions can limit the effective exploitation of these resources. In fact, operators affiliated to diverse teams cannot have prior knowledge of these devices and administrative constraints can obstruct their capability to locate and connect to them, limiting deployed resources to only support a subset of rescuers. Moreover, in case diverse emergency teams are able to share resources, the process of connecting and acquiring information from IoT devices can lead to further undesirable scenarios. For example, since the connectivity between different overlay networks might be achieved with improvised links, such as UAVs configured as wireless relay nodes between different sub-networks, unsupervised information sharing can overload these links and cause network congestion.

IoT scenarios call for specific solutions that can effectively enable IoT in challenging network characteristics. In fact, in order to support emergency IT services, rescuers will likely have to put in place dedicated solutions to discover existing still standing IoT and IT assets in the smart city and integrate with them and deploy, e.g., UAV bridges and/or peer-to-peer smartphone connections to support communications [113, 114, 115, 116]. Bandwidth will be limited and therefore, of premium value, and communications will have to rely on disruption-tolerant paradigms and building blocks. Finally, to exploit all resources of the smart city IT infrastructures that are still available and augment them with newly deployed ones. In addition, the processing of information will likely be performed locally, often relying on edge computing infrastructure solutions either purposely deployed or already in place - such as Mobile Edge Computing (MEC) [117, 118, 119].

3.4 IoT in Tactical Environments

Recently, the effectiveness of IoT in urban environments has led to a growth of interest in the application of this technology also for military environments [120]. In particular, IoBT has the potential to revolutionize tactical environments by enabling a pervasive interconnection of devices capable to share physical measure, share computational resources and supporting the definition of advanced application for data analysis. This will lead to a new generation of tactical network in which IoBT can

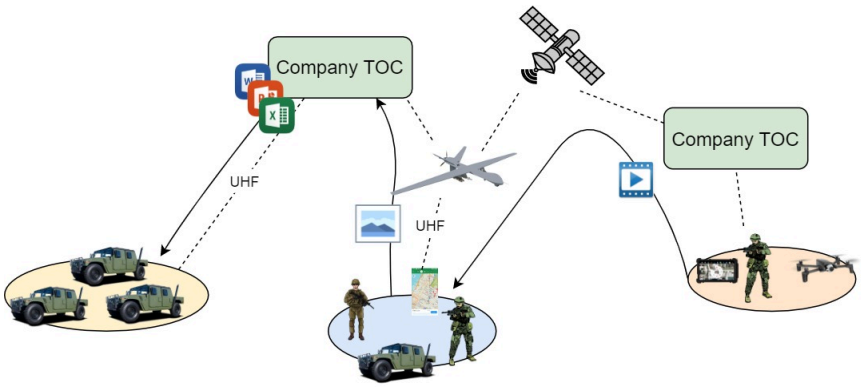

Figure 3.2: schematic representation of a tactical network

provide tactical information superiority and improve war resources management and combat efficiency [121]. For example, soldiers can monitor the battlefield area via their handheld devices which can retrieve information from remote vehicles and aircraft via the internet stack.

However, contrary to urban environments, tactical environments cannot enable IoBT by adopting sophisticated distributed or centralized architecture that highly rely on network capabilities. In fact, connectivity provided by cellular networks or optical fibers may not be available in these environments due to the lack of network infrastructure. Moreover, these environments can also be affected by adversary attacks such as jamming of radio channels, or direct damages to the physical infrastructure further reducing the connectivity between devices.

For these reasons, IoBT typically adopt specific communicative stacks to share information with troops. For example, they present a much more independent architecture that allows users to directly connect, via their handheld devices, to the sensors and actuators. Moreover, in order to improve data availability the transmission of information often involves broadcasting strategy in order to allow different nodes to cache the message and respond in case the data producer is unavailable.

3.5 Tactical Networks Characteristics

Similar to IoT scenarios, tactical environments present challenging network characteristics that limit communication and information sharing and thus obstruct the effective exploitation of IoT devices [122]. In particular, tactical networks are challenging communication environments, typically, as also depicted in Fig.3.2, characterized by the lack of a solid network infrastructure, frequent link disruption, limited and variable bandwidth, vast heterogeneity of hardware and software, and high node mobility.

Another important aspect of modern tactical networks is that they do not have a single administration entity; instead, a number of different domains, e.g., organizations, armed forces, or teams, share resources and cooperate for the success of the mission and each one has control over a portion of the battlefield. Nodes located at the tactical edge include users support devices, such as handhelds, portable PCs, and radios, deployed sensors and sensor gateways, and message-relaying nodes. The use of UVs and UAVs to store and disseminate information across otherwise disconnected portions of the network concretizes situations where the connection's uptime must be exploited to recover and share the information produced during channel downtimes.

Finally, sophisticated sets of software components running at the Tactical Operations Center (TOC) or Combat Operations Center (COC) are orchestrated and organized in order to form Information Management System (IMS). Example of IMS include the U.S. Marine Corps Command and Control Personal Computer (C2PC), the joint Tactical Common Operational Picture Workstation (JTCW), the Air Force Research Labs (AFRL) Phoenix Prime, and theTactical Service Oriented Architecture (TSOA) [123]. IMS store, process, and consolidate data gathered from sensors and units at the tactical edge or generated within the TOC/COC network to produce valuable information, distribute the information throughout the network to consumer applications and other IMS, and provide crucial functionalities in support of the mission, increasing situational awareness and improving operational and tactical decision making.

From a network-centric point of view, tactical networks are typically composed of sensors, handhelds and other portable devices, and (un-)manned vehicles exchanging data via WiFi, Bluetooth, LoRa, 4G, and other radio technologies. Such nodes are often configured to operate without any network infrastructure, i.e., in "ad hoc" mode. At a higher hierarchical level, typically connected to the edge networks via some long range, high latency communication technology like Satellite Communication (SATCOM) or Ultra High Frequency (UHF) radios, the TOC network houses powerful servers running mission management tools [124, 125] and IMS, often connected in high-bandwidth, low-latency Ethernet LANs.

Both levels suffer from different problems related to information sharing. For instance, one of the main challenges at the edge networks is the realization of an effective data dissemination strategy that can maximize information availability across the network, in order to tackle link disruptions and network partition problems, while making an effective use of the available resources and communication opportunities [126, 127].

In these scenarios, IoT infrastructure can rely on both distributed approaches, by exploiting computational resources mounted on vehicles, UVs, or directly ex-

ploiting military handheld devices as constrained devices gateways; or centralized approaches. For centralized approaches instead TOCs can provide a secure centralization point in which to store and forward IoT data. However, both of these approaches present a non-negligible impact on resources consumption, especially bandwidth, computation, and battery, thus affecting access to IoT resources. Therefore, finding efficient solutions that can offer the right trade-off between IoT accessibility and network resource utilization is critical. Moreover, modern warfare IoT applications also require specific resources and information sharing techniques that can guarantee privacy and integrity of data exchanged in multi-domain environments, where different entities need to securely share information with a subset of recipients and resist to possible adversary attacks over the shared network infrastructure. In particular, while increased resources availability at the TOC/COC networks places less concerns on the efficiency of the information sharing process when compared to the tactical edge, the security solution chosen needs to match domain-specific policies to support the coexistence and cooperation of different mission management tools and IMS.

Chapter 4

Enabling IoT in Disrupted Networks: Challenges and Requirements

Both smart cities and tactical networks are an extremely heterogeneous and dynamic type of environment and require specific solutions to enable the pervasive presence of IoT resources which provide location-, context-aware services that can support involved authorities during their mission. In particular, to enable IoT it is possible to identify general crucial requirements and their characteristics that allow the definition of an IoT management solutions for disrupted network environments.

4.1 Assets Discovery

Assets discovery represents the first step to introduce available IoT devices in the network [128, 129]. More specifically, since a disrupted network identifies scenarios in which the IoT infrastructure is fragmented, or compromised, the resources might be available within the edge network but operators are devoid of mediator services that commonly enable access to IoT devices. Due to these conditions, operators cannot rely on well-known services and they might solely rely on resources within their close proximity, which their access information is already known by their emergency applications, and that can be directly connected to. For example, in a HADR environment a Red Cross team that is still located within its hospital perimeter can solely connect to the resources within their facility's network and that information, such as the IP address, are well-known and can be manually input by the rescuers.

Moreover, even in case the network fragmentation does not impede the connectivity to resources to other remote areas, the solely availability of links between different sub-network is not sufficient to fully respond rescuers' needs. In fact, espe-

cially for remote resources the support of centralized services is indispensable. More specifically, while rescuers might possess the access information of all the resources within their facilities' network, for resources deployed in smart cities' public network they must rely on centralized registers or other types of discovery services in order to acquire awareness of the IoT resources disposed by the environment.

Mechanisms that enable emergency applications or services to autonomously locate assets represent a crucial solutions to restore access to IoT devices. In particular, proactive discovery of IoT devices comprehend the process of locating and registration of resources available in the environment without the need of any manual procedure from operators. Such type of discovery can be achieved by deploying emergency computational nodes in strategic area of the city network and exploiting them as discovery services for applications in their proximity [130, 131]. In this way, even if centralized resources registers are unreachable due to the network conditions, deploying specific solutions capable to autonomously locate available resources and respond to applications discovery queries allow to restore assets accessibility within each network fragments. Moreover, these discovery solutions can also be deployed in core nodes that commonly support disrupted network communications, such as gateways between public and private domains, since they are located close to both applications and sensors disseminated in the network. Once deployed, proactive discovery services can adopt both active broadcasting strategy or passive network auditing mechanisms to locate available devices and locally register the devices' access information. For example, the Constrained Application Protocol (CoAP), exposes a specific URI that, when queried, triggers standardized responses containing information on each resource registered in the interrogated CoAP servers.

However, enabling applications to connect directly to IoT resources cannot solely enable the effective use of these resources. In fact, without any other information related to the resources available, applications will be solely provided of information related to how to access available devices but with no information related to their capability or characteristics. Specifically, IoT resources can be summarized and described by the information related to their specific use as well as the communicative stacks adopted to connect to the internet. Information such as the type of a resource (e.g. sensor or actuator), in case of sensor which physical measure is observed by the device, the communication protocols adopted and even the communication technologies supported, can be used to recognize assets' taxonomies in the network, which help applications to effectively locate resources of interest by using more sophisticated or specific search strategies. In this way, if an emergency service does not support a specific communicative technology, due to lack of the required hardware components, it does not require trial-and-error procedure to filter out incompatible devices and solely rely on devices with the required characteristics.

Moreover, with the goal of also integrating resources deployed on-the-fly by diverse organizations, asset discovery should be achieved via an easily configurable and extendable infrastructure capable of adapting to different domain- and application-specific resources [38]. In fact, resources from military organizations, which can be encountered during both HADR and tactical operations, often adopt ad-hoc designed communicative stacks which might differ from the stacks commonly adopted for civilian devices. In such cases, discovery should be able to be rapidly reconfigured and extended to effectively locate all available IoT devices.

4.2 Context-Aware Filtering to Tame Data Deluge

While the recognition of assets' topologies allows emergency applications to locate resources of interest, this type of filtering can still be improved by context information. Resources' and users' context information such as their geographical position within the operation environment can be extremely valuable to both IoT services as well as clients of such services.

To better identify this requirement let's consider a fictional HADR scenario where a group of firefighters is moving toward an area afflicted by a fire. As described in the previous chapter, natural disasters can interrupt roads and deviate the path of the rescuers which delays the assistance of civilians. To avoid these situations, urban sensing devices allow rescuers to periodically monitor their surroundings in order to avoid roads obstructed or congested by the city traffic. For example, traffic cameras allow firefighters to immediately visualize the road conditions giving the opportunity to avoid obstructed roads. However, without any information related to resources' position, the involved operators cannot easily recognize which cameras can effectively respond to their needs. In fact, by providing the firefighters the sole list of traffic cameras the acquisition of visual data of the roads that they will go through to reach the critical area cannot be immediately achieved. Therefore, they must manually research the cameras that they need.

Instead, by enhancing the assets discovery process and retrieving assets' context information it is possible to enable context-based filtering of the resources. In particular, applications can specify the context of the resources in order to retrieve the sole access information of devices that meet the specific applications' requirement. In the previous example, firefighters supported by this enhanced discovery would be able to specify, via their handheld device, a geographical area of the city in which the cameras result are relevant and thus will not require to locally filter the discovery results obtained. Moreover, this type of filtering can also be based on the

relative location of a user in respect to an electronic device used for accessing digital information resulting in a further automatized process.

Context-based filtering can also provide an effective tool to simplify the implementation of emergency context-based services. More specifically, IoT enables the realization of context-, and location-based services that can provide critical support for both HADR and tactical operation. However, the realization of this type of services requires the definition of specific strategies and mechanisms to effectively exploit IoT devices to tailor their own clients contexts. Instead, by adopting and underlying discovery solutions enhanced by also context-based filtering, services can define without any previous support context-based features.

4.3 Network Domain Federation

As previously discussed, proactive discovery must be achieved by exploiting core network nodes of the network in order to respond to all applications scattered among the various network fragments. In fact, nodes such as gateways between different network domains, both physical or administrative, allow to implement a pervasive discovery solutions that restore access to IoT devices within each sub-network and domain.

However, without any federation mechanisms between these nodes, applications deployed within a network fragment cannot effectively exploit IoT devices available within the perimeters of other gateways. In particular, proactive discovery is designed to support scenarios in which the knowledge of the resources available is limited. Therefore, they typically exploit broadcasting strategies, or passive auditing of the network traffic, to locate IoT devices. However, these strategies cannot effectively support multi-domain network environments. In fact, multicast and broadcasting strategies are typically supported only within the perimeter of LAN. Moreover, even if broadcast queries are effectively forwarded between different sub-networks, administrative constraints and firewalls can still limit the effectiveness of proactive discovery strategies.

During HADR and tactical operations, which typically involve diverse authorities within the same network environments, federation mechanisms are crucial to enable cooperative missions. In fact, in both cases, authorities typically deploy proprietary IoT devices to enhance the awareness of the areas of interest. For example, during HADR operations, the Army teams can deploy UVs and Unmanned Aerial Vehicle (UAV)s equipped with cameras in order to safe monitor crumbling building and better prepare the activities pursued by human personnel.

These on-the-fly deployed resources cannot be accessed by operators affiliated with external partners without specific resource sharing mechanisms. In fact, pro-

prietary resources might not be directly accessible from external nodes due to administrative constraints defined by owners in order to protect IoT devices from malicious users. Typically, since the resources might not be able to support complex security mechanisms, these constraints are implemented by defining a private overlay network that exposes a gateway node which regulates the traffic between the protected and the public network. In CIMIC operations, this situation is further emphasized by the restrictive access policies that military partners specify to grant access to their resources. As a result, each team must invest time and resources to deploy their own sensing devices even if already provided by other partners and available within their overlay network. Moreover, due to the diffusion of the IoT, certain environments already offer swarms of resources local organizations have installed within their private network. However, these devices will remain inaccessible since, without specific solutions, they will remain inaccessible from external proactive discovery solutions [132, 133].

Hence, proactive discovery must be achieved via a distributed approach which also supports federating mechanisms that allow, when there are links available, applications to discover and connect to resources deployed within other network domains and extend their awareness beyond their local sub-network. In this way, applications can acquire all information required to properly respond to users deployed in the operation in the field. Furthermore, federation mechanisms become further relevant during CIMIC operations by enabling different teams to share their resources once they reach the critical area [134].

Moreover federation mechanisms can also benefit from context-based filtering. In fact, in case diverse teams are able to share resources, the process of connecting and acquire information from IoT devices can lead to further undesirable scenarios. More specifically, since the connectivity between different overlay networks might be achieved with improvised links, such as UAVs configured as wireless relay nodes between different sub-networks, unsupervised information sharing can overload these links and cause network congestion. Moreover, the congestion of such links does not affect only the communications between different authorities but can also affect communications between devices from the same organization.

However, in this case the adoption of network monitoring solutions capable of periodically supervising the inter-domain traffic it is possible to allow federation mechanisms to also avoid situations of network congestion. For example, by providing users of sufficient information to avoid to connect to resources that, due to the current network conditions, might result unreachable, that can cause network congestion, or that the link capacity does not support the application's QoS requirements (i.e. lower limit to video quality to support image recognition software). Moreover, similar to the resources context information, link's characteristics

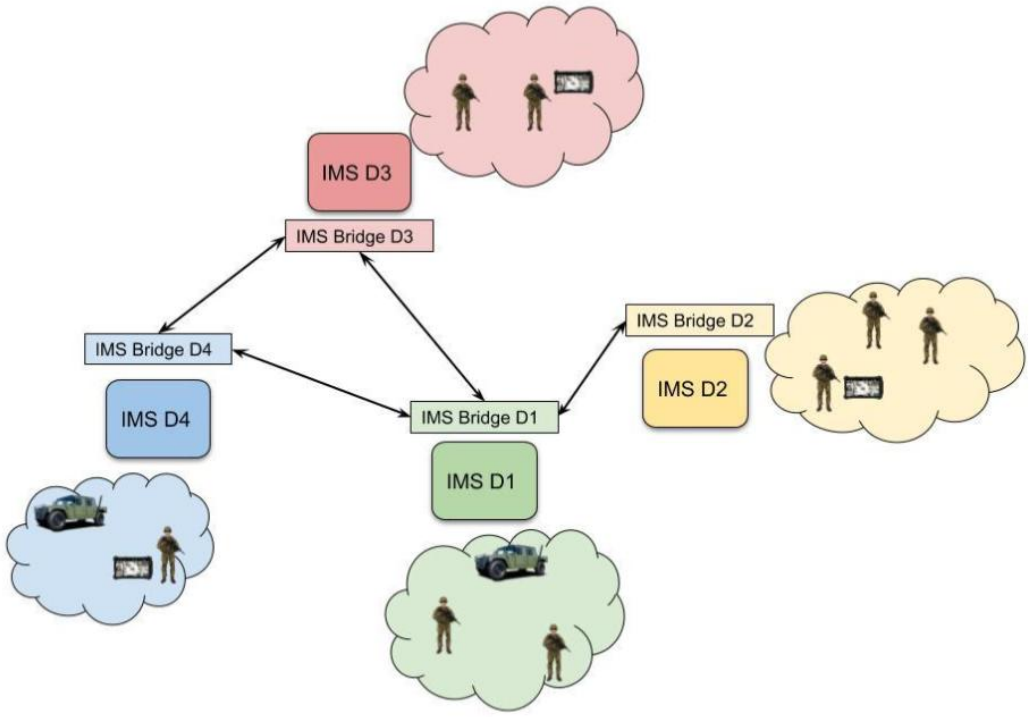

Figure 4.1: Multi-domain scenario involving multiple military authorities

and conditions can be exploited as an additional characterization for context-based filtering of the resources of interest.

4.4 Secure Multi-Group Communication

Since federation mechanisms allow nodes from different organizations to share information it is also crucial to consider specific solutions for secure information sharing. In particular, different civilian and military authorities could, for instance, want to share resources if involved in the same mission but present different security policies, which, if not respected, lead to a situation in which nodes illegitimately access sensitive information.

Fig. 4.1 illustrates an example of a multi-domain HADR scenario where different military authorities, involved in the same mission, have created a federation network to share information. Assuming that each domain has its own edge network, mainly composed of mobile nodes interacting with each other, and a gateway which enables inter-domain communication. The connection between gateways is realized through improvised and unreliable links.

In the example, an operator that belongs to domain D is patrolling an area of interest to scout victims. During the operation, the unit publishes some information,

such as its current position and high quality photographs of the patrolled zone. The local gateway stores that data, enabling mission management tools and other applications in that domain to be informed about the progress of the operation and coordinate the following activities. Furthermore, in order to share data with also other authorities and increase the global situational awareness of each involved operator, the gateway share to other federate the information received from the patrol-

However, without any security layer, data is accessible to also unauthorized consumers. Moreover, there is the necessity to also authenticate the patrol in order to avoid cache poisoning from malicious nodes. This problem is further aggravated in disrupted network environments where the physical links among the IMS bridges are typically provided by ad-hoc wireless channels that can be easily exposed and are often shared by multiple actors. In these situations, especially in tactical environment where the threat of malicious nodes is increased [135], it is crucial to define tools to

There are different security approaches that can enable secure sharing of information in multi-domain environments. A first approach is the use of point-to-point secure communication protocols, such as Transport Layer Security (TLS). Introducing solutions based on TLS in our scenario would secure the connections between the different communication nodes, but it would not suit well disrupted network, which inherently pose threats to the scalability and the reliability of the system [136, 137, 138], and it would not be a good match for securing information sharing in multi-domains environments, where two different communication end-points could not be able to communicate directly, and thus leading us to find approaches based on securing information objects instead of connections.

A different security approach for a multi-domain environment could be the use of secure group communication solutions. In fact, some secure group communication approaches [139, 140, 141], permit to generate ciphertexts that are accessible to all other participants of the group at the same time. Using these techniques, a federate that wants to share some information does not need to negotiate a secure connection to each eligible receiver, but the same information would be encrypted only once, using a shared group key, and then sent to all group members. This design enables to save bandwidth by taking advantage of multicast and broadcast communications, which are usually supported by many network technologies used in HADR scenarios and tactical networks. These solutions typically require a trusted service to provide user, group, and key management functionalities to set up secure group communications.

However, while secure group communication solutions enable bandwidth-efficient sharing of information at the edge, they fail to satisfy all the requirements of a multi-

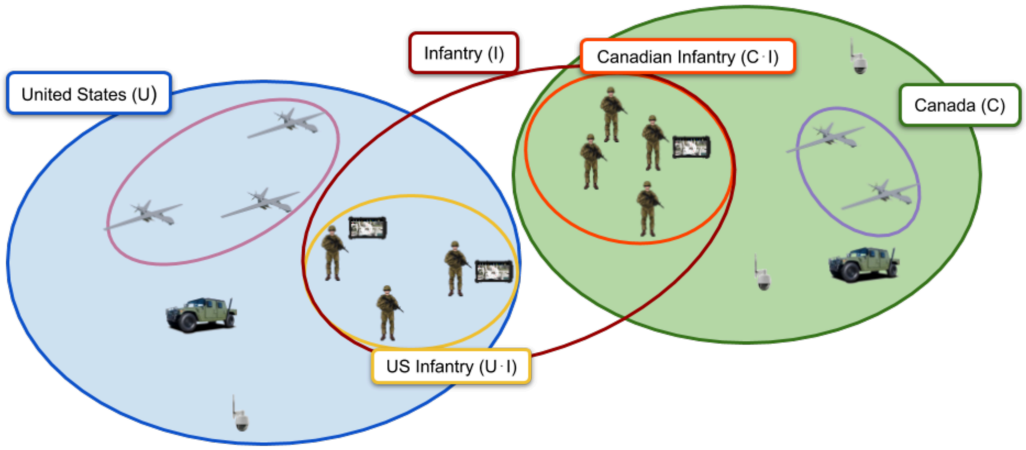

Figure 4.2: ABE application in a multi-domain scenario

domain scenario. In fact, secure multicast/broadcast-based solutions are typically designed for communications within the same edge network, assuming all nodes in the network belong to the same organization, and they do not match the security requirements of environments involving multiple entities located across different edge networks. On the other hand, the efficiency of group communication solutions enables the consideration of approaches that enhance and extend these solutions for multi-domain scenarios.

