Luostarinen, Riku; Manner, Jukka

Managing Hierarchically Structured DTN-Like Networks

Published in:
Journal of Computer Networks and Communications

DOI:
10.1155/2018/3659546

Published: 01/01/2018

Document Version
Publisher's PDF, also known as Version of record

Hindawi
Journal of Computer Networks and Communications
Volume 2018, Article ID 3659546, 26 pages
https://doi.org/10.1155/2018/3659546

Research Article

Managing Hierarchically Structured DTN-Like Networks

Riku Luostarinen and Jukka Manner

Aalto University School of Electrical Engineering, Department of Communications and Networking, Espoo, Finland

Correspondence should be addressed to Riku Luostarinen; riku.luostarinen@aalto.fi

Received 14 May 2018; Revised 12 August 2018; Accepted 15 August 2018; Published 24 September 2018

Academic Editor: Jemal H. Abawajy

Network management in Delay/Disruption Tolerant Networks (DTNs) is an active research topic and covers topics such as system architecture, roles of actors, and management protocol. The existing solutions either expect a flat management hierarchy or do not address the hierarchical structure used in management. However, in many real-world DTN use cases, particularly in emergency and military contexts, the actors using the DTN system are a part of an organizational or operational hierarchy, and the network design and topology follow the hierarchical structure. This paper introduces a DTN management scheme that is based on that hierarchy. The paper presents a node categorization that is based on the hierarchy, the characteristics of the hierarchical management, roles and responsibilities of the managed and managing nodes in the hierarchy, and the related concept of *management responsibility stack*. Further, the paper discusses the characteristics of the messaging and configurability of the nodes in a hierarchical network, and introduces a problem called *DTN management trilemma*. The paper also presents a use case where the concepts of this paper are applied to network management of a hierarchical organization in a reference scenario, and the performance of the hierarchical management methods is compared to an equivalent nonhierarchical solution.

1. Introduction

The modern Internet is heavily built upon the TCP/IP protocol stack, and the connections are characterized by a low latency, end-to-end connectivity, and low packet loss. The services are typically expected to be always-available and always-on. However, also a large number of other kinds of networks exist. For example, in the military, and in emergency and crisis management operations and environments, fixed network infrastructure may not be available. In these contexts, the organization structure of the operators is typically highly hierarchical, and the network topology follows the structure and patterns of the organization. Further, the network consists of different temporarily set up fixed, vehicular, or mobile nodes that may communicate with each other over various homogeneous wired and wireless links. Due to mobility and the characteristics of the network, the connections are intermittent, and delays and packet losses are high. In these kind of disconnected, intermittent, low-bandwidth (DIL) environments [1], as they are often referred to particularly in the military domain and context, the communication is based on, or has a lot of similarities to, Delay/Disruption Tolerant Networking (DTN) [2, 3].

In traditional IP networks, there must be an end-to-end connectivity between the nodes that want to communicate. In DTNs, no such requirement exists. In DTN architecture [4, 5], a bundle layer is used to form a message-oriented routing overlay on top of the transport layer of the OSI model. The bundle layer uses Bundle Protocol [6] to communicate with other DTN nodes through the underlying heterogeneous link technologies. The messaging is based on *store-carry-and-forward* paradigm and allows nodes to communicate in an intermittently connected network with high delays, high packet loss, and low link capacities.

To monitor, control, and guarantee the operability of the network, network management is needed. In traditional IP-based networks Simple Network Management Protocol (SNMP) [7] is a de facto standard for the management. SNMP runs on top of User Datagram Protocol (UDP) and is supported