In particular, approaches such as Attribute-Based Encryption (ABE) for policy-based secure communication enable the definition of an effective, salable and adaptable approach to the security layer for inter-domain information sharing.

ABE allows the management of information across different levels of privacy. More specifically, Ciphertext Policy Attribute-Based Encryption (CP-ABE) [142] allows one federate to securely share information with dynamically definable sets of federates over shared communication channels. In fact, one interesting characteristic of ABE is the possibility to express information access policies using boolean expressions, which are defined as combinations of attributes. This allows only those nodes that are in possession of the right subset of attributes to satisfy the boolean expressions used to secure the information and consequently access it. Therefore, thanks to ABE it is possible to correlate a different attribute with each domain involved in a mission, such as all operators that hold a certain role, all units that belong to the same authority, and so on. Applications can then use specific boolean expressions obtained from a combination of the attributes available to them to both secure the data and specify the subset of recipients that will be able to access those data.

For additional clarity, Fig.4.2 illustrates an example of the application of ABE in a multi-domain tactical scenario. This scenario involves two nations: the United States, identified by the attribute “U”, and Canada, identified by the attribute “C”. Each country makes use of infantry and air force forces, which are represented by the attributes “I” and “A”, respectively. As a consequence, all US infantrymen are identified by the attributes “U” and “I”, while all Canadian soldiers are identified by the attributes “C” and “I”. For instance, if the US wants to share information with its infantry, the ABE expression will be the conjunction of the attributes “U” and “I”, UI, and the encrypted information will be available only to those nodes that possess both those attributes.

Another example regards the communications between two different nations, such as the US and Canada. In order to communicate securely, a new communication group ”UC” needs to be defined. In fact, the attributes ”U” and ”C” cannot be used to secure the communications between the two nations, because they belong to their relative domains and need to remain secret. Secure information sharing between two specific domains requires the definition of a new attribute, in this example ”UC”, which needs to be shared among the subjects belonging to the two different domains.

This approach provides a security model that matches the nature of inter- and intra-domain information sharing in TNs, as it enables senders to secure the information against access from any node that cannot satisfy the boolean expression used for encryption. However, it is not possible to guarantee the transparency of this solution, as users and/or applications are required to specify a boolean expression to transmit data securely.

In order to realize this infrastructure, this solution also requires a trusted third party that acts as a Group Key Management Service (GkMS). The GkMS is a centralized entity, as shown in Fig.4.3, responsible for providing authentication and authorization (A&A) services to the federates that want to join the secure communications group and for the generation of attributes.

In addition, group key management can also be implemented via distributed approach. Centralized GkMS architectures are typically very effective in terms of consistency and manageability. However, centralized approaches have limited scalability when the number of clients grows, a condition that is further exacerbated in TNs by constrained network and computational resources. In addition, frequent link failures and network partitioning phenomena would expose a single point of failure to attackers that want to compromise the security infrastructure.

To tackle these problems,a distributed GkMS architecture can be defined. GkMS instances connect to each other to form a network and keep group keys and nodes A&A information synchronized across all GkMS instances. To manage synchronization, all GkMS network members agree on the election of a single GkMS master

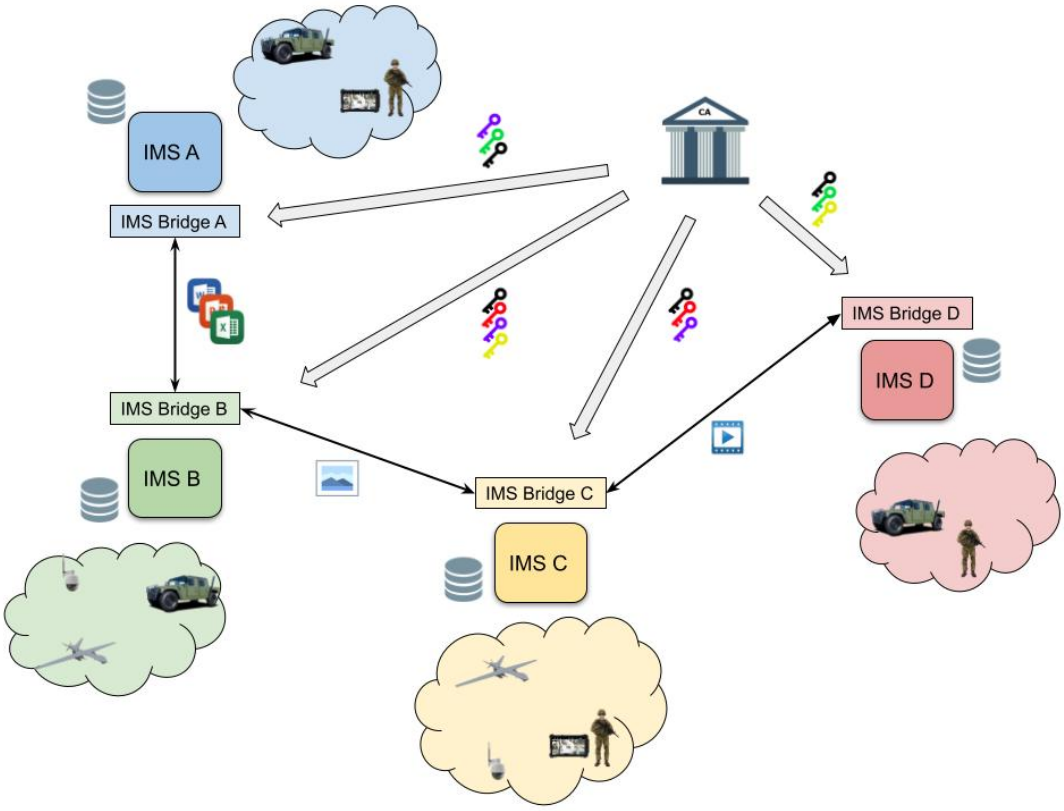

Figure 4.3: Secure group communications in a multi-domain network with centralized certificate authority server

that will make all decisions concerning keys management and nodes A&A. In case one or more GkMS instances loses connection with its master, a new master will be elected among the remaining GkMS instances. Autonomous master election is a key feature to ensure that the security infrastructure can work in presence of extended network partitioning phenomena.

When a communication channel between two previously separated networks becomes available, the security infrastructure must be realigned as soon as possible. In fact, de-synchronization conflicts, expressed as the simultaneous presence in the network of different keys for the same attributes, would make information sharing impossible between nodes that originally belonged to different network partitions even after merging. The synchronization process involves the re-election of a master, chosen by the two conflicting masters, and the generation and distribution of a new version of all attribute keys to all federates.

Federates that want to join the secure network can do so by connecting to one of the GkMS nodes, which could be either a slave or the master and will provide all group key management services to the federate. We refer to the group of nodes connected to a certain GkMS as a "GkMS-Federates Cluster" or, more simply, a "cluster".

As an example, Fig.4.4 shows the application of distributed GkMS to a multi-

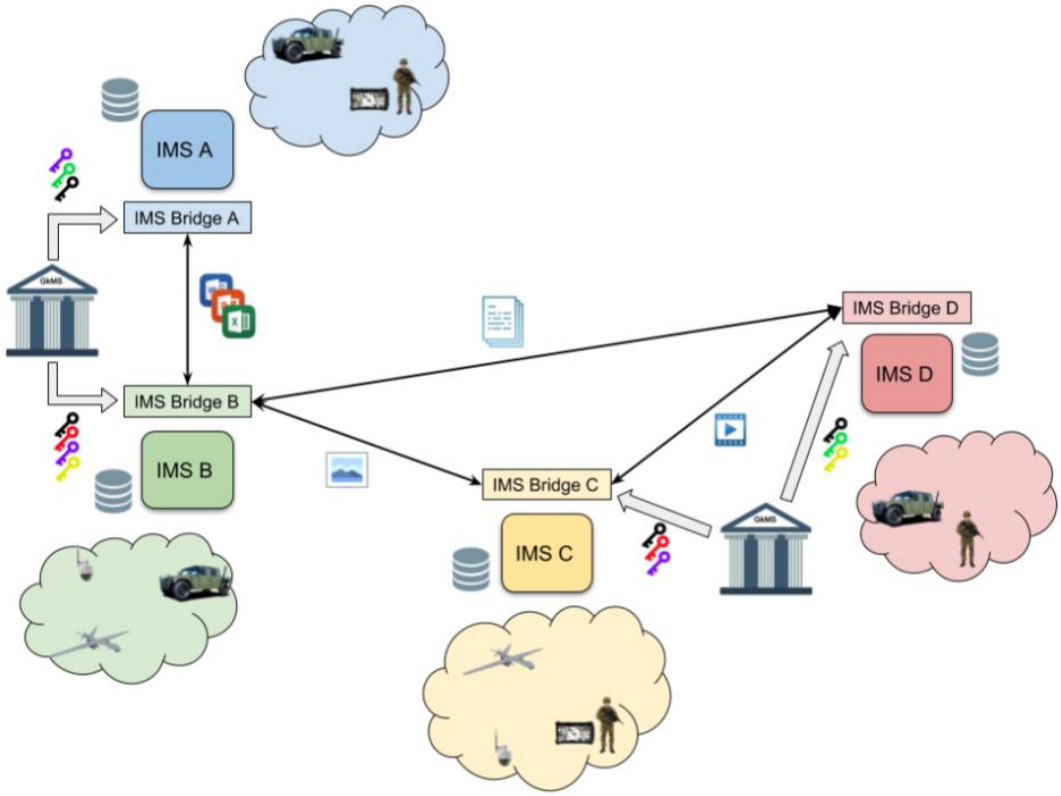

Figure 4.4: Secure group communications in a multi-domain network with distributed certificate authorities

domain tactical network running Federation Services and the security solution based on CP-ABE presented in the previous Section, where keys of different colors represent different attribute keys. In the figure, two GkMS instances, a master and a slave, are running in the network. During a stable scenario in which at least one network route between the two GkMSs is available, the slave GkMS refers to its master to maintain synchronization and determine the attribute keys to assign to the federates in its cluster. This ensures that all federates in the network are assigned the same attribute keys and can therefore securely share information. If partitioning occurs in the network and the slave GkMS gets disconnected from its master, it would detect the master's absence and elect itself as the master of a new GkMS network of which it is the only member. From that moment, and until the two networks merge back again or the mission is over, the new master will decide autonomously on attribute key management, nodes A&A, and deauthorization for all federates in its cluster.

Chapter 5

Enabling Access to IoT via a Middleware-Based Approach: a Proof-of-Concept Solution

Middleware represent the backbone of IoT based applications by managing network and hardware complexity and providing application more comprehensive interface to interact with remote devices. More specifically, middleware for IoT based application comprehend multiple features, implemented in both services and users devices that enables the interactions between IoT devices and applications. Therefore, as a proof-of-concept, this thesis presents a middleware based solution named Multi-Domain AsynchRonous Gateway Of Things (MARGOT).

5.1 MARGOT

MARGOT, that has the goal of simplifying the development of IoT applications. MARGOT permits to discover IoT resources across separated domains and network segments and supports context-aware applications by providing them with a query interface that accepts parameters to refine the search criteria. IoT applications can interact with MARGOT via a JSON RESTful API.

5.1.1 Architecture

The architecture of MARGOT is represented in Fig. 5.1, which shows the major components, i.e., the Discovery Agents, the Information Processor, Federation Services (or Information Management System Bridge, IMSBridge), and the REST API, and the interactions between them. The MARGOT platform is designed in such a way that each instance is responsible for the discovery of resources within one or more domains, and each domain has one and only one MARGOT instance of refer-

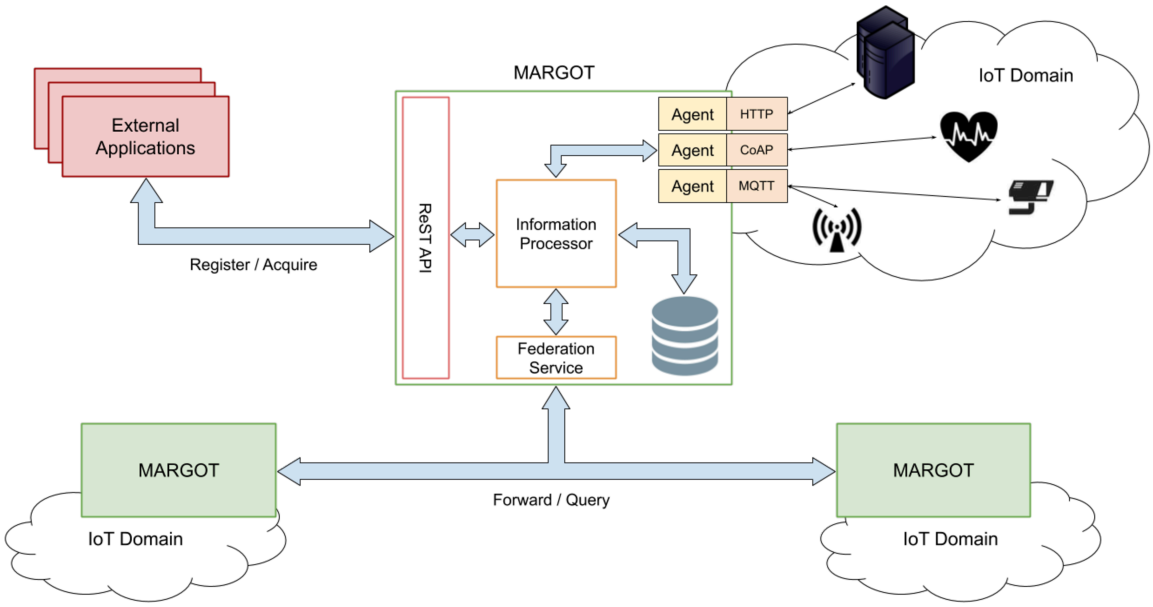

Figure 5.1: MARGOT architecture

ence, which will typically be deployed within the same network or close to it, i.e., geographically and/or in terms of network hops. The knowledge of all discovered IoT resources is distributed across all MARGOT instances in the system, which exchange information via Federation Services (this configuration will sometimes be referred "federated MARGOT instances" or "MARGOT federation").

Discovery Agents

Discovery Agents are pluggable independent software modules that proactively or reactively discover and register all the assets available in the IoT domain. More specifically, Discovery Agentss implement domain-specific resource discovery and store data about the discovered assets in the local MARGOT cache. In order to achieve this, each agent implements, as shown in Fig.5.2, a communication protocol typically adopted for IoT devices, such as MQTT or CoAP, and implements the protocol's discovery procedure, if provided, or other mechanisms, to locate and interact with the available resources within a network domain. For example, CoAP provides a resource discovery mechanism that can leverage on unicast, in case of the Information Processor of a particular resource is known or retrievable via DNS, or on top of multicast by broadcasting a specific look-up message as described in [143]. Instead, MQTT offers specific topics that can be used to subscribe to multiple topics without any previous knowledge.

While, the execution of such discovery procedures is necessary for agents to connect and register a resource within the MARGOT database. However, the effectiveness of these modules is based also on obtaining context information about

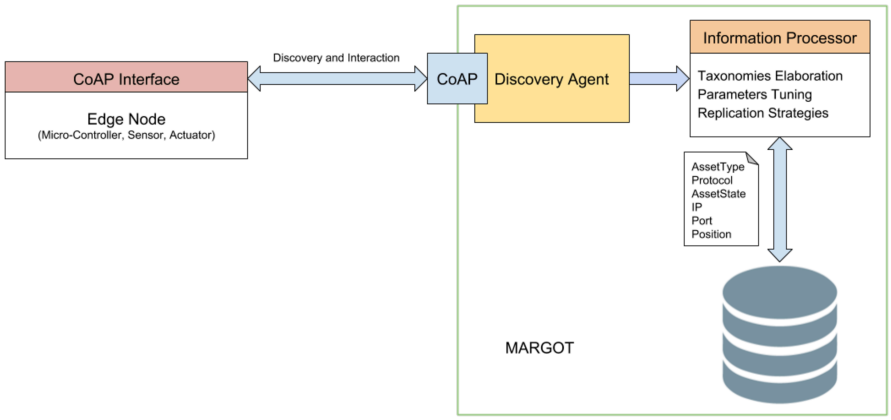

Figure 5.2: Sensor discovering and information elaboration

assets (e.g. position, owner, power supply). This will allows MARGOT to record more complete information about the discovered resources. At this matter, [134] presents how MQTT topics can be used for IoT-based application to achieve interoperability and light-weight protocol based data exchange. These MQTT topics can contain "Request Specification Topics", which can ask for and thus return asset and capability information as JSON payloads from the resources. Moreover, to support the discovery of the resources made available via preexisting services like 511NY or Airly, MARGOT also disposes of Discovery Agents that implements the API of the chosen service. In this way, MARGOT does not replace to preexisting discovery services but cooperate with them in order to enhance the discovery

Discovery Agents can perform the discovery process either proactively, if the process is executed periodically or the implemented protocol supports some form of proactive discovery, or reactively, when specific events trigger discovery, such as a request issued by a client or coming from a federated MARGOT instance. Generally speaking, proactive behavior trades query latency for the freshness of the information. Depending on the rate of user requests and the cost of the discovery process, proactive discovery could lead to either a decrease or an increase in bandwidth consumption. To tackle this matter and provide more efficient proactive discovery strategies, Discovery Agents allow MARGOT to tune parameters that control the discovery process, e.g., the frequency of the process. Protocols that naturally support proactive discovery, such as CoAP and MQTT, can increase the efficiency of detecting new assets, but they still require some form of periodic probing to ensure that previously discovered Things are still alive and reachable. This mechanism specifically addresses Smart City and tactical environments. In fact, both are characterized by events such as the possible deployment of new assets, the downtime of constrained nodes, the deactivation of resources or mobile devices that might

migrate through different IoT domains, which constantly mutate the set of assets disposed by the environments.

Discovery Agents are also responsible for complying with any security requirement imposed by the domains. For instance, an IoT domain could require Discovery Agents to be authenticated and authorized before they can access the domain resources.

Information Processor

The Information Processor is the MARGOT module designed for the processing the data received from the Discovery Agents, storing them in the local database, and handling clients' requests. While doing this, the Information Processor also collects statistics that characterize the domain and user requests, including the variability of the discovered IoT assets and frequently requested resources, and merges them with the statistics received from federated MARGOT instances. Finally, the Information Processor manages data exchange via Federation Services: it decides which IoT resources to publish into the MARGOT federation, forwards user queries if necessary, replies to queries received from remote MARGOTs, and shares updated domain statistics.

The Information Processor is also responsible for tuning the behavior of proactive Discovery Agents and perform other optimizations based on the acquired statistics. For instance, in presence of highly variable local domains, MARGOT will typically request registered Discovery Agents to increase the frequency of discovery, adjust its caching policy by decreasing the expiration time for the resources in those domains, and notify federated MARGOTs about the changes. This will affect the number of user queries that will be forwarded to federated MARGOTs against the number of requests that will be resolved using the data cached in the local database, which in turn will have an impact on the system bandwidth utilization and the accuracy of the information returned to clients.

The module is also responsible for MARGOT *proactive querying*, which guarantees that the information cached in its local database about certain IoT assets is always up-to-date. MARGOT activates this mechanism if the number of user requests for the same set of resources in a given period of time is above a *"trigger"* threshold and preserves it until that number falls below a *"maintenance"* threshold. Both thresholds are configurable and can be tuned by the local MARGOT administrator. When proactive querying for a certain set of resources has been activated, whenever one of the Discovery Agents reports an update to at least one element of the set, MARGOT automatically pushes the updated information to all federates. By doing this, other MARGOT instances will be able reply to user requests that involve any elements in the set directly, without having to send queries to other

federates. Proactive querying affects the amount of data exchanged by MARGOT via Federation in a way that ultimately depends on the number of assets in the set, how often they are updated, the number of MARGOT instances involved, and the amount of user requests received that can be resolved from the resources in the set.

REST API

To respond to clients' discovery requests, MARGOT disposes of an API D2D compliant based on REST architectural style. This design choice is due to the main consumer

MARGOT provides to stakeholders a standardized interface based on the REST [76] architectural style. The implementation of a REST Interface is due to its peculiar characteristics in providing a simple and uniformed interface through a limited set of standard operations to interact and manipulate resources. MARGOT REST interface allows users to recover information about the available assets in the network or to manually register new resources to the MARGOT database. These interactions, in order to provide a M2M compliant communication, adopt JSON as data format to represent the exchanged resources

5.1.2 Data Sharing via Federation Services

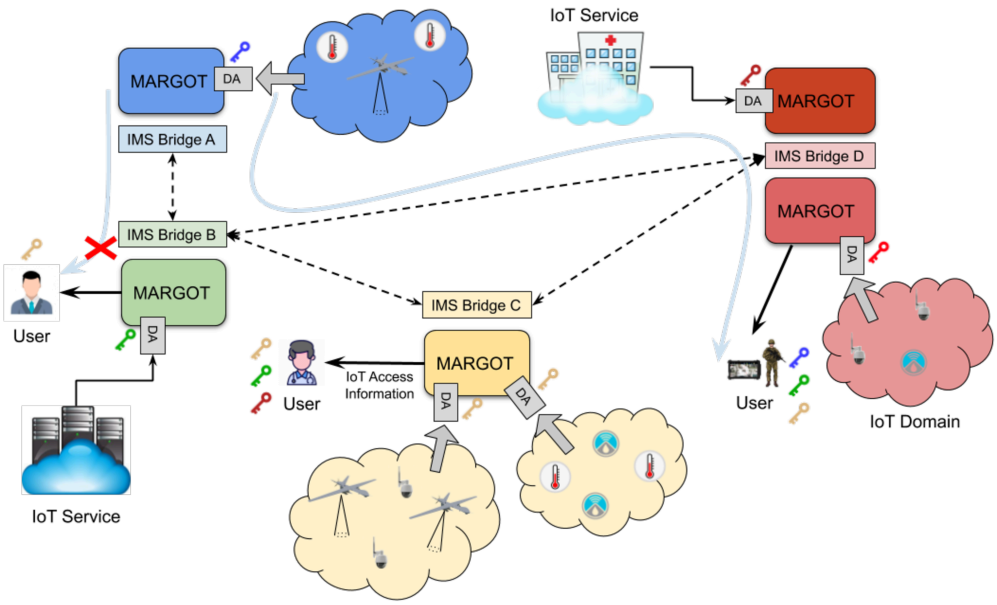

Figure 5.3: MARGOT with Federation Services and ABE security

MARGOT relies on Federation Services [144, 145] to exchange data with other MARGOT instances. Federation Services provide clients of the IMSBridge (federates) with a completely distributed publish-subscribe communications infrastructure

that supports and simplify information exchange in multi-domain scenarios. *Topics* control the routing of data over the Federation network. Topics are named abstractions (i.e., identified by unique strings) that can be thought of as independent communication channels over the network; some topics are typically predefined via configuration files, but federates can also create new ones dynamically at run-time. New messages published within one topic are delivered to all federates subscribed to that topic, which will receive the message directly from the publisher or from another IMSBridge. Clients can join and leave these channels at any time.

Each IMSBridge instance can specify policies that determine what information it is allowed to share with every other instance. This allows one domain's administrators to define rules locally that implement that domain's security and data sharing policies, without the need to coordinate with other domains' administrators. Sharing policies can be enforced using a security system based on ABE, which permits to combine different security layers on a per-message basis and ensures that only clients in possess of the right keys can access the messages. Federates can use ABE to encrypt the payload and metadata of messages before publication, but the information relative to the topic of publication will not be encrypted to guarantee correct routing.

Within Federation ABE is implemented as a proxy fashion service, which is used as a black-box for the encryption and the decryption of the messages. In the implementation, each attribute has an associated unique identifier and a symmetric encryption key, which is shared between the federates that have been authenticated and authorized by the GkMS to possess the attribute.

In order to encrypt messages using ABE, senders use these attributes to specify per-message information access policies. Such information access policy is described as a boolean expression and assumes the form of a sum of products (SoP), where minterms are products of attributes (variables in the expression). In order to decrypt a message, a federates needs to have at least one product of the SoP. For example, the following SoP expression denotes that the message can be decrypted only by entities that possess both the attributes U and I or the attributes C and I.

$$(U \wedge I) \vee (C \wedge I)$$

. When the sender has defined the SoP for the information, the message is encrypted first using a random generated AES 256 bits encryption key, also called payload key. Then, the payload key is encrypted as expressed in the SoP. More specifically, for each conjunction contained in the SoP, the payload key is recursively encrypted using the attributes-related encryption keys specified in the conjunction, and thus creating a set of encrypted copies of the payload key. For instance, encrypting a

message using the above specified SoP will generate two encrypted copies of the payload key, one for each product.

Finally, these copies will be used to specify, along with the SoP encoded as plain text, the header of the encrypted message. In order to decrypt the payload, a receiver needs to solve the SoP specified in the header of the message to decrypt at least one of the copies of the payload key.

Federation Services support other useful features for distributed multi-domain environments. *Distributed queries* allow federates to send and run queries on all or a subset of IMSBridge instances, in order to retrieve published messages that match client-defined criteria. Distributed queries take advantage of metadata information specified during publication to select relevant messages. *Smart synchronization* enables two IMSBridge instances to exchange updated messages after a disconnection period, e.g., caused by network-problems or other issues. Depending on the nature of the messages, all or only the most recent updates will be exchanged.

MARGOT leverages Federation Services' capabilities for all distributed actions, including the discovery of new instances, proactive querying and IoT data replication on federated instances, remote query execution, and the exchange of control information and usage and domain statistics. This enables users and clients to discover resources available across multiple IoT domains and allows the platform to adapt its behavior, e.g., concerning caching and query forwarding, under certain circumstances. Finally, MARGOT leverages Federation Services' policy-based data sharing and ABE security to control the access to IoT assets information.

Figure 5.3 depicts a distributed MARGOT deployment connected via Federation Services. Each MARGOT in the Figure is connected to a single IMSBridge that enables it to exchange data with other MARGOT instances through the Federation network; dashed arrows represent connections between IMSBridge instances that compose the Federation network. Users can connect to any MARGOT instance to obtain access information about IoT assets discovered by any federated MARGOT, in accordance with the data sharing and security policies implemented by the IMSBridge instances involved. For example, on the right side of the Figure, a user with permissions to access IoT asset information for three different domains (represented by the blue, green, and yellow keys next to him) connects to its local MARGOT instance (light red box in the Figure) to retrieve information about sensors in the blue domain. MARGOT translates the request received from the user into a Federation query, which is disseminated from *IMS Bridge D* across the whole Federation network, solved locally by each federate, and finally the answers are relayed back to the node from which the query originated. MARGOT then caches the information locally for future requests and generates a response for the user with the requested information. As a second example, another user that only has permissions to access

information about Things located in the yellow domain also issues a request for assets in the blue domain; however, this user does not have permissions to access IoT information from the blue domain and so he or she will not be able to decrypt the data received from MARGOT.

5.1.3 Main Features and Advantages

The MARGOT platform offers a number of extremely interesting features to IoT application developers, which significantly facilitate and speed up development. Without doubt, one of the most appealing functionalities is that MARGOT grants access to all discovered IoT resources via a single API. This feature can help cutting down development and maintenance costs of applications tremendously. Additionally, the API offered by MARGOT already provides the possibility to formulate selective queries to filter out unwanted resources, for instance restricted to a specific geographical area, domain, and/or sensor type. As a consequence, developers do not have to write the code to handle filtering (or, at least, that code can be simplified considerably) and applications will save bandwidth and battery life by downloading and processing less data, which is especially good for mobile applications. Finally, since MARGOT instances will generally be deployed in locations that enable the discovery of new IoT assets (think about the case of sensors whose discovery necessitates the use of multicast over the local network segment), MARGOT will be able to give users and applications access to resources that they would not be able to discover otherwise, because of network and protocol limitations.

MARGOT can also help IoT service providers at multiple levels. First, MARGOT's caching capability can reduce the amount of traffic that service providers will need to handle. Moreover, the local MARGOT instance will typically be deployed closer, e.g., in terms of network hops, to IoT resources and service providers than users, whose location cannot be predicted or controlled easily, and lower distances tend to increase network efficiency. Finally, MARGOT essentially decouples users' requests for information on IoT resources from their discovery; this allows discovery-related traffic to become independent from the number of system users, thus generating more stable and predictable traffic loads over the sensor networks. All this translates into lower costs for infrastructure, network service, and power consumption for providers.

MARGOT can also help the Smart City administrators by simplifying the management of permissions required to access resource discovery for different domains and IoT services. In future Smart Cities is likely that many services that will offer access to IoT assets will require users to authenticate before being able to call their API. This is already the case today, with services like Airly, which require users

to acquire an API key to pass to each call to enforce service throttling and ensure that the requesting users have the right permissions. With the rise in the number of IoT services and domains that will require authentication, managing permissions will grow increasingly complex for both users and service providers. Thanks to the combined use of MARGOT and Federation Services, it becomes possible to offload some of the complexity to the platform. For instance, they will be able to create separated federations of trusted MARGOT instances within which information can be shared securely without any external access. This would allow city officials, law Enforcement, or emergency response personnel to share access information to IoT infrastructure segments managed by the different administrations without limitations through a dedicated and secured MARGOT federation.

Finally, MARGOT can help during certain HADR situations by mitigating some of the effects of partially unavailable network infrastructures, e.g., due to damage or power failure. In these scenarios, it might become impossible for users to reach the servers of a provider, such as 511NY, but they may still have access to part of the edge and sensor networks. A distributed solution like MARGOT, which replicates data across federates, enhances the whole system's fault tolerance by leveraging redundant instances running in dispersed geographical locations. Therefore, clients are more likely to still have their queries resolved even when the network infrastructure is partially down because they can issue requests to remote MARGOT instances that were unaffected by the disaster. Once an instance receives a client request, even if the MARGOT with the responsible Discovery Agent remains unreachable, it can still respond with the requested data via Federation, as long as at least one federate has cached those data in the past.

5.2 A MARGOT Use Case

To better highlight the effectiveness of MARGOT, lets consider a part of an HADR operation involving a Red Cross and an Army team in the process of aiding injured civilians. When arrived in the critical area both teams define a proprietary subnetwork in which they deploy IoT sensors, such as cameras, several computational nodes, in which instantiate emergency services, and a gateway node that enables communication between the private and the public network. In this way each teams is provided of the essential resources, both hardware and software, to allow operators to acquire situational awareness. Moreover, in their gateway nodes both teams instantiate a MARGOT that respond to the local nodes.

Each operator is equipped with a handheld device to connect to IoT resources and acquire live images of the area of operation. In particular, each handheld device runs the Android Team Awareness Kit (ATAK), an applications designed to

support operations in challenging environments. ATAK allows users to monitor vast geographical areas via a geospatial map that can be enriched by different overlay grids to display real-time events, resources, or other information directly on top of it. Moreover, ATAK is easily extendable with new features and middleware thanks to its plug-in compliant design. In this fictional scenario, ATAK is extended via a MARGOT client plug-in capable to interact with the user to acquire its interests, send discovery requests to the nearest MARGOT instance, and display an resources discovered on the overlay grid.

Therefore, by means of ATAK and the plug-in when the two teams wants to acquire situational awareness can define the area of interest and specify the characteristics that identify relevant resources in the selected region of the map. Then, by connecting with the local MARGOT instance, ATAK sends a discovery request. MARGOT resolve the request by either using the information locally cached, if the information stored are still updated, or it can interrogate the other federates whenever the resource is located in another domain. Once the list of resources that matches the user's interests has been obtained it is then encoded in a JSON message and sent back to the client. For example, the Army team requested for all CoAP temperature sensors deployed in a specific geographical area. Then MARGOT ATAK plug-in parse the message and displays, using a proper icon, all the resources on the ATAK geospatial map. The rescuer can the click on icon of the most suited resource and the ATAK, exploiting the access information received from MARGOT, connect to the devices.

5.3 Experimental Evaluation

In this section 2 experimental evaluation of MARGOT are presented. The first evaluates MARGOT in a single network domain, and how it improve resources discovery in term of response latency. The second evaluation highlights the advantages of the various mechanisms disposed by MARGOT to distribute the discovery results among different MARGOT instances.

5.3.1 Single Network Domain

In this experiment MARGOT is evaluated within an emulated network using the Extensible Ad-hoc Networking Emulator (EMANE). The testbed is composed of 20 nodes connected by network links, which present a latency varying from 40 to 100 milliseconds. Each node plays a specific role in the experiment: sensor, client, or gateway. Sensors nodes represent IoT devices that generate data and communicate using a specific protocol, e.g. CoAP. Clients represent nodes interested in discovering

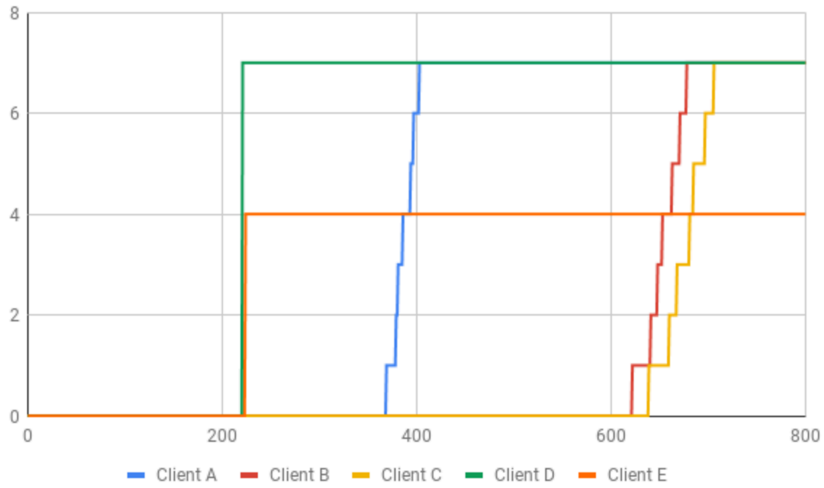

Figure 5.4: CoAP discovery time (in milliseconds).

all or part of the sensors in the network. Finally, gateways are dedicated nodes running centralized services, e.g. information brokers. Therefore, the emulated network counts a total of 14 sensors (7 CoAP and 7 MQTT), 5 clients, and 1 gateway node. Furthermore, the results also present the effectiveness of MARGOT in terms of information prefetching and interface enhancement by connecting our emulated network to a Cloud service, namely 511ny.org[1] that provides information about public traffic camera in New York City, NY, USA.

CoAP sensors communicate via TCP, UDP and UDP Multicast to allow multicast discovery. These resources were initialized with the attributes defined in the CoRE link format that describes the sensor in terms of type, MTU, and the URI that can be used to retrieve the data. For the experiment these attributes were exploited in order to enable clients with specific requests to perform context aware resource discovery, if possible, and to enable MARGOT to identify possible resource taxonomies.

MQTT sensors connect and periodically publish data to the active broker running on the gateway node. Each sensor has a different transmission frequency that varies between 4 and 12 seconds. Sensors do not share the topic except for the part of the topic describing the resource type. For example, `/nodemqtt5/companyA/camera` or `/companyB/temperature/roof`. The topic naming scheme might be different from node to node in order to simulate the presence of different owners, who may adopt mismatching naming schemes.

Client nodes on the other hand were either applications implementing the mechanisms required to retrieve information about the available sensors in the network (client A, B, and C) or that interrogate MARGOT to fetch information about dis-

[1] 511ny.org, available online at `https://511ny.org/`

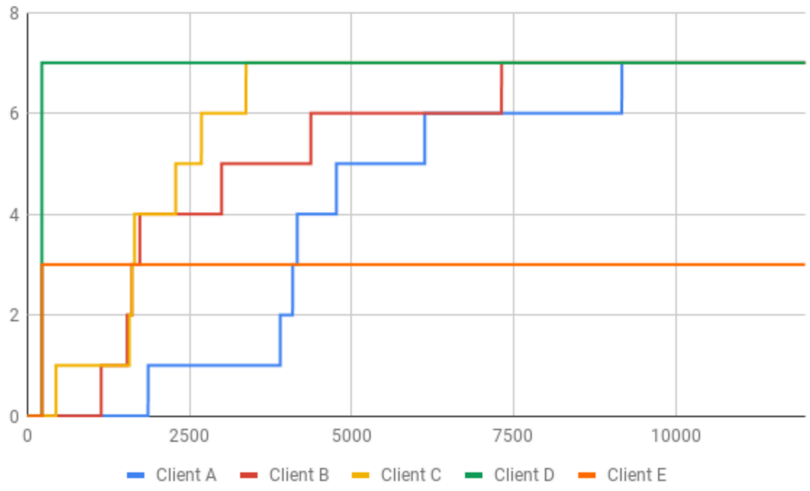

Figure 5.5: MQTT discovery time (in milliseconds).

covered sensors (client D and E). The gateway is instead a node responsible to run a MARGOT instance and the MQTT broker (in this experiment Eclipse Mosquitto[2]).

The clients that do not use MARGOT instead rely on the intrinsic mechanisms provided by the communication protocol to retrieve information about the available sensors. More specifically, to discover the CoAP sensors, Clients A, B and C perform a multicast request to the URI `/.well-known/core`, which is the URI specified by the protocol to retrieve information about the registered resources on the CoAP server. For the experiment multicast was the preferred choice, since the clients do not have prior knowledge about the addresses of each sensor. MQTT sensor discovery instead was performed by exploiting the MQTT wildcard mechanism. In fact, by subscribing to the topic `#`, each client will receive all messages that MQTT sensors publish. This process stopped when at least one message per topic has been received. Both CoAP and MQTT procedures are also performed by MARGOT Discovery Agents to discover the sensors of this experiment.

The results obtained compare the total time required by each client to discover the sensors in the emulated network. To do that, the cumulative number of sensors that were discovered over time is determined in order to highlight when each client acquires information about each sensor. Fig. 5.4 depicts the necessary time for each client to obtain information about all the available CoAP resources. Clients leveraging on the CoAP discovery mechanism require between 400 to 700 milliseconds to have a complete list of the available CoAP sensors. On the other hand, clients that requested the list of available sensors from MARGOT obtained the same results in approximately 200 milliseconds. In addition to MARGOT clients having a shorter wait time, what is more interesting is the capability to obtain an instantaneous

[2]Eclipse Mosquitto, available online at `https://mosquitto.org/`

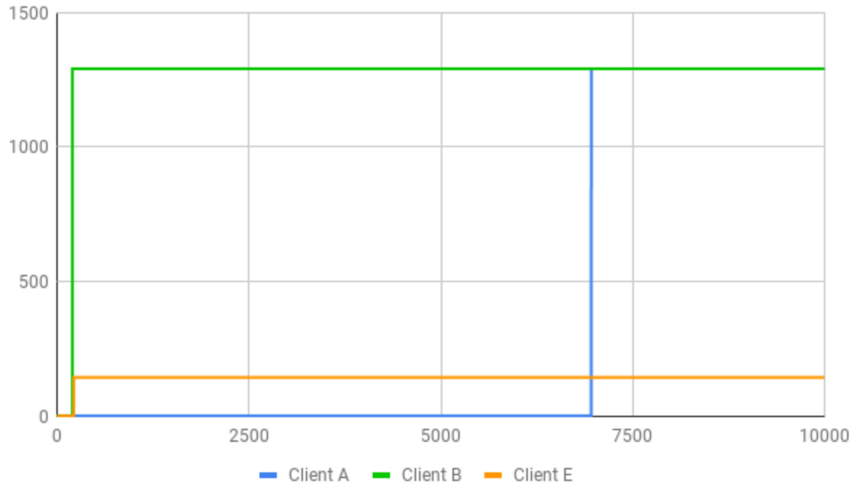

Figure 5.6: HTTP discovery (in milliseconds).

snapshot of the available sensors in the network. For instance, in a real scenario, Clients A, B, and C would have to indefinitely wait for discovery replies given that the number of sensors is unknown. Furthermore, the results from Client E show a filtered request to MARGOT and thus received information about only the resources that fit its specific interests (four in this experiment), without the need for filtering the response locally.

The time required by each client to discover every MQTT sensor is shown in Fig. 5.5. To collect these results, the clients have been started asynchronously in order to highlight different results that our scenario might present. In fact, due to the passive nature of the discovery mechanism the overall time to discover the MQTT sensors depends on the sensor publication frequency and when the time when a client subscribes to all topics. The longest time required to receive information about all publishers is 9.1 seconds but in the worst case it might take until 12 seconds to discover every single MQTT resource since it is the longest publication interval among all the publishers. Clients that instead use MARGOT to discover the MQTT sensors waited only about 200 milliseconds. As in the CoAP case, clients that adopted MARGOT obtained complete information about the MQTT resources without any uncertainty based on the time when they subscribe. Furthermore, when using MARGOT, the clients do not need to subscribe to topics that do not fulfill their interests. This is even more evident for client E that performed a request limited to all the topics containing a specific keyword.

Finally, Fig.5.6 presents the time that nodes required to interrogate a remote cloud service to obtain information about traffic cameras. In this case, only 3 clients were involved, of which only one is directly connected to the remote service. Fig. 5.6 shows the HTTP request performed by Client A has been resolved in 6.9 seconds by

the remote service. This period is not just related to the network latency but also to the computation time required by the remote service to fetch all the requested information (there were 1290 cameras available). Instead, the clients connected to MARGOT received a response within 200 milliseconds. As shown for the other experiments, Client E performed a filtered request to MARGOT and thus was served with limited amount of information. The same result would be impossible in other ways since the remote service does not provide methods to perform specific requests, e.g. cameras available in a specific area, again underlying MARGOT's effectiveness.

5.3.2 Multi-Domain Network

The evaluation of the effectiveness of a Federation of MARGOT in a multi-domain scenario is achieved via three separate experiments. Also in this case the experiments has been conducted in an emulated network created using EMANE, in order to control latency, bandwidth, and packet loss of the emulated network links. The emulated network, which is also depicted in Fig. 5.7, consists of 3 separated networks named: Domain A, B, and C, respectively, connected via EMANE-controlled links. More specifically, the link between Domain A and Domain B presents a latency of 30 ms; Domain A is connected to Domain C via a 80 ms latency link; finally, the link that connects Domain B and Domain C has a latency of 100 ms.

Each domain has a node running MARGOT that can acquire information about IoT resources that the instances running in the other domains cannot directly obtain. To do so, Discovery Agents in each MARGOT has been configured to interface with one of the three Cloud IoT services: NY511, Airly, and Digitraffic. In addition,

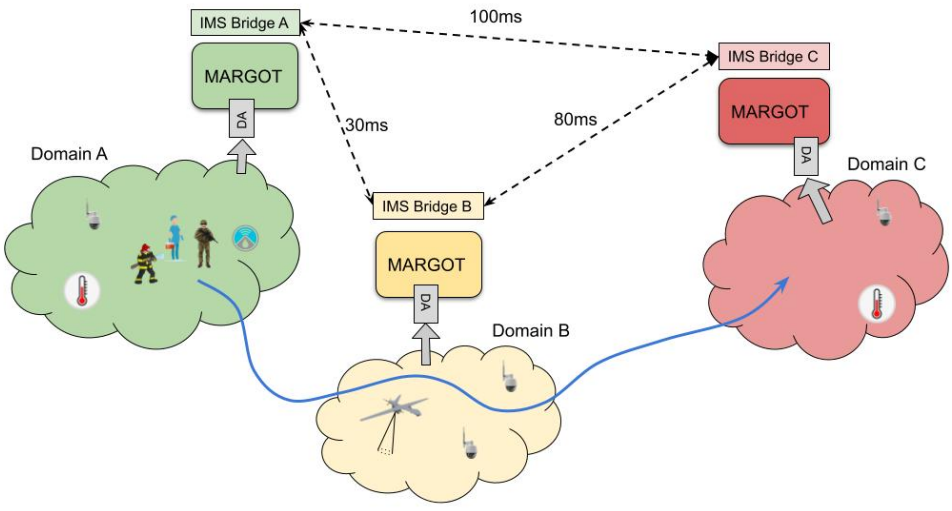

Figure 5.7: Emulated multi-domain network experiment

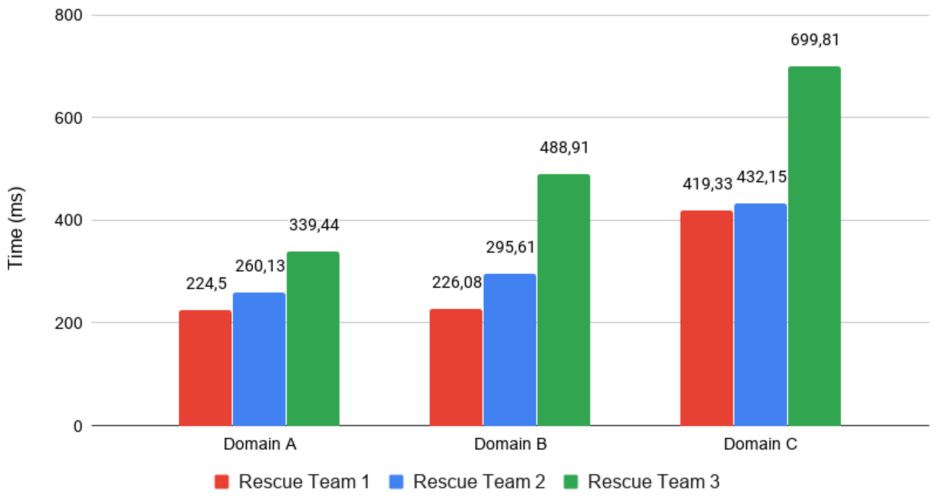

Figure 5.8: Average response latency per domain with MARGOT caching disabled

each MARGOT is connected to a local IMSBridge that can federate with the other bridges via the EMANE-controlled links. As the results will show, MARGOT clients are able to connect to any MARGOT instance and retrieve information about IoT resources from any domain, as all MARGOT instances are federated.

In the scenario, three clients, which represent three rescue teams (Rescue Team 1 through Rescue Team 3), move from one domain to another and periodically query the local MARGOT instance to retrieve access information about devices that satisfy their interests. For simplicity reasons, each client issue requests for the same subset of sensors each time: the first team generates requests for 10 sensors, the second team receives information about 50 sensors, and the third one about 250 sensors. To obtain avarage measurement the experiments have been performed multiple times. In particular, first two experiments 100 times for each combination of Rescue Team and Domain, measuring the average response time from the client application. For the third experiment, since it is designed to evaluate the effectiveness of the proactive query which is a dynamic behavior, the experiment is composed by only 2 runs with the duration of about one hour each, the first time with proactive query disabled and the second time after enabling that feature.

In the first experiment measures the average time that each rescue team in each domain had to wait to receive all the requested data without any use of caching within MARGOT; the goal was to measure the expected latency whenever new resources are queried for the first time or the cached entries are expired. As a consequence, for this experiment all MARGOT instances always forward the clients' queries to the other domains via Federation. Figure 5.8 shows the measured latency for each team receiving data from each domain. The first rescue team register the lowest latency since their requests involve less traffic, but the growth is less than

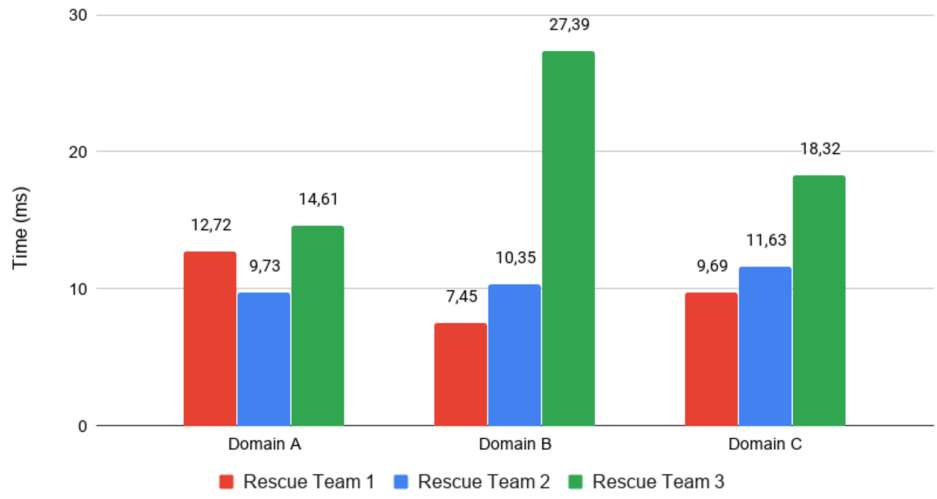

Figure 5.9: Average response latency per domain with MARGOT caching enabled

linear (the number of assets queried increases five-fold when going from Rescue Team 1 to Rescue Team 2, and from Rescue Team 2 to Rescue Team 3, but the measured latency only increases by a fraction of that, with 65.4% being the highest increase), which suggests that the processing and forwarding of the queries has the largest impact on the total latency. Moreover, all teams present a higher average latency when connected to Domain C; this happens because the local MARGOT database contains a much larger number of entries compared to the other MARGOT instances, which increases the processing time.

The second experiment is similar to the first one, but in this case the caches of all MARGOT instances are primed with the necessary data and then configured them to solve all clients' queries from the local database. The results, shown in Fig 5.9, are particularly relevant for scenarios in which clients request information about "popular" resources, which would often be resolved from cache, and in cases where the requested resources are fairly static and, consequently, high cache validity timeouts have been set for those resources. As the Figure shows, caching allows the system to reduce response times by one order of magnitude. As the response latency decreases, now ranging between 10 and 20 milliseconds, other factors, such as delays in the communications caused by the TCP protocol, thread scheduling and context switches, memory access, and others, whose impact in the first experiment was negligible, now have a visible effect on the measured response times. The cause of the increased delay experienced by Rescue Team 1 and 3 is related to their positioning within the perimeters of Domain A and B, respectively.

The final experiment is designed to show the effects of MARGOT's proactive querying on bandwidth consumption. This experiment only involves Domain A and Domain C; two static clients are instantiated in Domain A that send a total of 5

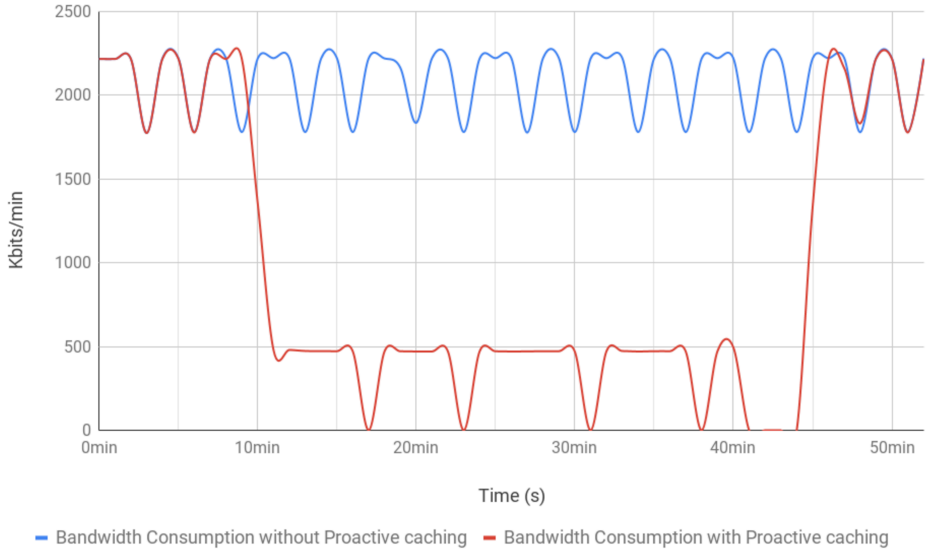

Figure 5.10: Bandwidth consumption comparison between reactive querying and proactive querying

requests per minute to retrieve data on the same subset of IoT resources located across the two domains, for a total of about 150 sensors. The experiment is repeated twice: in the first run, proactive querying was disabled (reactive querying), whereas, in the second run, the MARGOT in Domain A was configured to enable it (proactive querying). In the second run, after about 10 minutes, the frequent requests for the same set of resources trigger proactive querying, which enables the MARGOT running in Domain A to send a special request to the MARGOT in Domain C, after which the receiving MARGOT will start to push proactively any update regarding any of the resources that have been requested frequently.

The results obtained are shown in Figure 5.10. The graph shows the amount of traffic generated by the two MARGOT instances every minute with (in red) and without (in blue) the proactive querying optimization enabled. After the first 10 minutes and until one of the clients starts making requests for different sensors (which happens about 40 minutes after the beginning of the test), the bandwidth consumption in the second run decreases to an average of 500 Kbits per minute against a baseline traffic of 2.2 Mbits per minute measured during the first run. This reduction is caused by two factors: first, the absence of query messages, which do not need to be forwarded between federates when proactive querying is active; and second, a reduction in the amount of data sent, since updates are limited to the resources that have changed state. Note that proactive querying also reduces the latency of requests to values very close to those shown in our second experiment, as all client requests for assets updated proactively will be resolved by MARGOT

from the local database.

Chapter 6

Information Exchange in Disrupted Network Environments

The acquisition of IoT devices' access information represents the first of the two major aspects to enable IoT-based applications in disrupted network environments. In fact, once IoT devices are accessible by consumer applications it is also crucial to investigate effective mechanisms to exchange information. Moreover, this aspect also is crucial for network domain federations. Specifically, while certain IoT devices might force the use of specific communicative stacks, the diverse federated gateways must adopt tailored communication protocols to effectively exploit the scarce network resources.

As introduced in Chapter 2, communication protocols that implements information centric concepts represent a valuable approach to IoT. In particular, producer-consumer decoupling achieved by these communication protocols represent an extremely valuable feature in IoT network sine constrained devices might be inactive due to power constraints. Similar to these situations, in disrupted environments, in which nodes might be disconnected due to links' conditions, producer-consumer decoupling also allows to mitigate network conditions. Moreover, protocols paradigms such as publish subscribe that also enable group communication can represent a valid solution for disrupted network environments. Group communications effectively reduce impact on bandwidth for one-to-many communications, which are often adopted to spread critical messages or context updates to allies in tactical environment, and also allow to better exploit distributed caching mechanisms that increase information availability. Group communication is not limited to tactical environments. Urban environment are adopting protocols that provides this feature to improve dissemination of IoT information.

However, the protocols adopted in tactical networks differs from the protocols for urban environments since the latter are designed for reliable networks. But since in case of natural disasters the two scenarios presents several commonalities,

in order to define a common approach for disrupted network environments it is crucial to design and study communication protocols that effectively responds to the IoT applications and constrained devices. In particular, both specifically designed and commercial off-the-shelf (COTS) communication protocols have been refined with the focus to tackle disrupted network characteristics and enable and reliable information sharing between nodes. Typically, these protocols are implemented on top of the TCP/IP stack. However, ICN protocols such as NDN presents several interesting characteristics that might result effective future standard to enable IoT in disrupted network environments [146].

Therefore, in the large panorama of different communication protocols is crucial to understand the fundamental characteristics that can help stakeholders to correctly choose the correct solution based on their targeted environment. To achieve this, this chapter presents the experimental evaluation of several communication protocols within the Anglova scenario.

6.1 The Anglova Scenario

The Anglova Scenario is a purposely designed testbed developed by the NATO IST-124 Research Task Group (RTG) on Improving Connectivity and Network Efficiency in Heterogeneous Tactical Environment. The scenario allows to evaluate the performance of communication solutions in a realistic tactical environment in a controlled and reproducible way and consists of an emulated network environment that depicts an operation conducted by a battalion after receiving reconnaissance data alerting of an attack by insurgent forces against coalition forces in an operational zone. Due its characteristics, the NATO STO IST-161 RTG has been using the Anglova scenario [147] [148] to evaluate the relative performance of a variety of Group Communications Protocols to disseminate information within a tactical domain [149]. The network connectivity, which is emulated via EMANE, is enabled by a wide range of wireless communication technology such as High Frequency (HF), Very High Frequency (VHF), and Ultra High Frequency (UHF) radios, UAV systems, and Satellite Communication (Satcom).

The emulated operation is composed by three separated phases also referred as vignettes and counts a total of 283 nodes. The first vignette covers all the intelligence preparation action such as sensor deployment and information gathering. In particular, the vignette comprehend the deployment of sensors which share data via Satcom links and UAVs, the transit of UAVs in different area to disseminate the data from sensor, and also it emulates the presence of naval support for recognition and surveillance. The second vignette describes the deployment of the coalition forces. In this vignette the forces move into the operational zone and so the connection

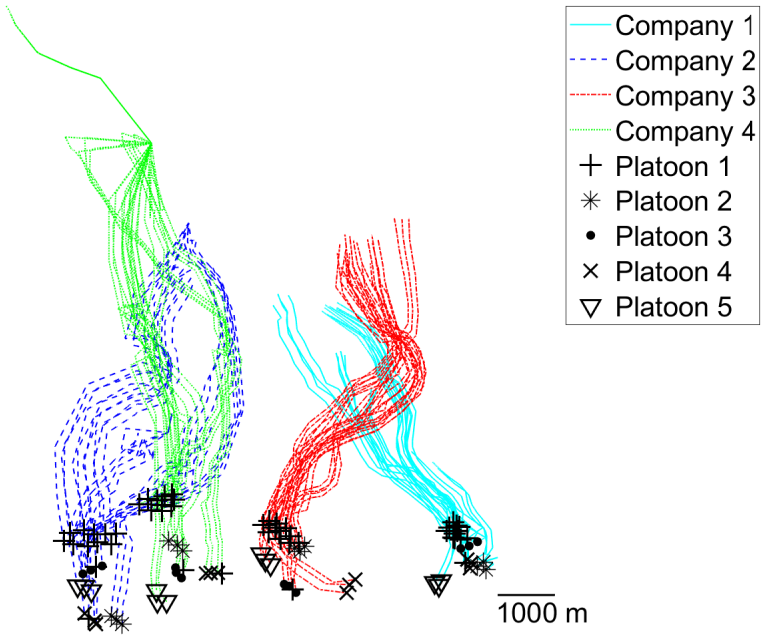

Figure 6.1: Platoons' movement patterns during vignette 2 emulation

between the nodes is discontinued due to the increasing distance between them. To coordinate with one another the emulated troops rely on VHF radio communications. The final vignette describes war-events in the network with: neutralization of insurgents, medical operation and even attacks of the enemy position. Also in this vignette the main communication enablers are HF and VHF radios and the support of UAVs that act as network rely nodes.

So far the NATO IST-161 RTG on Efficient group and information centric communications in mobile military heterogeneous networks exploited this scenario for evaluation of communication protocols. In particular, the RTG rely on a subset of vignette 2 consisting of the nodes forming two tank companies (Company 1 and Company 2) and two mechanized infantry companies (Company 3 and Company 4), with movements from scenario time 5000 seconds to 7000 seconds, depicted in details in Fig.6.1. A schematic representation of the network environment is shown in Fig.6.2.

Each company had its own wideband network on a separate frequency, and a wideband overlay network was used to connect the company networks. Two nodes per company (respectively, nodes with ID 1, 2, 25, 26, 49, 50, 73, and 74) also participate in an overlay network, acting as gateways, to support inter-company communications. Each gateway essentially has two network interfaces – the first one to communicate with other nodes in the local company, and the second one to communicate with the other gateways that were also part of the overlay network.

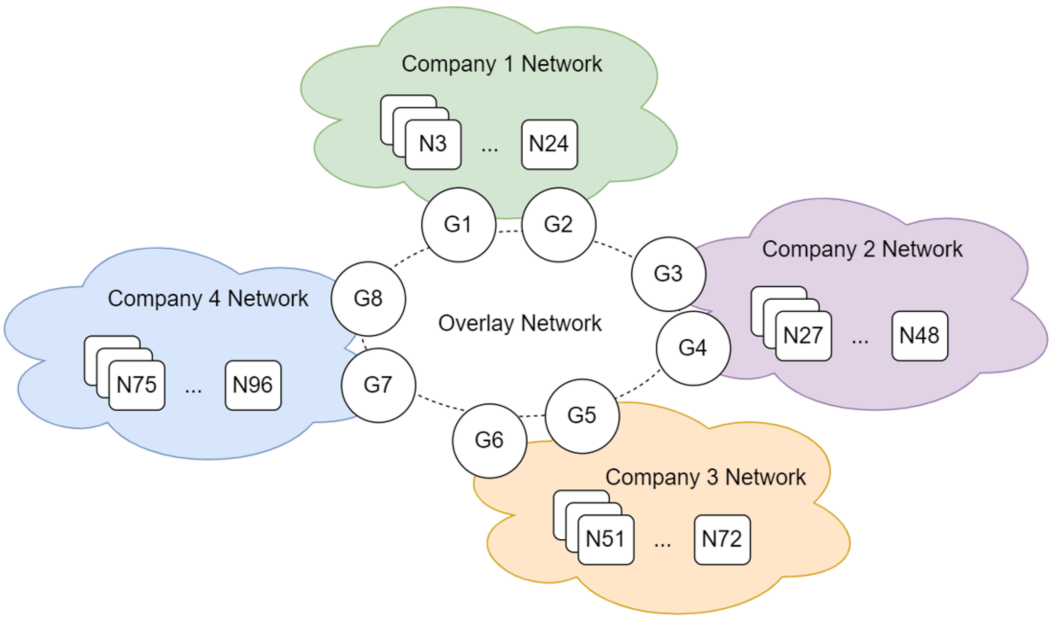

Figure 6.2: Schematic Representation of Vignette 2 Network Environment for Communication Protocols Experimental evaluation

6.1.1 Vignette 2 Details

Radio Waveform

The Synchronized Cooperative Broadcast (SCB) waveform is designed to primarily transport broadcast traffic within multi-hop mobile networks (and hence is well-suited to group communications). In particular, the SBC topology is precomputed and mimics a static SCB network and allows us to fully model SCB's cooperative effects. With this configuration the sum of the calculated received power from the simultaneous transmissions was compared to a system pathloss threshold, $L_{b,max}$, to determine the nodes reachable (via multi-hop) from each source node. The Estimated packet latency based on the number of hops between the source and the destination nodes and set the SCB network data rate to $R_L/(N*D)$ where R_L is the link data rate, N is the network size, and D is the SCB slot size, which is typically set to 4 for the RTG investigations.

Moreover, SCB is typically configured with $L_{b,max}$ set to 139 dB, bandwidth 1.25 MHz, and link data rate 875 kbit/s. The overlay network consisted of four nodes per company, with two of the four nodes acting as gateways. From the pre-computed SCB topologies, the connectivity of the networks can be calculated as the fraction of node pairs that are connected via one or more hops in the SCB waveform, and it is shown in Fig. 6.3 for the four-company overlay network.

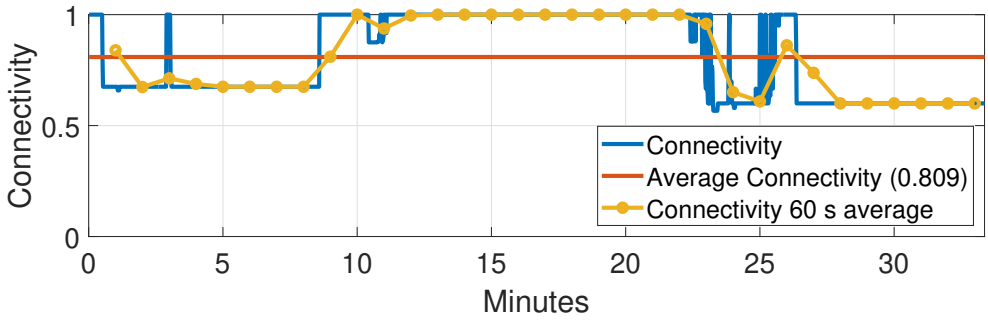

Figure 6.3: Connectivity of the overlay network.

Inter- and intra-company communications

Within each company network, the bandwidth available has to be divided among the 24 nodes in the network. With a total bandwidth of 875 kbits/s and the formula previously shown, and assuming a static TDMA schedule with equal bandwidth allocation, that results in an average bandwidth per node of 9.1 kbits/s. However, given that the first two nodes of each company are also part of the overlay network and have to relay data between the local company and the rest of the companies and given that node 6 of each company also sends sensor data, the TDMA is adjusted to give more slots to nodes 1, 2 and 6. In particular, the nodes 1 and 2 can transmit at a rate of 29 kbits/s, node 6 28 kbits/s, and each of the other nodes in the company 6.2 kbits/s.

For the overlay network, the total bandwidth was divided between the number of nodes that participate in the overlay. With four companies, each of the eight nodes were allocated 27 kbits/s.

Jamming models

The anglova scenario also offers diverse configurations to model attackers' disturbs to tactical information sharing. By default the company networks does not present any jamming attacks. In that case the company networks were connected and had a connectivity of one. However, diverse special cases can be considered. For example, as a rough model of a jammer quite far away with free line of sight to the nodes in the company networks, it is possible to set $L_{b,max}$ with a value of 10, 20 and 30 dB. This resulted in more and more fragmented company networks, as shown in Fig. 6.4.

6.1.2 The Test Harness

To evaluate protocols within the subsection of Vignette 2 described a specific common test harness is deployed within each node. In particular, the test harness implements generic required features such as message generation, logging, configurable

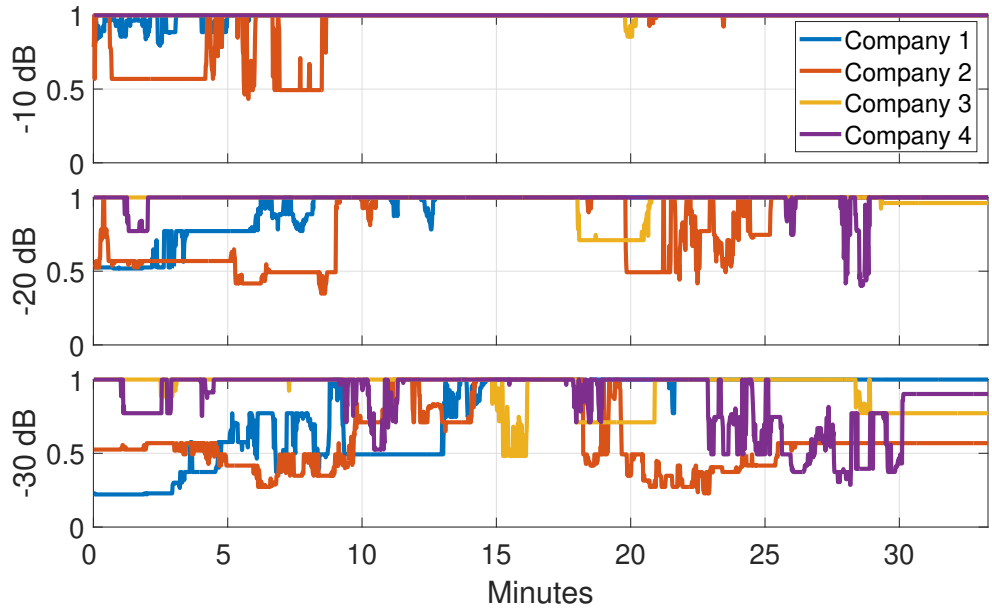

Figure 6.4: Connectivity of the company networks, with $L_{b,max}$ reduced by 10, 20 and 30 dB.

QoS requirements and allows to specify the content of interests of each particular node. For example a node can be configured to receive all message generated from a specific node or all contents related to a certain topic. Meanwhile to support different communication protocols, the test harness provides a plug-in based system so that new protocols can be added and selected for the experiment. Moreover, the test harness also implements the necessary tools for the data analysis. In particular, it register the number of messages successfully delivered to each node, the delivery latency, and the overall bandwidth consumed throughout an emulation.

To also better evaluate protocols, the test harness implements 3 different type of messages with different requirements: Blue Force Tracks (BFT), Sensor Datas (SD), and Headquarter Documents or reports (HQs). BFTs are small size messages, from 128 to 512 Bytes, and disseminated every few seconds (usually 5 to 10). A BFT messages represent a sort of position update messages or hello messages disseminated with best effort policies and all nodes produce and should receive these type of messages. SDs are medium size messages, from 128KB to 256KB, that represent messages produced and sent by sensors scattered within the whole network. Contrary to BFTs, SDs are produced by a node within each company and only few nodes should receive them. Moreover, SDs should be disseminated reliably. Similarly to SDs, HQs share sent and received by few nodes and should be re-transmitted in case of packet loss. HQs represent complex analysis that should be sent to the headquarter in order to provide detailed description of the condition of the mission, therefore, the size of these messages are between 512KB and 1MB.

6.2 Evaluation of Pub/Sub in the Anglova Scenario

Within the Anglova scenario these thesis presents the experimental evaluation of 4 pub/sub protocols. More specifically, 2 communication protocols specifically designed for tactical environments, namely Generic Data Exchange Mechanism (GDEM) and Dissemination Service (DisService), and 2 COTS communication protocols typically used in enterprise IT environments but also sometimes adopted at the tactical edge, such as NACK-Oriented Reliable Multicast (NORM) and NATS. The objective of this comparison is to evaluate whether the aggressive solution to recover lost messages that GDEM and DisService implement actually bring some benefits from the jamming resilience perspective. However, while the experiments make use of the jamming they will only evaluate the dissemination of BFTs and SDs

6.2.1 GDEM

GDEM is a, currently, proprietary group communication protocol developed by TNO [150]. It is based upon the Joint Dismounted Soldier System Information Exchange Mechanism (JDSS-IEM) [151] that is specified in STANAG-4677 [152]. In the experiments 2 configurations of GDEM were considered. Firstly, a reliable configuration (GDEM in the results) where GDEM will attempt to repair missing messages of both message types. Secondly, an unreliable configuration (GDEM-U in the results) where GDEM will only attempt to repair missing sensor data messages and the blue force message type is sent unreliable.

Compared to [150] TNO introduced the following changes to GDEM: 1) GDEM is now able to repair partially received messages whereas JDSS-IEM needs to resend the full message when a single segment of that message is not received; 2) GDEM is now able to segment and reassembles messages, it does no longer rely on a separate transport layer for message segmentation; 3) Each message segment has a full GDEM header; 4) Next to 16 byte UUID's, GDEM is now also able to use a 4 byte gateway identifier, 4 byte identifiers were used for the experiments in this paper; 5) a newly designed message relaying mechanism is used.

The GDEM protocol now supports networking compartmentalisation through the use of its relay function. A relaying gateway that implements the relay function can be part of multiple networks. Relaying rules describe how messages should be relayed between these networks. There is no theoretical limit to the amount of networks a relaying gateway is part of. When a message is relayed a hop counter, that is part of the message envelope, will be updated. A GDEM will also add its own

address as the last hop address to the message header. This allows other gateways to detect whether a message was relayed or not. For each network a relaying gateway will learn what other gateways are active on that network. As messages can find their way to the relaying gateways via multiple networks the relaying gateway will consider the network(s) with the lowest message hop counter as the home network for a gateway. A relaying gateway will only relay messages to a network when that network is not the home network of the sender of the message. When a message has a specific target address the message will only be relayed to the home network that belongs to the target address. As multiple relaying gateways may be active in the same networks messages are only relayed to networks where the message was not seen before. To accommodate this a relaying gateway will wait for a random delay interval to listen for a duplicate message before it decides whether or not to relay the message. If in that time frame an identical message is received that message will be dropped as that message was already relayed to that network by another relaying gateway. To optimise message synchronisation between nodes in different networks the relaying gateway is able to answer synchronisation requests on behalf of other nodes. When, according to the relaying rules, a synchronisation request message may be relayed by a relaying gateway that relaying gateway may answer that synchronisation request if it has the requested data in its message cache.

6.2.2 DisService

Dissemination Service (DisService) is a peer-to-peer (P2P) communication middleware that provides the applications with group-based, publish-subscribe information dissemination capabilities built on top of UDP multicast [**?**] [153]. DisService is designed to be disruption-tolerant and promote information survivability in the network by leveraging "opportunistic listening" and implementing an aggressive data caching policy [154]. This increases the probability that a subscribing node is able to retrieve missing messages (or fragments) from its peers, even in presence of temporary link disconnections and network partitioning that render the sender unreachable.

DisService allows application to create and participate in multiple groups and select how they want messages to be sent, e.g., reliably or best effort, and retrieved, e.g., in publication order or as soon as a message is received. DisService implements reliable multicast communications through a NACK-based system, whereby the subscribing node periodically notifies its peers of any missing message (or fragment) until the whole message is finally retrieved. In the experiments, both reliable and unreliable configurations were considered (respectively DS and DS-U in the plots) for DisService.

DisService implements several features that can reduce network bandwidth consumption, information access latency, and improve scalability, which are particularly interesting when dealing with limited network resources. Chunking permits to divide the data that represent some information, such as an image or a video, into a number of smaller, lower-resolution, but singularly intelligible parts that peers can retrieve independently upon demand and only if needed. Finally, DisService provides advanced features such as the on demand dissemination of metadata and the distributed querying of messages cached by the nodes in the network.

6.2.3 NORM

NORM is an IT enterprise protocol often proposed for tactical environments [155] [156] [157] [150]. In the experiments, the NORM implementation of choice is the version that ZeroMQ (https://zeromq.org) leverages to realize PUB/SUB sockets.

To accommodate the network compartmentalisation that is used in the experiments, the message forwarding between the company and overlay network is based upon their source IP address. Because ZeroMQ SUB sockets do not provide a way to find the source address of an incoming message, the message sender adds the 4 bytes of its IP address to each message it sends, as an additional header. To prevent messages from looping around a method that drops duplicate messages when they are received within a preset threshold is introduced. Note that this dropping of messages occurs at the application level and dropping messages that have an identical payload may not have the intended behavior. An application that uses our implementation must provide its own way of differentiating between messages that share the same payload. In the case of the experiments conducted for this paper, the group communication testbed is implemented in such a way that all messages that are sent are unique. The ZeroMQ socket does not support message fragmentation but once multiple fragments of a message are offered ZeroMQ takes care of delivering the fragments in order. Downside of this approach is that no relaying of (parts of) messages is done before the message was fully received on the relaying node. An implementation that builds message fragmentation and reassembly on top of ZeroMQ may have been more efficient for this use-case.

6.2.4 NATS

NATS is an Open Source (Apache 2.0) broker based publish and subscribe messaging system developed by SYNADIA [158]. Binary messages are published into channels called subjects; by default, NATS implements a fire and forget policy meaning that all the messages published in a specific subject that predate a node subscription to that subject are lost for that subscriber.

Delivery guarantees for messages generated after subscription are based on the characteristics of TCP while the connection between brokers and subscribers is monitored through a simple PING/PONG protocol implemented with the objective of breaking inactive connections before TCP timeouts.

NATS supports compartmentation through broker federation. NATS brokers can in fact be connected to each other in a mesh. Once connected, the brokers exchange topology information and are able to forward published messages for up to 3 hops.

6.3 Pub/Sub Experimental Results

The first analysis conducted on the capabilities of protocols to actually deliver messages in a denied environment uses as a reference metric *delivery ratio, i.e., the percentage of destination nodes that received a message within the pool of intended recipients*. Fig. 6.5 and 6.6 show, respectively, the histograms of the delivery ratios of BFT and sensor data messages for the 6 group communication solutions under 4 different conditions: no jamming (i.e., 0 dB), 10, 20, and 30 dB jamming. The x axis of the figure indicates the delivery ratio of a message (which of course is limited between 0 and 1) and the y axis indicates the number of messages that exhibited the corresponding delivery ratio. This plot format allows to grasp visually in a relatively straightforward fashion the performance of a protocol with respect to its capability to deliver messages. In fact, the ideal performance would be a single column at the right of the plot, corresponding to a 1.0 delivery ratio. The more a histogram differs from this ideal inverted-L shape, the worse its performance. Even from a quick glance to the figures it is possible to identify the impact of jamming on the procotols' performance, with higher levels of jamming leading to histogram shapes that exhibit a higher density of low delivery ratios and look less and less like the ideal inverted-L.

As one can see, a higher jamming value has an impact on the performance of all protocols - albeit to very different degrees. NATS exhibits poor performance across the board. Although it performs much better than NATS, NORM suffers considerably from jamming. Finally, jamming impacts the performance of DisService and GDEM to a much lesser extent, with reliable versions of those protocols performing decently well even in cases of severe jamming.

Table 6.1 and 6.2 enable a less immediate but more in depth analysis of the protocols' performance. Table 6.1 displays the delivery ratio recorded for BFT messages after 5 seconds from their initial transmission (column @5s), after 10 seconds (column @10s), and overall (column OA), with 0, 10, 20, and 30 dB jamming levels. The results indicate that DisService outperforms its competitors, either in reliable or unreliable mode, with GDEM being a close second both in terms of overall and

short-term delivery ratio. However, this discrepancy might be caused by the new GDEM's message relay implementation which is not working optimal yet.

Table 6.2 shows the delivery ratio (DR) and throughput (in Bps) used at the company level network (BCN) and at the overlay network (BON) recorded for sensor data messages with 0, 10, 20, and 30 dB jamming levels. The data indicate that GDEM has the lowest footprint from the bandwidth consumption perspective, DisService comes in second place, exhibiting a relatively aggressive tendency in trading off bandwidth consumption (for retransmissions) to achieve the highest possible message delivery ratio. Both NATS and NORM shows very poor performance due to their poor handling of large messages such as those used to carry sensor data.

Finally, the *delivery latency, i.e., the time elapsed between a packet*[1] *transmission and its receival* is analyzed. To investigate the dynamic behavior of protocols and see how they reacted to changes in the emulation environment, the dataset is sliced in different time windows, each corresponding to 5 minutes of time, and separately analyzed the protocol performance in those windows.

Using box and whiskers plots, Fig. 6.7 and 6.8 show the distribution of delivery latencies obtained for BFTs and sensor data for the 6 group communication solutions under 4 different conditions: no jamming (i.e., 0 dB), 10, 20, and 30 dB jamming. Of course lower values are better as they correspond to quicker delivery times. Note that the better performance that protocols seem to demonstrate as the jamming level intensifies needs to be offset by the fact that the protocols are indeed delivering a significantly lower amount of messages to their destination. So, basically protocols exhibit a bipolar behavior: either deliver messages quickly or they don't deliver them at all. As anticipated by the data in Table 6.1, GDEM has very good performance. DisService, its closest competitor, is also quite good but exhibits higher latencies, especially in reliable mode, perhaps because it aggressively leverages store-and-forward communication semantics to deliver messages.

Overall, both DisService and GDEM seem to be solid performers across the board, with the former exhibiting a relatively lower responsiveness - perhaps due to its more aggressive stance in trading off latency for a higher delivery ratio. NORM is a decent performer for BFT but is incapable or dealing with larger sensor data messages and NATS is basically out of its element in this scenario.

[1]Note that, since a message has many recipients, each message will generate many packets.

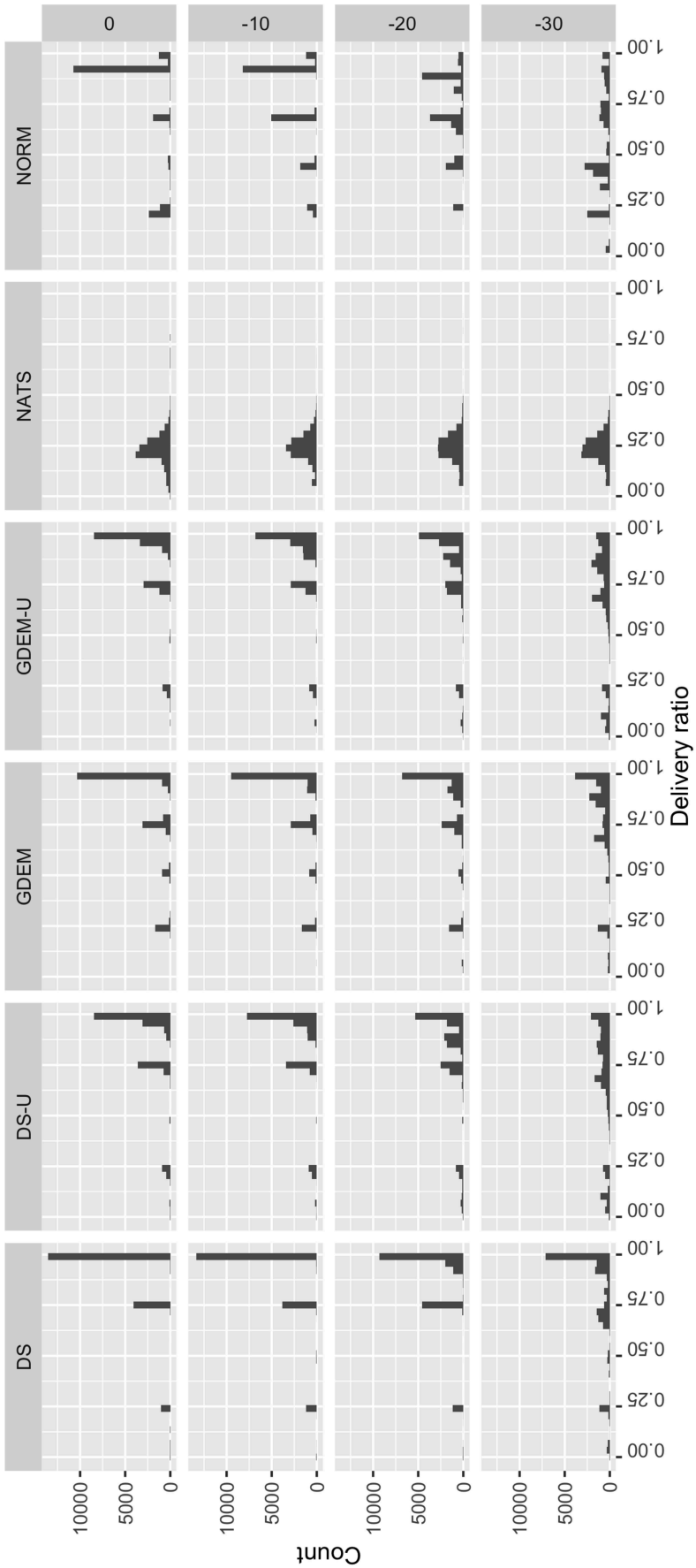

Figure 6.5: Delivery ratio distribution for BFT messages with 0, 10, 20, and 30 dB jamming levels.

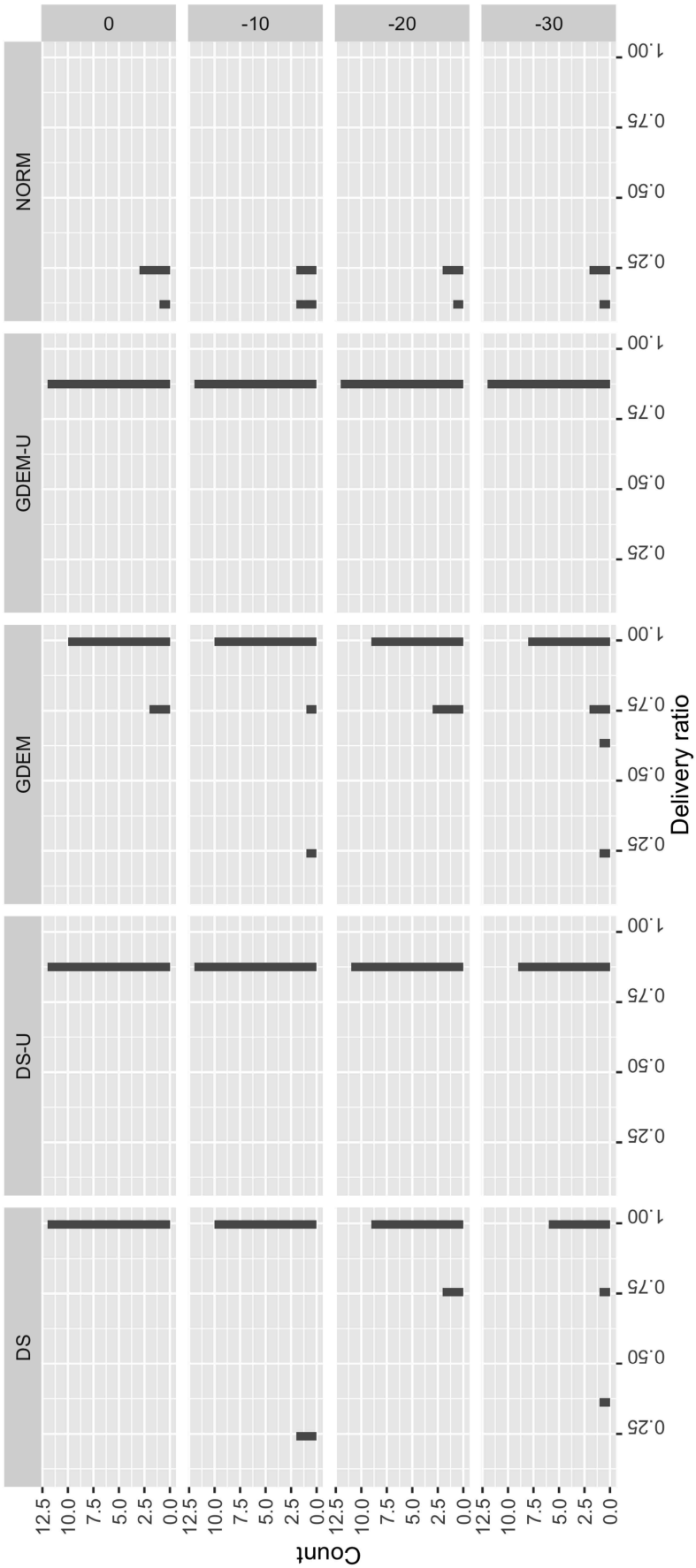

Figure 6.6: Delivery ratio distribution for sensor data with 0, 10, 20, and 30 dB jamming levels.

Table 6.1: Delivery ratio recorded for BFT messages after 5 seconds from their initial transmission (@5s), after 10 seconds (@10s), and overall (OA), with 0, 10, 20, and 30 dB jamming levels.

Protocol	Jamming Level											
	0 dB			10 dB			20 dB			30 dB		
	@5s	@10s	OA	@5s	@10s	OA	@5s	@10s	OA	@5s	@10s	OA
DS	36,15	40,31	88,31	25,9	27,77	87,85	20,41	22,04	84,14	22,59	24,86	77,47
DS-U	43,92	46,96	84,43	33,86	36,49	83,49	30,22	32,13	78,7	36,91	39	66,2
GDEM	34,5	34,95	81,65	34,77	35,31	81,68	33,18	33,78	78,63	32,11	33,4	74,68
GDEM-U	65,52	66,56	85,84	66,01	67,11	84,35	64,08	66,4	79,98	55,48	57,32	67,24
NATS	4,1	5,82	17,89	4,2	5,83	17,94	4,2	5,81	17,78	4,01	5,69	17,49
NORM	25,62	30,2	75,66	23,76	27,07	75,23	20,02	22,2	64,6	16,35	17,71	49,9

Table 6.2: Delivery ratio (DR) and bandwidth (in Bps) used at the company level network (BCN) and at the overlay network (BON) recorded for sensor data messages with 0, 10, 20, and 30 dB jamming levels.

Protocol	Jamming Level											
	0 dB			10 dB			20 dB			30 dB		
	DR	BCN	BON	DR	BCN	BON	DR	BCN	BON	DR	BCN	BON
DS	100	201294	47029	93,75	233501	50868	78,12	271490	53365	59,38	296559	47976
DS-U	87,5	169594	47290	87,5	162507	44412	79,7	165612	44585	65,62	141728	40120
GDEM	94,2	104516	31291	89,58	104185	29908	87,5	116593	31454	87,5	136230	31500
GDEM-U	87,5	120397	26598	87,5	115891	25722	87,5	126825	27968	87,5	128875	27168
NATS	0	289497	85296	0	285352	82784	0	279064	83614	0	278285	81880
NORM	7,29	464799	95343	6,25	463313	92233	5,2	444529	86560	5,21	409464	75103

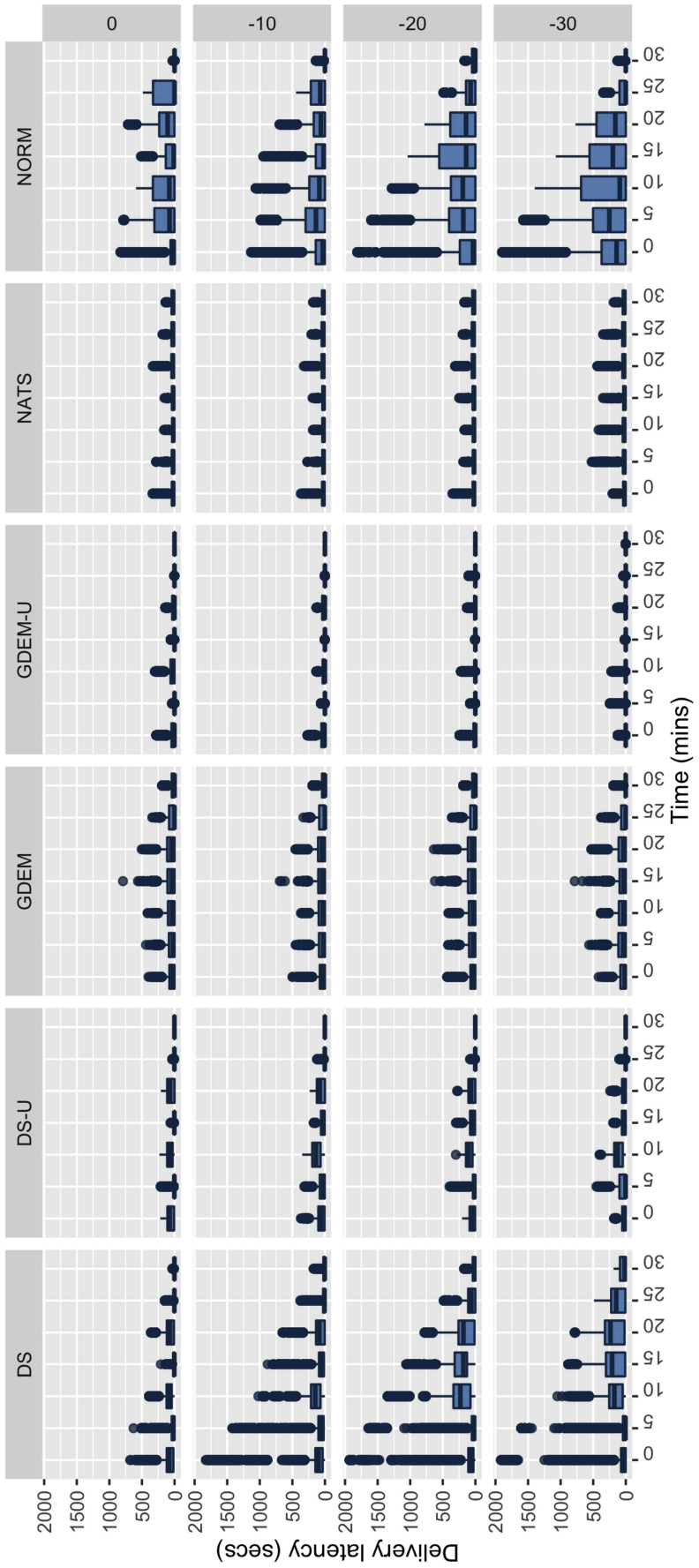

Figure 6.7: Delivery latency distribution of BFT messages aggregated within 5-minute time intervals, respectively reduced by 0, 10, 20, and 30 dB.

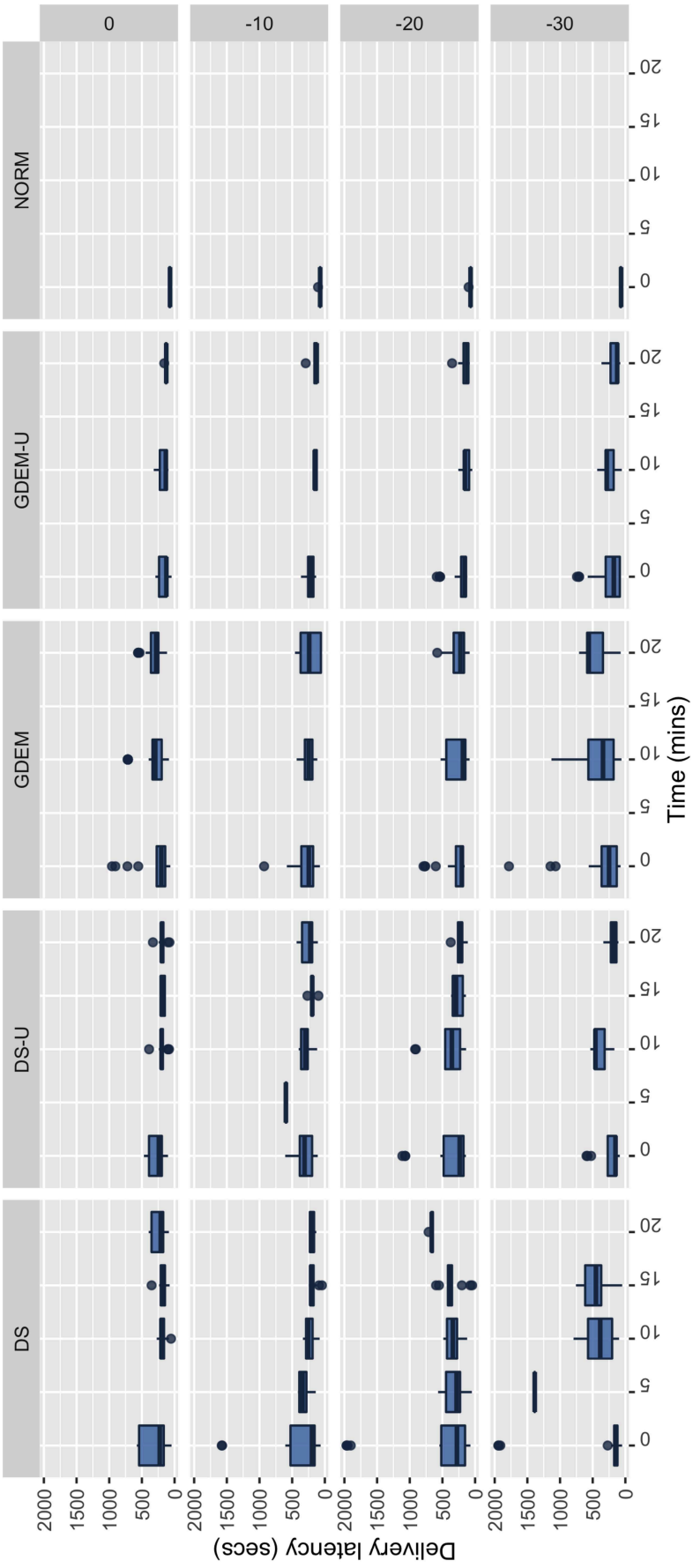

Figure 6.8: Delivery latency distribution of sensor data aggregated within 5-minute time intervals, respectively reduced by 0, 10, 20, and 30 dB.

Chapter 7

Named Data Networking

Another protocol that us evaluated in this thesis is NDN. In the classic Internet protocol stack, IP represents the "glue" in the network. IP is host centric, which means that IP addresses are used to locate the requested information. Information is therefore bound to the one unique IP address of the server where it is produced or stored.

7.1 NDN

NDN is a clean slate architecture that does away with the host centric architecture of the classical Internet [159]. In NDN the focus is on finding the information (content) that a client wants to retrieve irrespective of where it is stored. This is done by addressing the information by name rather than by its source (host name or IP address).

In NDN, the content naming scheme is a fundamentally important and application specific design choice. This means that the first step in NDN application development is defining a naming scheme that fits the content characteristics and the application's particular needs. For instance, in a tactical application, the content name can be built hierarchically and be human readable. An example would be:

```
[Mission_network_xx/Intelligence_reports/
Geographic_area_x,y/role_xx/today]
[Mission_network_xx/weather_sensor/
Geographic_area_x,y/windspeed/current]
```

While this forces software engineers to address information production and consumption related aspects early on in the development process, it also affords considerable liberty to explore a wide range of naming schemes, from simple hierarchical to tag-

and/or keyword-based ones [160], to find the best suited one for the particular application. Of course, the names must be commonly agreed upon by all the consumers and producers in the information domain where they operate (e.g. mission network, national network, etc.).

NDN is built on two simple basic primitives; request for a specific content and the response with the matching data. In NDN the two packets that perform these primitives are called *Interest* and *Data*. In order for any content to flow in the network, the consumer must issue an *Interest* that specifies the name of the content that the consumer in looking for.

When the consumer has issued the Interest for the required content, the Interest is forwarded through the network in search for a node that holds the content. When the content is found, it is wrapped in a Data packet, which follows the reverse path of the Interest packet back to the consumer.

The Interest and Data management primitives are implemented in a forwarding engine that is installed in all network nodes (routers, clients and servers) in the NDN architecture, as shown in Fig. 7.1. In NDN, an interface over which content is transmitted or received is called a Face, which can be an internal interface towards higher layers (the application), a network interface, or other types of connections, like a TCP connection in case of hybrid NDN/IP solutions.

When an Interest is generated by the application, it reaches the forwarding engine of the node over an internal interface (Face 2 in Fig. 7.1). First, the forwarding engine checks if the requested content is available in the Content Store, an internal cache that stores copies of the recent Data packets received or forwarded. If the content is not available in the Content Store, the forwarding engine registers the Interest, as well as the originating Face, in an internal table called Pending Interest Table (PIT). If the Interest was already registered in the PIT, the engine simply adds the new Face in order to forward back the Data message to all interested consumers.

In case the Interest is not already in the PIT, the forwarding engine checks its Forwarding Information Base (FIB) to see which Faces to forward the Interest on in order to start looking for the content in the network. The FIB is similar to the routing table in IP architectures. The Interest is forwarded on one or several of the Face(s) that the FIB points at. This procedure is repeated in all forwarding nodes until the Interest arrives at a node where the FIB points at the Face to the application that produces the information or there is a match in the Content Store.

When a node is able to fulfill the request, meaning it either has content in its Content Store or is the node that produced the information, it resolves the Interest by sending back a Data packet. During this phase there is no need of FIB since the Data packet simply follows the bread crumb trail from the path taken by the Interest. In fact, each node in the path has stored the Face(s) that received the Interest so

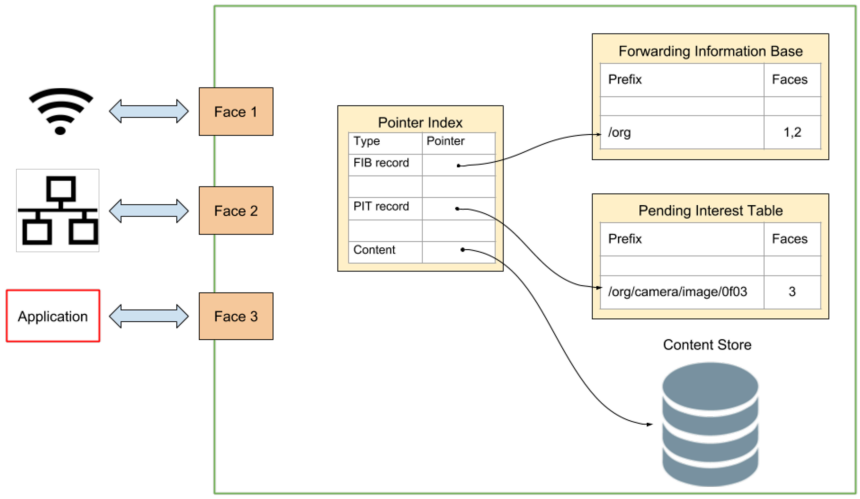

Figure 7.1: Overview of the forwarding engine in a NDN node.

once they receive the Data packet they update the PIT, and since the Interest is resolved, they cache the content (to increase data availability and performance when the same content is requested in the future), and forward back the message through the Face(s) that received the Interest packet. This simple procedure is repeated for all content the consumer wants.

Forwarding in NDN is stateful, which enables NDN to perform much more intelligent forwarding decisions than IP (which is stateless). This functionality, managed by the Strategy layer, can be used to enable access control (only allow Data with certain names to be forwarded), intelligent caching (cache and forward data based on its priority, which can be name-driven), as well as robustness to disruption (cache data based on known delay/disruption on the incoming link).

7.2 NDN Advantages

With the basic principles of NDN established, this section discusses some aspects of NDN that makes the architecture appealing for heterogeneous military networks, particularly with mobile nodes.

7.2.1 Disruption Tolerance

Several solutions, adopting both COTS and purposely developed approaches [19], have been proposed to address disrupted communications in tactical networks. Unlike native IP, the store and forward functionality of NDN enables it to withstand disruptions without the need for any specific extensions.

In fact, NDN adopts a communication model that decouples information producers and consumers and is not dependent on a stable connection between them. If at some point in time an Interest message can get to the producer or a Content Store that holds the content, the responding Data packet can start its way back to the consumer and will be stored at each hop. If the Data message is lost due to broken links, the consumer can simply reissue the Interest after a timeout.

This approach provides a simple mechanism for communication reliability. In fact, after the first Interest, each successive Interest sent has a higher likelihood of succeeding since the content is now likely stored in a Content Store closer to the consumer.

7.2.2 Node Mobility

Node mobility is a significant challenge for IP due to the dual nature of IP addressed being identifiers and locators. Supporting mobility requires purposefully developed solutions to enable connections to withstand node mobility or to support multipath communications [161]. Instead, NDN copes significantly better with mobility.

Node mobility can be split in consumer mobility and producer mobility. Consumer mobility is handled automatically by NDN. A consumer that has moved simply asks for new content in the standard way by issuing an Interest. The Interest builds a path from the new position of the consumer to the first node that holds a copy of the content and the Data packet follows the reverse path back to the new node position. If a node moves after it has issued an Interest and before it has received Data, then the application can resend the Interest after a timeout. In this way the new Interest will research the content required from the new position established and thus generating the new path necessary to the Data packet to reach the consumer.

Producer mobility is more problematic. In fact, when a Mobile Producer (MP) moves to a new position, a consumer might be unable to fulfill its Interest since the path to the MP might become invalid. To address this problem, the MP might hasten the creation of a new Interest path by generating a "breadcrumb trail" to a rendezvous node to enable Interests to trace content at its the new position. The rendezvous node can also be used as a deposit for content, allowing it to resolve consumers' Interests while the routing strategies have not aligned to the new position of the MP. Furthermore, consumers and router nodes can also adapt their behaviour for this particular situation by choosing more aggressive Interest flooding strategies in order to find the new location of the producer.

7.2.3 Multicasting and Multihoming

The NDN architecture seamlessly handles the dissemination of content to a single consumer or to a group of consumers. NDN manages all the interfaces over which data are sent or received with components named Faces. Each Face can be connected to higher layer entities, such an application, a physical network interface or even a virtual link, such as a TCP connection in hybrid NDN/IP architectures. As a result, NDN supports multicast communications out of the box.

Furthermore, NDN overcomes the well-known problems that IP architectures presents with multihoming. In fact, since Interest routing is based on the content name, NDN works out of the box in heterogeneous network environments with multiple channel technologies and is capable of aggregating Interests received from different faces, so that only one Interest per content is sent over a shared communication link. Furthermore, nodes with multiple networks interface are also profitable in term of caching. When a Data packet is sent back to the consumer, NDN leaves a copy of the message in the local Content Store of all nodes along the reverse Interest path. Popular content is then automatically made available close to its consumers.

7.2.4 Synergies with IoT

NDN presents interesting synergies with IoTapplications [162]. In fact, while NDN was designed with *"one Interest, one Data"* semantics, other ICN implementations (most notably PURSUIT) were specifically designed to support publish/subscribe communications. As a result, borrowing on concepts and tools from the aforementioned ICN implementations, and through the clever adoption of specifically designed naming schemes [160], NDN could be extended to realize publish/subscribe communications with support to topics and distributed caching - a communication model that is particularly well suited for IoT.

More specifically, an IoTapplication based on NDN could adopt naming schemes that, for example, organize devices and the information they generate in a strict hierarchy or adopt less restrictive taxonomies that can generalize multiple devices. This would allow directly using naming schemes to as part of the basic discovery mechanisms by investigating the header of the Data requested or even information cached in the local content store without deploying dedicated discovery protocols.

In the tactical environment, this means that NDN could become an interesting enabling technology for IoBT [15]. More specifically, the NDN capability to leverage locally available information is likely to become a sought-after feature for military operations in the future where information is expected to be scattered all over the network and the challenge lies in finding the right information.

7.2.5 Interface between application and network layer

TCP/IP applications leverage a range of solutions including unicast, multicast, and DTN, to implement realize a wide range of communications patterns. However, those solutions are mostly designed to masquerade network behavior from the application – leading to poor performance in tactical environments unless specifically developed solutions proposing an enriched network programming model are adopted [19] [20].

Instead, NDN provides applications with a simple but powerful programming model that allows tuning application behaviour to the current network performance. More specifically, by properly tuning the Interests transmission rate, a node can attempt to mitigate node mobility issues and control the traffic flow according to the available network resources. This gives NDN a flexible flow control function that works in a distributed manner and can make local decisions.

In addition to application specific configuration, node level configurations also represent convenient "knobs" for performance tuning. For example, a router can for example set a limit on the number of pending Interest that it allows on an outgoing Face, or the number of Data messages that are forwarded back.

7.3 NDN Challenges

Clearly, the previous section has identified a number of advantages of NDN over traditional IP networks. Howerver, NDN raises a number of new issues that must be studied further to evaluate its utility in heterogeneous military networks. This section lists some of the important topics within NDN that need further examination.

7.3.1 Naming

Naming is one of the most challenging tasks in NDN application design. While it is certainly impossible to devise a standardized naming scheme that applies to a wide range of application domains for traditional Internet applications, this problem could at least be partially addressed for military networks. In fact, military applications consider a significantly smaller amount of content types than Internet ones. Furthermore, military information already present standardized formats that provide common information structures, which can be used to regroup contents and obtain content taxonomies. These properties will simplify the design of naming schemes, and also potentially increase its effectiveness since it is directly tailored to the information. Nevertheless, this topic requires much attention.

7.3.2 Security

The NDN security architecture [163] is very flexible but raises some challenges - particularly for mobile military networks. In NDN, information security (INFOSEC) is placed on the content. Each content chunk is signed and optionally encrypted. This is flexible but introduces overhead. The public key of the signer, a certificate for that public key or a pointer to them must be sent with the packet and a trust chain must be in place. For low data rate mobile connections this can be problematic. Traffic analysis is also a challenge, in NDN the Interest is sent in clear text. This is not adequate for military networks [164]. The security model does not handle network security (NETSEC). There must be functions in place that perform network security.

The security discussion in [165] and [166] shed some light on some of the security challenges. Some traditional solutions for mobile military networks such as preloaded keys and link level encryption can solve some of these problems but more research is needed.

7.3.3 Strategy

The strategy layer is a powerful component in the NDN architecture but must be studied and tuned to achieve its full potential. The flexibility introduced by this layer enables to tailor the forwarding policy of Interest and Data to the application specific requirements. In fact, NDN allows to implement a wide range of strategies, from naive ones that simply mimic standard IP communication mechanism (e.g. unicast, multicast) to more sophisticated ones that continuously analyze their performance and self-learn how to improve their performance.

For instance, FIB can be based on overheard traffic such that the Interests might sent on one or several Face(s) in the direction of where Interest for similar content has been resolved before. However, strategies that introduce high redundancy in the Interest distribution should be carefully adopted for networks where links capacity is limited or present different resource constraints. Strategies should be designed to achieve the best tradeoff between delay in fetching the information and network resource utilization. At this matter, proactive routing solutions that announce cached content might allow the strategy layer to improve the Interest forwarding toward nearest producers and thus reduce resource utilization.

Finding the best trade-off between complexity, efficiency and robustness is crucial to exploit the flexibility of the strategy layer to adapt it to the network properties and communication requirements. At this matter, the experience from the vast research on routing in MANETs can be reused and extend to NDN strategy for mobile networks. The survey in [166] gives a comprehensive overview of different

routing approaches for mobile NDNs. The policies of the strategy layer as well as the supporting routing functions are still an open research area.

7.3.4 Reliability

NDN does not provide an integrated solution for a reliable communication. As a result, unlike the TCP/IP model in which reliability functions are provided by transport layer protocols, in NDN the responsibility to recover a lost message relies on applications – thus complicating the task of software engineers.

However, NDN properties allow to manage reliability through the already available mechanisms used for the nodes interaction. For example, a mechanism to provide reliability can be performed by issuing a new Interest packed if the previous Interest for the same content was not resolved within a certain timeout period. In this way application can achieve reliable information sharing with a simple and lightweight mechanism. This mechanism allows application to have full control to the timeout period and thus adapt this to the network capacity, QoS needed and so on.

7.3.5 Performance Tuning

Since NDN is a relatively new technology, the performance tuning of NDN applications is an aspect that still needs to be thoroughly investigated.

For instance, cache replacement strategies for Content Stores and values for several timeouts (including Interest retransmission) are critical parameters for application performance and robustness (as they not only influence delay and jitter of the requested Data but availability as well). Cache replacement strategies of classic IP architectures are a mature research field and this knowledge can be reused here. But more work is needed to learn which strategies are best to use for which routing strategy and for different data types. Furthermore, applications might reissue the an Interest if no Data has been received within a certain timeout. This parameter must be properly tuned to fit traffic requirements and network properties. More experience that can result in guidelines for how to set the parameters, are needed. [167] is one example reference that discuss Cache replacement strategies for NDN.

Chunk size is another very relevant performance related research topic in NDN. How large data elements should the consumer be able to ask for in one Interest? The NDN architecture promotes tiny chunks, as small as single voice samples or video frames. The advantage of this is a very responsive network. The Interest can be routed a different way for each voice sample and thus be able to handle mobility and avoid network congestion (do flow control) etc. very quickly. This comes at the cost of a large overhead; this Interest packet and headers in the Data packet

(that includes a security certificate) for each tiny chunk of data. Larger chunk sizes such as whole documents or videos reduce the overhead but also reduce some of NDN architecture's qualities. In fact, with larger chunks it is more likely that the forwarding of Data will fail.

7.4 BFT Dissemination Using NDN

To evaluate the effectiveness of NDN in the context of tactical edge networks, the test harness presented in the previus section has been extended to support NDN-based communications. This allows to compare NDN with different data dissemination solutions designed for tactical environments.

To bridge the mismatch between the push-based nature of tactical applications and the pull-based semantics of NDN, the experiment only focuses BFT dissemination which arguably represents the most challenging component of a tactical application for NDN. As a result, within the test harness previously described a simple NDN-based BFT dissemination application is implemented based on the assumption of isochronous (and known) BFT generation times. However, only a section of the network will be considered. More specificially, BFT information is published by each node every 10 seconds, and consumed by all other nodes. The payload of the BFT messages is 128 Bytes, resulting in a setup with 24 publishers and 24 subscribers.

In order to address these differences the following naming scheme has been adopted:

```
/anglova/blue_force/<node_id>/<seq>
```

where *anglova* is the root, *blue force* is the topic, *node id* is the node identifier and *seq* is an incremental value that distinguishes each data item the node has published. This scheme allows to uniquely identify the data published by each node while allowing each consumer to request it without the need of a content discovery mechanism. For example if a consumer needs the 16^{th} content item published by node 4 it can simply send an interest specifying the following name:

```
/anglova/blue_force/node_4/16
```

with the assumption of isochronous generation of BFT information each consumer can issue interests at the same rate as data is produced and thus emulating a push communication model through proactive Interest dissemination.

Once addressed the incompatibilities between the test harness and the NDN communicating model had to properly configured to the network stacks to allow information sharing. NDN supports different mechanisms to transmit messages to

other nodes via the so called *NDN face system* that allows seamless handling of native communication that bypasses IP or message encapsulation in COTS transport protocols such as UDP. For the experiment NDN leverages on UDP Multicast for message transmission. In this way, applications will broadcast Interests to all reachable nodes in the network that, if they have cached the requested content, will broadcast back the Data message. While this approach could limit the possibility of interest aggregation or data caching, due to the synchronous dissemination of interest, it will allow to obtain a first coarse grained evaluation of NDN capabilities to be refined in the future wit more sophisticated schemes and configuration.

7.5 Experimental Evaluation of NDN

Simlar to the pub/sub evaluation the behavior of the naive NDN-based BFT application is analyzed according to 3 key performance indicators: delivery ratio, delivery latency, and bandwidth utilization. Moreover NDN's performance with that achieved by two other group communication solutions: DisService and NATS, which represent important and relevant baseline references for comparison. As shown in [149], DisService outperforms many other group communication solutions, in terms of bandwidth utilization, delivery latency, and delivery ratio, in tactical environment conditions. Although BFT message transmission typically leverages best-effort communications, in the experiment considers DisService in both unreliable and reliable message transmission configurations. those two configurations will be referred with the names: DisService and DisService Reliable respectively. Furthermore, NATS is included in the comparison since, even if not as effective as DisService, it proved to be best performing broker based group communication solutions in our previous experiments [149]. Fig. 7.2 depicts the total number of messages received per node. The figure clearly shows that one group of nodes receive a smaller amount of messages. This is caused by the events of the scenario, that frequently disconnect nodes 13, 14, and 15 from the other nodes. Almost all the group communication solutions examined are very sensitive to this disconnection and fail to deliver a significant amount of messages to these nodes. However, NDN seems less affected by this issue and delivers more messages to nodes 13, 14, and 15 compared to DisService and NATS.

Fig. 7.3 further illustrates the effectiveness of protocols in delivering BFTs by showing the delivery ratio of messages, i.e., the ratio of nodes that actually received the BFTs. To better illustrate the dynamics of the protocol behavior throughout the duration of the scenario, the figure divides messages in 20 categories, according to the respected minute of generation time. Each category of messages is depicted using a Boxplot that allows to graphically evaluate the distribution of delivery ratios in the

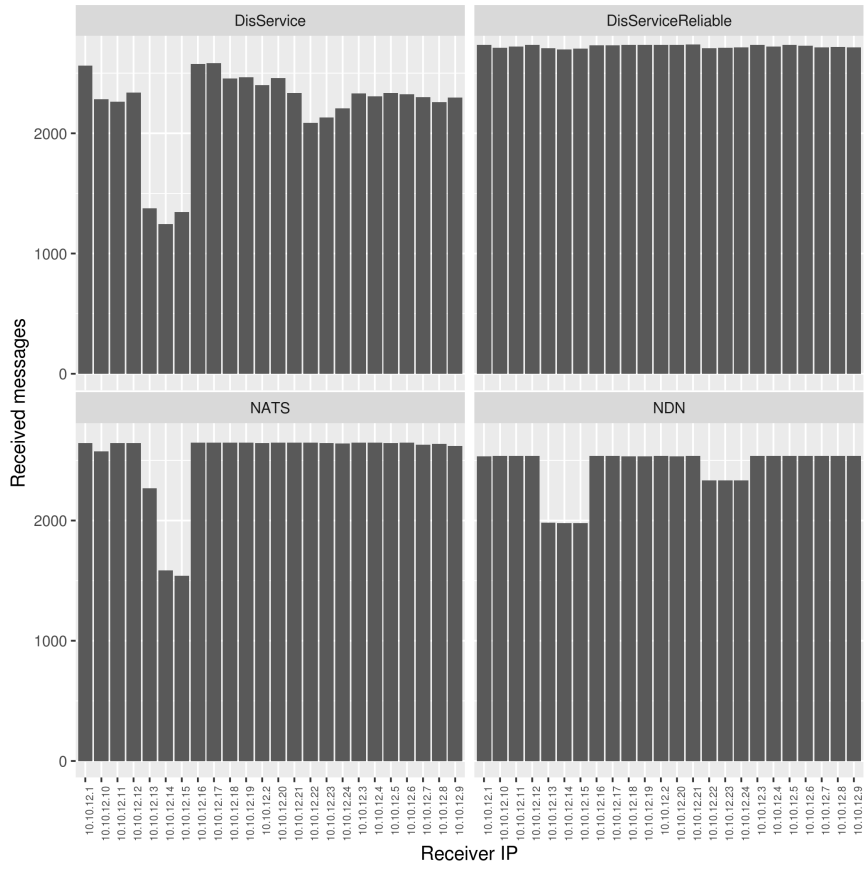

Figure 7.2: Number of messages received by each node.

messages of each category. The Boxplots allow us to clearly describe the sets of data gathered using their quartiles and thus highlighting dispersion and asymmetries.

Fig. 7.4 completes the effectiveness analysis of the group communication solutions from the application perspective by presenting the distribution of delivery latency. The figure uses the same format of Fig. 7.3, which classifies messages according to their relative minute-of-simulated-time of their generation and using Boxplots to present the distribution of delivery latency minute-by-minute. In this case, the graphs present different scales for the y axes since NATS and DisService Reliable presented outliers with high delivery latency values. However, even if DisService and NATS deliver some message after more than 10 seconds, the majority of BFTs is received quickly after their initial generation. NDN instead seems slightly slower in delivering messages, requiring on average 1 second for delivery. This might be caused by the proactive interest dissemination approach adopted to reenact push communication model. In fact, due to the NDN communication model, producers do not proactively push content to the consumers after it has been generated but instead wait to receive Interests from the consumers before transmitting data – with the result of increasing the overall delivery times.

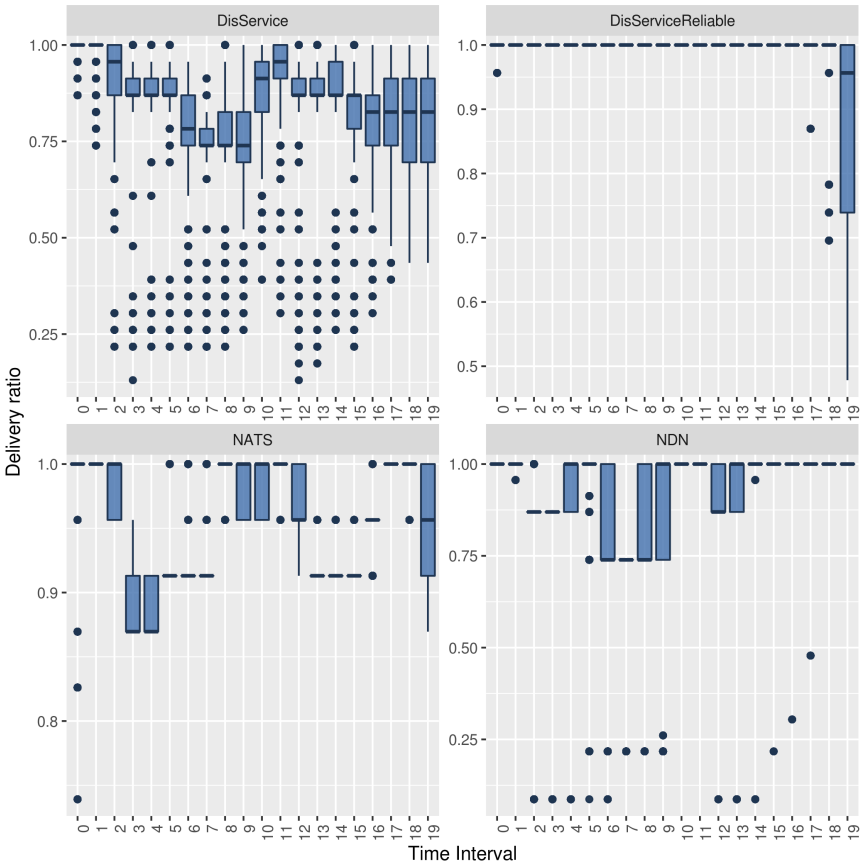

Figure 7.3: Delivery ratio per minute.

Finally, the total amount of outgoing traffic is presented in Fig. 7.5, as metric to evaluate and compare the network resource consumption of each protocol. This measure is obtained as the sum of the average amount of bits per second sent by each node. The figure shows that the naive message dissemination strategy adopted for the NDN-based application causes considerable pressure on the network. In fact, even when NDN relies on UDP Multicast and thus can leverage link-local multicast dissemination of data, it still presents a high overhead compared to DisService and DisService Reliable. This might be caused by the absence of tailored caching, Interest aggregation and forwarding strategy configurations that could theoretically reduce the bandwidth consumption of NDN while at the same time increase its effectiveness.

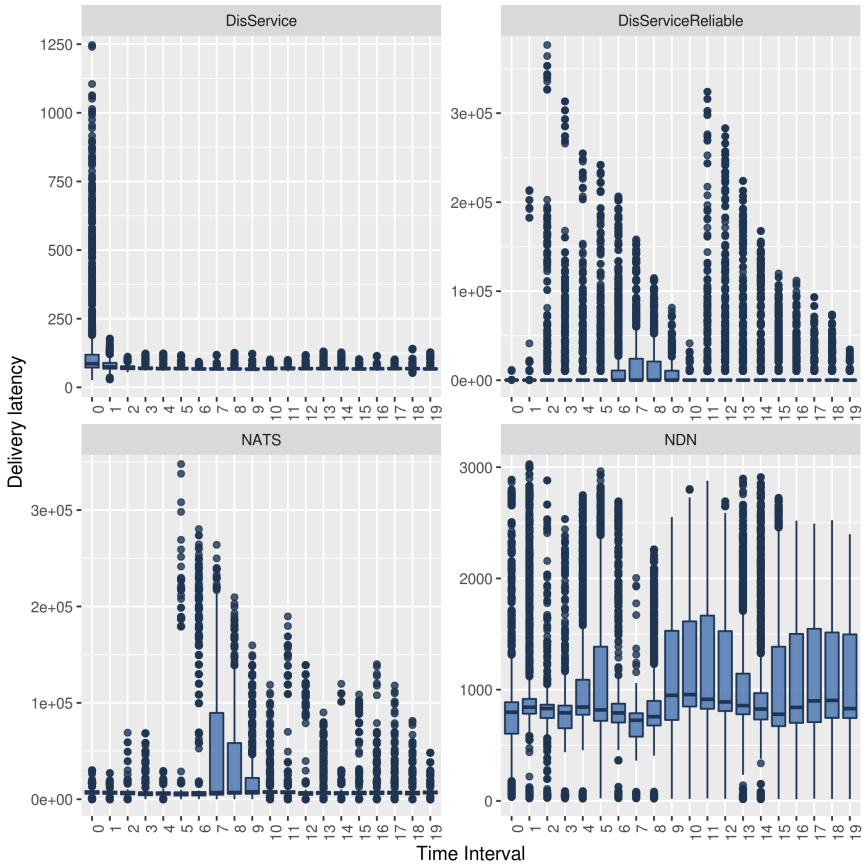

Figure 7.4: Delivery latency per minute.

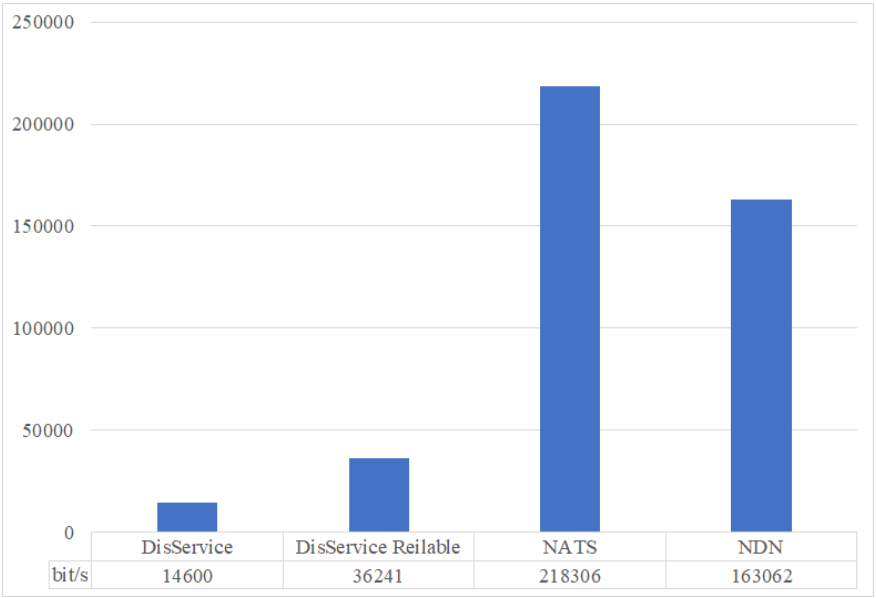

Figure 7.5: Cumulative bandwidth consumption per second.

Chapter 8

Conclusion

Enabling IoT-based applications in disrupted network environments is crucial to allow rescuers to rapidly acquire situational awareness and support critical activities. However, without the use of resilient solutions capable of rapidly restoring the accessibility to IoT assets, the IoT-based applications cannot fully take advantage of the sensing and computational devices that disastrous environments offer. Moreover, the network characteristics also require resilient communication protocols that allow reliable information sharing and exploit the scattered network resources to increase data availability.

To tackle the challenges of these environments, this thesis investigates specific methodologies that can enable the definition and adoption of IoT-based applications. In particular, this thesis proposes the adoption of proactive discovery mechanisms implemented by a distributed federation of gateways. Via this approach it is possible to define a rapidly adaptable and re-configurable middleware capable of surviving the dual nature of network fragmentation, due to both physical events and administrative administrative, that characterizes disrupted network environments.

Moreover, this thesis also investigates the effectiveness of the information-centric and ICN paradigms for information exchange in order to discuss the characteristics that communication protocols should offer in the targeted challenging environment. To this end, experimental evaluation of diverse communication protocols are presented and discussed.

To provide a more exhaustive and detailed summary of the research presented in this thesis, Chapter 2 presents the solutions and approaches that enable IoT in reliable network environments. Chapter 3 instead discusses renowned environments that widely adopt IoT devices, such as Smart Cities and modern warfare environments, and explains how natural disasters and the tactical networks' characteristics negatively impact the adoption of IoT.

In the Chapter 4, an in-detail discussion of the requirements and challenges to enable IoT in disrupted networks is presented. In particular, the chapter discusses

how mechanisms such as proactive discovery, federation and secure group communication can restore access to IoT devices.

Following, in Chapter 5, a proof-of-concept middleware named MARGOT is presented. MARGOT implements the mechanisms and the concept introduced in the previous chapter in order to both provide a possible architecture for a IoT management middleware for disrupted network environments and also enable experimental evaluation of the mechanisms. To this end, the chapter presents two experimental evaluations of the middleware in two scenarios: a single edge network domain and a multi-domain scenario. The first presents the effectiveness of the middleware in terms of discovery and context-filtering to simplify context-based strategies, while the second investigates the effectiveness of a federated approach and possible enhanced mechanisms for inter-domain information sharing.

Next, Chapter 6 presents an experimental evaluation of diverse communication protocols in a realistic emulated tactical environment. In particular, for the experiments this thesis makes use of the Anglova scenario to evaluate several communication protocols adopted in both enterprise and tactical environments. The results obtained for each protocol are then compared on the basis of three main indicators: bandwidth consumption, delivery ratio and delivery latency. Similarly, Chapter 7 presents an evaluation of NDN. More specifically, NDN is described, discussed within tactical network requirements and evaluated in comparison with other publish-subscribe protocols.

The concepts discussed in this thesis and the experimental evaluations present a new possible paradigm to restore IoT-access in disrupted network environments. In particular, proactive discovery achieved via a distributed middleware approach represents a novelty over the state-of-the-art of nowadays infrastructures, which highly rely on Cloud computing. Moreover, the study of ICN protocols for information sharing is still an open topic in the literature, especially within tactical environments.

Future works will continue the study of tactical networks and disaster recovery scenarios to study more comprehensive solutions. For example, network awareness can improve the definition of more exhaustive resource topology that improves context-based filtering. Moreover, the integration of the Value of Information paradigm to support inter-domain communication can further improve the federate middleware approach in order to efficiently exploit the scarce network resources. Finally, communication protocols would still require more studies related to also inspect the possible advent of new communication technologies. For example, 5G enabled M2M communication might further foster the development of fully-decentralized communication protocols.

In conclusion, this thesis summarizes the research pursued during my PhD's pro-

gram at the Department of Engineering of the University of Ferrara. During the program, I was part of the Distributed Systems Research Group under the supervision of my tutor Prof. Cesare Stefanelli and my co-tutor Prof. Mauro Tortonesi. Moreover, during my research I also cooperated with PhD Niranjan Suri, leader of the NOMADS group at IHMC, and the NATO IST 161 Research Task Group.

Bibliography

[1] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, pp. 49 – 69, 05 2011.

[2] H. Saha, A. Mandal, and A. Sinha, "Recent trends in the internet of things," 01 2017, pp. 1–4.

[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128610001568

[5] R. Morello, S. C. Mukhopadhyay, Z. Liu, D. Slomovitz, and S. R. Samantaray, "Advances on Sensing Technologies for Smart Cities and Power Grids: A Review," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7596–7610, Dec 2017.

[6] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, Oct 2017.

[7] T. Hui, R. Sherratt, and D. Díaz-Sánchez, "Major requirements for building smart homes in smart cities based on internet of things technologies," *Future Generation Computer Systems*, vol. 76, 11 2016.

[8] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mob. Netw. Appl.*, vol. 24, no. 3, p. 796–809, Jun. 2019.

[9] P. Barnaghi and A. Sheth, "On searching the internet of things: Requirements and challenges," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 71–75, Nov 2016.

[10] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[11] T. Khan, S. Ghosh, M. Iqbal, G. Ubakanma, and T. Dagiuklas, "Rescue: A resilient cloud based iot system for emergency and disaster recovery," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2018, pp. 1043–1047.

[12] N. Suri, Z. Zielinski, M. Tortonesi, C. Fuchs, M. Pradhan, K. Wrona, J. Furtak, D. B. Vasilache, M. Street, V. Pellegrini, G. Benincasa, A. Morelli, C. Stefanelli, E. Casini, and M. Dyk, "Exploiting smart city iot for disaster recovery operations," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 458–463.

[13] T. Tsubaki, R. Ishibashi, T. Kuwahara, and Y. Okazaki, "Effective disaster recovery for edge computing against large-scale natural disasters," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–2.

[14] S. Ferdousi, M. Tornatore, F. Dikbiyik, C. U. Martel, S. Xu, Y. Hirota, Y. Awaji, and B. Mukherjee, "Joint progressive network and datacenter recovery after large-scale disasters," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1501–1514, 2020.

[15] S. Russell and T. Abdelzaher, "The internet of battlefield things: The next generation of command, control, communications and intelligence (c3i) decision-making," in *IEEE MILCOM*, 2018, Conference Proceedings, pp. 737–742.

[16] M. Pradhan, "Federation based on mqtt for urban humanitarian assistance and disaster recovery operations," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 43–49, 2021.

[17] G. Riberto, M. Govoni, C. Stefanelli, N. Suri, and M. Tortonesi, "Leveraging civilian iot infrastructures to support warfighting activities in urban environments," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 118–123.

[18] N. Suri, E. Benvegnu, M. Tortonesi, C. Stefanelli, J. Kovach, and J. Hanna, "Communications middleware for tactical environments: Observations, experiences, and lessons learned," *IEEE Communications Magazine*, vol. 47, no. 10, pp. 56–63, 2009.

[19] N. Suri, G. Benincasa, M. Tortonesi, C. Stefanelli, J. Kovach, R. Winkler, R. Kohler, J. Hanna, L. Pochet, and S. Watson, "Peer-to-peer communications for tactical environments: Observations, requirements, and experiences," *IEEE Communications Magazine*, vol. 48, no. 10, pp. 60–69, 2010.

[20] M. Tortonesi, A. Morelli, C. Stefanelli, R. Kohler, N. Suri, and S. Watson, "Enabling the deployment of cots applications in tactical edge networks," *IEEE Communications Magazine*, vol. 51, no. 10, pp. 66–73, 2013.

[21] G. Benincasa, A. Morelli, C. Stefanelli, N. Suri, and M. Tortonesi, "Agile communication middleware for next-generation mobile heterogeneous networks," *IEEE Software*, vol. 31, no. 2, pp. 54–61, Mar 2014.

[22] M. Tortelli, D. Rossi, G. Boggia, and L. Grieco, "Icn software tools: Survey and cross-comparison," *Simulation Modelling Practice and Theory*, vol. 63, pp. 23 – 46, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X15300654

[23] J. Burke, A. Afanasyev, T. Refaei, and L. Zhang, "Ndn impact on tactical application development," in *IEEE MILCOM*, 2018, Conference Proceedings.

[24] C. Gibson, P. Bermell-Garcia, K. Chan, B. Ko, A. Afanasyev, and L. Zhang, "Opportunities and challenges for named data networking to increase the agility of military coalitions," in *IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI*, 2017, Conference Proceedings, pp. 1–6.

[25] F. T. Johnsen, L. Landmark, M. Hauge, E. Larsen, and Kure, "Publish/subscribe versus a content-based approach for information dissemination," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct 2018, pp. 1–9.

[26] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *ACM CoNEXT*. 1658941: ACM, 2009, Conference Proceedings, pp. 1–12.

[27] C. Gibson, P. Bermell-Garcia, K. Chan, B. Ko, A. Afanasyev, and L. Zhang, "Opportunities and challenges for named data networking to increase the

agility of military coalitions," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug 2017, pp. 1–6.

[28] L. Campioni, M. Hauge, L. Landmark, N. Suri, and M. Tortonesi, "Considerations on the Adoption of Named Data Networking (NDN) in Tactical Environments," in *2019 International Conference on Military Communications and Information Systems (ICMCIS 2019)*, May 2019, pp. 1–8.

[29] S. N. Swamy and S. R. Kota, "An empirical study on system level aspects of internet of things (iot)," *IEEE Access*, vol. 8, pp. 188 082–188 134, 2020.

[30] O. Said, Z. Al-Makhadmeh, and A. Tolba, "Ems: An energy management scheme for green iot environments," *IEEE Access*, vol. 8, pp. 44 983–44 998, 2020.

[31] J. Hwang, A. Aziz, N. Sung, A. Ahmad, F. Le Gall, and J. Song, "Autoconiot: Automated and scalable online conformance testing for iot applications," *IEEE Access*, vol. 8, pp. 43 111–43 121, 2020.

[32] S. Alvisi, F. Casellato, M. Franchini, M. Govoni, C. Luciani, F. Poltronieri, G. Riberto, C. Stefanelli, and M. Tortonesi, "Wireless middleware solutions for smart water metering," *Sensors*, vol. 19, no. 8, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/8/1853

[33] D. Bandyopadhyay and J. Sen, "Internet of things: Applications and challenges in technology and standardization," *Wireless Personal Communications*, vol. 58, pp. 49 – 69, 05 2011.

[34] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[35] E. Al-Masri, K. R. Kalyanam, J. Batts, J. Kim, S. Singh, T. Vo, and C. Yan, "Investigating messaging protocols for the internet of things (iot)," *IEEE Access*, vol. 8, pp. 94 880–94 911, 2020.

[36] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of things (iot) communication protocols: Review," in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 685–690.

[37] R. Muñoz, R. Vilalta, N. Yoshikane, R. Casellas, R. Martínez, T. Tsuritani, and I. Morita, "Integration of iot, transport sdn, and edge/cloud computing for dynamic distribution of iot analytics and efficient use of network resources," *Journal of Lightwave Technology*, vol. 36, no. 7, pp. 1420–1428, 2018.

[38] I. Yaqoob, I. A. T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, "Enabling communication technologies for smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 112–120, 2017.

[39] R. Petrolo, V. Loscrí, and N. Mitton, "Towards a smart city based on cloud of things," in *Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities*, ser. WiMobCity '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 61–66. [Online]. Available: https://doi.org/10.1145/2633661.2633667

[40] K. Townsend, C. Cuf, Akiba, and R. Davidson, *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*, 1st ed. O'Reilly Media, Inc., 2014.

[41] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "Ble beacons for internet of things applications: Survey, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 811–828, 2018.

[42] J.-S. Lee and Y.-C. Huang, "Itri zbnode: A zigbee/ieee 802.15.4 platform for wireless sensor networks." in *SMC*. IEEE, 2006, pp. 1462–1467. [Online]. Available: http://dblp.uni-trier.de/db/conf/smc/smc2006.html#LeeH06

[43] T. de Almeida Oliveira and E. P. Godoy, "Zigbee wireless dynamic sensor networks: Feasibility analysis and implementation guide," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4614–4621, 2016.

[44] M. Tao, X. Hong, C. Qu, J. Zhang, and W. Wei, "Fast access for zigbee-enabled iot devices using raspberry pi," in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 4281–4285.

[45] C. Paetz, *Z-Wave Basics: Remote Control in Smart Homes*. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2013.

[46] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[47] "Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks -

specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. 1–1076, 2007.

[48] S. Saloni and A. Hegde, "Wifi-aware as a connectivity solution for iot pairing iot with wifi aware technology: Enabling new proximity based services," in *2016 International Conference on Internet of Things and Applications (IOTA)*, 2016, pp. 137–142.

[49] D. Saliba, R. Imad, S. Houcke, and B. E. Hassan, "Wifi dimensioning to offload lte in 5g networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0521–0526.

[50] W. Jia, "Enlightenment from the innovative application of 4g communication technology in the mobile library," in *2016 International Conference on Smart City and Systems Engineering (ICSCSE)*, 2016, pp. 153–156.

[51] U. Varshney, "4g wireless networks," *IT Professional*, vol. 14, no. 5, pp. 34–39, 2012.

[52] J. Hu and W. Lu, "Open wireless architecture - the core to 4g mobile communications," in *International Conference on Communication Technology Proceedings, 2003. ICCT 2003.*, vol. 2, 2003, pp. 1337–1342 vol.2.

[53] R. Vidhya and P. Karthik, "Coexistence of cellular iot and 4g networks," in *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2016, pp. 555–558.

[54] A. Hassebo, M. Obaidat, and M. A. Ali, "Commercial 4g lte cellular networks for supporting emerging iot applications," in *2018 Advances in Science and Engineering Technology International Conferences (ASET)*, 2018, pp. 1–6.

[55] B. O. hAnnaidh, P. Fitzgerald, H. Berney, R. Lakshmanan, N. Coburn, S. Geary, and B. Mulvey, "Devices and sensors applicable to 5g system implementations," in *2018 IEEE MTT-S International Microwave Workshop Series on 5G Hardware and System Technologies (IMWS-5G)*, 2018, pp. 1–3.

[56] M. Yang, S. Lim, S.-M. Oh, and J. Shin, "An uplink transmission scheme for tsn service in 5g industrial iot," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 902–904.

[57] A. Yarali, *AI, 5G, and IoT*, 2022, pp. 117–131.

[58] W. Lee, T. Na, and J. Kim, "How to create a network slice? - a 5g core network perspective," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019, pp. 616–619.

[59] D. Sattar and A. Matrawy, "Optimal slice allocation in 5g core networks," *IEEE Networking Letters*, vol. 1, no. 2, pp. 48–51, 2019.

[60] ——, "Proactive and dynamic slice allocation in sliced 5g core networks," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020, pp. 1–8.

[61] A. Khalifeh, K. A. Aldahdouh, K. A. Darabkh, and W. Al-Sit, "A survey of 5g emerging wireless technologies featuring lorawan, sigfox, nb-iot and lte-m," in *2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, 2019, pp. 561–566.

[62] M. Kanj, V. Savaux, and M. Le Guen, "A tutorial on nb-iot physical layer design," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.

[63] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3gpp narrowband internet of things," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 117–123, 2017.

[64] N. Nikolov, "Research of mqtt, coap, http and xmpp iot communication protocols for embedded systems," in *2020 XXIX International Scientific Conference Electronics (ET)*, 2020, pp. 1–4.

[65] I. Kassem and A. Sleit, "Elapsed time of iot application protocol for ecg: A comparative study between coap and mqtt," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2020, pp. 1–6.

[66] C. Sharma and N. K. Gondhi, "Communication protocol stack for constrained iot systems," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018, pp. 1–6.

[67] H. S. Kim, K. Choi, H. Jung, and S. Kim, "A subscription-based push mechanism for iot-icn," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 1201–1203.

[68] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent advances in information-centric networking-based internet of things (icn-iot)," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2128–2158, 2019.

[69] T. Yokotani, "Iot use cases analysis and possibility of adopting icn technologies for these lot use cases," in *2018 IEEE World Symposium on Communication Engineering (WSCE)*, 2018, pp. 1–6.

[70] J. Shahapure, A. Shinde, Y. Madwanna, V. Sontakke, and P. Laturkar, "Iot based system for passenger service in railways with secure icn architecture," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2018, pp. 819–821.

[71] J. Min and Y. Lee, "An experimental view on fairness between http/1.1 and http/2," in *2019 International Conference on Information Networking (ICOIN)*, 2019, pp. 399–401.

[72] N. Oda and S. Yamaguchi, "Http/2 performance evaluation with latency and packet losses," in *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2018, pp. 1–2.

[73] T. Yokotani and Y. Sasaki, "Comparison with http and mqtt on required network resources for iot," in *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*, 2016, pp. 1–6.

[74] B. Wukkadada, K. Wankhede, R. Nambiar, and A. Nair, "Comparison with http and mqtt in internet of things (iot)," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2018, pp. 249–253.

[75] Y. Sueda, M. Sato, and K. Hasuike, "Evaluation of message protocols for iot," in *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science Engineering (BCD)*, 2019, pp. 172–175.

[76] R. T. Fielding and R. N. Taylor, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, 2000, aAI9980887.

[77] H. Garg and M. Dave, "Securing iot devices and securelyconnecting the dots using rest api and middleware," in *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2019, pp. 1–6.

[78] S. Mishra and C. Hota, "A rest framework on iot streams using apache spark for smart cities," in *2019 IEEE 16th India Council International Conference (INDICON)*, 2019, pp. 1–4.

[79] C. V. Phung, J. Dizdarevic, and A. Jukan, "An experimental study of network coded rest http in dynamic iot systems," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[80] D. Ugrenovic and G. Gardasevic, "Coap protocol for web-based monitoring in iot healthcare applications," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, 2015, pp. 79–82.

[81] O. Kleine, "Coap endpoint identification - a protocol extension for crowd sensing in the mobile internet," in *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 2014, pp. 348–351.

[82] W.-T. Su, W.-C. Chen, and C.-C. Chen, "An extensible and transparent thing-to-thing security enhancement for mqtt protocol in iot environment," in *2019 Global IoT Summit (GIoTS)*, 2019, pp. 1–4.

[83] Y. SASAKI, T. YOKOTANI, and H. MUKAI, "Proposals on iot communication through mqtt over l2 network and their performance evaluation," in *2018 International Conference on Innovations in Information Technology (IIT)*, 2018, pp. 30–35.

[84] D. Schuster, P. Grubitzsch, D. Renzel, I. Koren, R. Klauck, and M. Kirsche, "Global-scale federated access to smart objects using xmpp," in *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 2014, pp. 185–192.

[85] M. Hayes and T. Omar, "End to end vanet/ iot communications a 5g smart cities case study approach," in *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2019, pp. 1–5.

[86] H. Wang, D. Xiong, P. Wang, and Y. Liu, "A lightweight xmpp publish/subscribe scheme for resource-constrained iot devices," *IEEE Access*, vol. 5, pp. 16 393–16 405, 2017.

[87] W. Lee, S. Chung, M. Choi, S. Cho, I. Joe, J. Park, S. Lee, and W. Kim, "A robust inter-domain dds gateway based on token passing for large-scale cyber-physical systems," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 868–871.

[88] S. Kumar, B. Bansal, and V. Aggarwal, "Integration of dds based system using routing service," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2020, pp. 142–145.

[89] R. S. Auliva, R.-K. Sheu, D. Liang, and W.-J. Wang, "Iiot testbed: A dds-based emulation tool for industrial iot applications," in *2018 International Conference on System Science and Engineering (ICSSE)*, 2018, pp. 1–4.

[90] S. Saxena, N. A. El-Taweel, H. E. Farag, and L. S. Hilaire, "Design and field implementation of a multi-agent system for voltage regulation using smart inverters and data distribution service (dds)," in *2018 IEEE Electrical Power and Energy Conference (EPEC)*, 2018, pp. 1–6.

[91] M. A. López Peña and I. Muñoz Fernández, "Sat-iot: An architectural model for a high-performance fog/edge/cloud iot platform," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 633–638.

[92] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "A survey on cloudlets, mobile edge, and fog computing," in *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, 2021, pp. 139–142.

[93] F. Firouzi, B. Farahani, E. Panahi, and M. Barzegari, "Task offloading for edge-fog-cloud interplay in the healthcare internet of things (iot)," in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 2021, pp. 1–8.

[94] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[95] S. Naveen and M. R. Kounte, "Key technologies and challenges in iot edge computing," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 61–65.

[96] M. Alrowaily and Z. Lu, "Secure edge computing in iot systems: Review and case studies," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 440–444.

[97] Y. Song, S. S. Yau, R. Yu, X. Zhang, and G. Xue, "An approach to qos-based task distribution in edge computing networks for iot applications," in *2017 IEEE International Conference on Edge Computing (EDGE)*, 2017, pp. 32–39.

[98] C.-L. Tseng and F. J. Lin, "Extending scalability of iot/m2m platforms with fog computing," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 825–830.

[99] M. Tortonesi, M. Govoni, A. Morelli, G. Riberto, C. Stefanelli, and N. Suri, "Taming the iot data deluge: An innovative information-centric service model for fog computing applications," *Future Generation Computer Systems*, vol. 93, pp. 888–902, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17306702

[100] M. Al-khafajiy, T. Baker, A. Waraich, D. Al-Jumeily, and A. Hussain, "Iot-fog optimal workload via fog offloading," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 359–364.

[101] A. Karamoozian, A. Hafid, and E. M. Aboulhamid, "On the fog-cloud cooperation: How fog computing can address latency concerns of iot applications," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2019, pp. 166–172.

[102] A. R. Biswas and R. Giaffreda, "Iot and cloud convergence: Opportunities and challenges," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 375–376.

[103] D. Park and J. Cho, "Cloud-connected code executable iot device with on-cloud virtually memory controller for dynamic instruction streaming," in *2015 International Conference on Cloud Computing and Big Data (CCBD)*, 2015, pp. 29–30.

[104] V. C. Emeakaroha, N. Cafferkey, P. Healy, and J. P. Morrison, "A cloud-based iot data gathering and processing platform," in *2015 3rd International Conference on Future Internet of Things and Cloud*, 2015, pp. 50–57.

[105] G. R. Ceballos and V. M. Larios, "A model to promote citizen driven government in a smart city: Use case at gdl smart city," in *2016 IEEE International Smart Cities Conference (ISC2)*, 2016, pp. 1–6.

[106] M. Abu-Matar and R. Mizouni, "Variability modeling for smart city reference architectures," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–8.

[107] E. Mardacany, "Smart cities characteristics: importance of buit environments components," in *IET Conference on Future Intelligent Cities*, 2014, pp. 1–6.

[108] Y. k. Tolcha, H. M. Nguyen, J. Byun, K. Kwon, J. Han, W. Yoon, N. Lee, H. Kim, N. Pham, and D. Kim, "Oliot-opencity: Open standard interoperable smart city platform," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–8.

[109] M. S. Adam, M. H. Anisi, and I. Ali, "Object tracking sensor networks in smart cities: Taxonomy, architecture, applications, research challenges and future directions," *Future Generation Computer Systems*, vol. 107, pp. 909 – 923, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17310385

[110] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219 – 235, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18319903

[111] M. Jamali, A. Nejat, S. Ghosh, F. Jin, and G. Cao, "Social media data and post-disaster recovery," *International Journal of Information Management*, vol. 44, pp. 25–37, Feb. 2019. [Online]. Available: https://doi.org/10.1016/j.ijinfomgt.2018.09.005

[112] T. H. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, Oct. 2003. [Online]. Available: https://rfc-editor.org/rfc/rfc3626.txt

[113] A. M. Hayajneh, S. A. R. Zaidi, D. C. McLernon, M. D. Renzo, and M. Ghogho, "Performance analysis of UAV enabled disaster recovery networks: A stochastic geometric framework based on cluster processes," *IEEE Access*, vol. 6, pp. 26 215–26 230, 2018. [Online]. Available: https://doi.org/10.1109/access.2018.2835638

[114] Z. Lu, G. Cao, and T. L. Porta, "TeamPhone: Networking SmartPhones for disaster recovery," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3554–3567, Dec. 2017. [Online]. Available: https://doi.org/10.1109/tmc.2017.2695452

[115] E. Ever, E. Gemikonakli, H. X. Nguyen, F. Al-Turjman, and A. Yazici, "Performance evaluation of hybrid disaster recovery framework with d2d communications," *Computer Communications*, vol. 152, pp. 81–92, Feb. 2020. [Online]. Available: https://doi.org/10.1016/j.comcom.2020.01.021

[116] I. F. Kurniawan, A. T. Asyhari, F. He, and Y. Liu, "Mobile computing and communications-driven fog-assisted disaster evacuation techniques for context-aware guidance support: A survey," *Computer Communications*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421002802

[117] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Generation Computer Systems*, vol. 70, pp. 138–147, 2017, cited By 50.

[118] C. Cicconetti, M. Conti, A. Passarella, and D. Sabella, "Toward distributed computing environments with serverless solutions in mec systems," *IEEE Communications Magazine*, vol. 58, no. 3, 2020.

[119] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the internet of things," *Pervasive and Mobile Computing*, vol. 52, pp. 71 – 99, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119218301111

[120] N. Suri, M. Tortonesi, J. Michaelis, P. Budulas, G. Benincasa, S. Russell, C. Stefanelli, and R. Winkler, "Analyzing the applicability of internet of things to the battlefield environment," in *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, 2016, pp. 1–8.

[121] R. Doku, D. B. Rawat, M. Garuba, and L. Njilla, "Fusion of named data networking and blockchain for resilient internet-of-battlefield-things," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–6.

[122] M. Björkbom, J. Timonen, H. Yiğitler, O. Kaltiokallio, J. M. V. García, M. Myrsky, J. Saarinen, M. Korkalainen, C. Çuhac, R. Jäntti, R. Virrankoski, J. Vankka, and H. N. Koivo, "Localization services for online common operational picture and situation awareness," *IEEE Access*, vol. 1, pp. 742–757, 2013.

[123] R. Lenzi, G. Benincasa, E. Casini, N. Suri, A. Morelli, S. Watson and J. Nevitt, "Interconnecting tactical service-oriented infrastructures with federation services," in *Military Communication Conference (MILCOM)*, 2013.

[124] J. Loyall, N. Soule, J. Cleveland, A. Uszok, L. Bunch, and J. Milligan, "A template based approach to specifying the information needs of military missions," in *IEEE Military Communications Conference (MILCOM 2015), Tampa, FL*, 2015, pp. 895–902.

[125] G. Benincasa, L. Bunch, E. Casini, R. Lenzi, A. Morelli, M. Paulini, N. Suri, A. Uszok, "Bridging the gap between enterprise and tactical networks via mission and network sensitive adaptation," in *International Conference on Military Communications and Information Systems (ICMCIS 2018), Warsaw, Poland*, May 2018.

[126] N. Suri, E. Benvegnu, M. Tortonesi, C.Stefanelli, J. Kovach, J. Hanna, "Communications middleware for tactical environments: Observations, experiences, and lessons learned," *IEEE Communications Magazine*, vol. 47, no. 10, pp. 56–63, October 2009.

[127] N. Suri, G. Benincasa, M. Tortonesi, C. Stefanelli, J. Kovach, R. Winkler, U. R. Kohler, J. Hanna, L. Pochet, and S. Watson, "Peer-to-peer communications for tactical environments: Observations, requirements, and experiences," *IEEE Communications Magazine*, vol. 48, no. 10, pp. 60–69, 2010.

[128] A. Bröring, S. K. Datta, and C. Bonnet, "A Categorization of Discovery Technologies for the Internet of Things," in *Proceedings of the 6th International Conference on the Internet of Things*, ser. IoT'16. New York, NY, USA: Association for Computing Machinery, 2016, p. 131–139.

[129] Venanzi *et al.*, "MQTT-Driven Sustainable Node Discovery for Internet of Things-Fog Environments," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[130] G. Tanganelli, E. Mingozzi, C. Vallati, and C. Cicconetti, "A distributed architecture for discovery and access in the internet of things," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2013, pp. 45–46.

[131] G. Tanganelli, C. Vallati, and E. Mingozzi, "Edge-centric distributed discovery and access in the internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 425–438, 2018.

[132] P. Bellavista, D. Belli, S. Chessa, and L. Foschini, "A social-driven edge computing architecture for mobile crowd sensing management," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 68–73, April 2019.

[133] M. A. Rahman and M. S. Hossain, "A location-based mobile crowdsensing framework supporting a massive ad hoc social network environment," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 76–85, March 2017.

[134] M. Pradhan, N. Suri, C. Fuchs, T. H. Bloebaum, and M. Marks, "Toward an architecture and data model to enable interoperability between federated mission networks and iot-enabled smart city environments," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 163–169, 2018.

[135] J.-P. Hubaux, Levente Buttyán, Srdan Capkun, "The quest for security in mobile ad hoc networks," in *2nd ACM International Symposium on Mobile ad hoc networking & computing (MobiHoc 2001)*, 2001, pp. 146–155.

[136] A.K. Lenstra, "Unbelievable security matching aes security using public key systems," *Advances in Cryptology, Lecture Notes in Computer Science*, vol. 2248, 2001.

[137] P. Patil, P. Narayankar, D.G. Narayan, S.M. Meena, "A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish," *Procedia Computer Science*, vol. 78, pp. 617–624, 2016.

[138] J. A. Cooley, R. I. Khazan, S. McVeety, "Secure channel establishment in disadvantaged networks: Optimizing tls using intercepting proxies," in *Military Communications Conference (MILCOM 2010)*, 2010, pp. 32–37.

[139] F. Poltronieri, R. Fronteddu, C. Stefanelli, N. Suri, M. Tortonesi, M. Paulini, J. Milligan, "A secure group communication approach for tactical network environments," in *International Conference on Military Communications and Information Systems (ICMCIS)*, May 2018.

[140] Refaei M., Bush J., "Secure reliable group communication for tactical networks," in *Military Communication Conference (MILCOM), Baltimore, MD, USA*, 2014.

[141] W. Stallings, "Secure group communication using multicast key distribution scheme in ad hoc networks (sgcmkds)," *International Journal of Computer Application*, vol. 1, no. 25, pp. 165–188, 2010.

[142] J. Bethencourt, A. Sahai, B, Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy (IEEE Symposium)*, May 2017.

[143] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, Jun. 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7252.txt

[144] R. Lenzi, G. Benincasa, E. Casini, N. Suri, A. Morelli, S. Watson, and J. Nevitt, "Interconnecting tactical service-oriented infrastructures with federation services," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, 2013, pp. 692–697.

[145] R. Lenzi, N. Suri, A. Uszok, J. Hanna, and J. Milligan, "Supporting information management and information superiority via federation services," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–6.

[146] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, 2012.

[147] N. Suri, A. Hansson, J. Nilsson, P. Lubkowski, K. Marcus, M. Hauge, K. Lee, B. Buchin, L. Mısırhoğlu, and M. Peuhkuri, "A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks," in *2016 International Conference on Military Communications and Information Systems (ICMCIS 2016)*, May 2016, pp. 1–8.

[148] N. Suri, J. Nilsson, A. Hansson, U. Sterner, K. Marcus, L. Misirlioğlu, M. Hauge, M. Peuhkuri, B. Buchin, R. in't Velt, and M. Breedy, "The angloval tactical military scenario and experimentation environment," in *2018 International Conference on Military Communications and Information Systems (ICMCIS 2018)*, May 2018, pp. 1–8.

[149] N. Suri, R. Fronteddu, E. Cramer, M. Breedy, K. Marcus, R. i. '. Velt, J. Nilsson, M. Mantovani, L. Campioni, F. Poltronieri, G. Benincasa, B. Ordway, M. Peuhkuri, and M. Rautenberg, "Experimental evaluation of group communications protocols for tactical data dissemination," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct 2018, pp. 133–139.

[150] N. Suri, M. Breedy, R. Fronteddu, A. Morelli, E. Cramer, J. Nilsson, A. Hansson, K. Marcus, and A. Martens, "Evaluating the scalability of group communication protocols over synchronized cooperative broadcast," in *2021 International Conference on Military Communications and Information Systems (ICMCIS)*, 2021, pp. 1–9.

[151] "AEP-76 VOL IV (RESTRICTED) Specifications Defining The Joint Dismounted Soldier System Interoperability Network (JDSSIN) - Information Exchange Mechanism," NATO, Tech. Rep. [Online]. Available: https://standards.globalspec.com/std/14359780/aep-76-vol-iv

[152] "STANAG 4677 - Dismounted Soldier Systems Standards and Protocols for Command, Control, Communications and Computers (C4) Interoperability (DSSC4 Interoperability)," NATO, Tech. Rep. [Online]. Available: https://standards.globalspec.com/std/9968493/STANAG

[153] G. Benincasa, A. Rossi, N. Suri, M. Tortonesi, and C. Stefanelli, "An experimental evaluation of peer-to-peer reliable multicast protocols," in *2011 - MILCOM 2011 Military Communications Conference*. IEEE, Nov. 2011.

[154] M. Marchini, M. Tortonesi, G. Benincasa, N. Suri, and C. Stefanelli, "Predicting peer interactions for opportunistic information dissemination protocols," in *2012 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, Jul. 2012.

[155] B. Adamson, C. Bormann, M. Handley, and J. Macker, "Nack-oriented reliable multicast (norm) transport protocol," Internet Requests for Comments, RFC Editor, RFC 5740, November 2009.

[156] N. Suri, R. Fronteddu, E. Cramer, M. Breedy, K. Marcus, R. i. t. Velt, J. Nilsson, M. Mantovani, L. Campioni, F. Poltronieri, G. Benincasa, B. Ordway, M. Peuhkuri, and M. Rautenberg, "Experimental evaluation of group communications protocols for tactical data dissemination," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, 2018, pp. 133–139.

[157] N. Suri, M. R. Breedy, K. M. Marcus, R. Fronteddu, E. Cramer, A. Morelli, L. Campioni, M. Provosty, C. Enders, M. Tortonesi, and J. Nilsson, "Experimental evaluation of group communications protocols for data dissemination at the tactical edge," in *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–8.

[158] S. T and S. N. K, "A study on modern messaging systems- kafka, rabbitmq and nats streaming," 2019.

[159] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2656877.2656887

[160] O. Ascigil, S. Reñé, G. Xylomenos, I. Psaras, and G. Pavlou, "A keyword-based icn-iot platform," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ser. ICN '17. New York, NY, USA: ACM, 2017, pp. 22–28. [Online]. Available: http://doi.acm.org/10.1145/3125719.3125733

[161] A. Bouacherine, M. R. Senouci, and B. Merabti, "Multipath forwarding in named data networking: Flow, fairness, and context-awareness," in *E-Business and Telecommunications*, M. S. Obaidat, Ed. Cham: Springer International Publishing, 2017, pp. 23–47.

[162] D. Mars, S. Mettali Gammar, A. Lahmadi, and L. Azouz Saidane, "Using information centric networking in internet of things: A survey,"

Wireless Personal Communications, Jan 2019. [Online]. Available: https://doi.org/10.1007/s11277-018-6104-8

[163] D. Smetters and V. Jacobson, "Securing network content," PARC, Tech Report, Oct. 2009.

[164] J. B. Evans, S. G. Pennington, and B. J. Ewy, "Named data networking protocols for tactical command and control," in *SPIE Defense + Security*, vol. 10651. SPIE, 2018, Conference Proceedings, p. 7.

[165] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: seeing the forest for the trees," in *ACM HotNets*. 2070563: ACM, 2011, Conference Proceedings, pp. 1–6.

[166] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Computer Networks*, vol. 72, pp. 1–13, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614002497

[167] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental evaluation of memory management in content-centric networking," in *IEEE ICC*, 2011, Conference Proceedings, pp. 1–6.

Author's Publication List

Lorenzo Campioni, Niccolò Fontana, Alessandro Morelli, Niranjan Suri, and Mauro Tortonesi. A federated platform to support iot discovery in smart cities and hadr scenarios. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 511–519, 2020.

Lorenzo Campioni, Mariann Hauge, Lars Landmark, Niranjan Suri, and Mauro Tortonesi. Considerations on the adoption of named data networking (ndn) in tactical environments. In *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–8, 2019.

Lorenzo Campioni, Rita Lenzi, Filippo Poltronieri, Manas Pradhan, Mauro Tortonesi, Cesare Stefanelli, and Niranjan Suri. Margot: Dynamic iot resource discovery for hadr environments. In *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pages 809–814, 2019.

Lorenzo Campioni, Mauro Tortonesi, Bastiaan Wissingh, Niranjan Suri, Mariann Hauge, and Lars Landmark. Experimental evaluation of named data networking (ndn) in tactical environments. In *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*, pages 43–48, 2019.

Roberto Fronteddu, Alessandro Morelli, Mattia Mantovani, Blake Ordway, Lorenzo Campioni, Niranjan Suri, and Kelvin M. Marcus. State estimation for tactical networks: Challenges and approaches. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 1042–1048, 2018.

Filippo Poltronieri, Lorenzo Campioni, Rita Lenzi, Alessandro Morelli, Niranjan Suri, and Mauro Tortonesi. Secure multi-domain information sharing in tactical networks. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2018.

Niranjan Suri, Maggie R. Breedy, Kelvin. M. Marcus, Roberto Fronteddu, Eelco Cramer, Alessandro Morelli, Lorenzo Campioni, Mike Provosty, Conner Enders, Mauro Tortonesi, and Jan Nilsson. Experimental evaluation of group communications protocols for data dissemination at the tactical edge. In *2019 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–8, 2019.

Niranjan Suri, Roberto Fronteddu, Eelco Cramer, Maggie Breedy, Kelvin Marcus, Ronald in 't Velt, Jan Nilsson, Mattia Mantovani, Lorenzo Campioni, Filippo Poltronieri, Giacomo Benincasa, Blake Ordway, Markus Peuhkuri, and Mathias Rautenberg. Experimental evaluation of group communications protocols for tactical data dissemination. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 133–139, 2018.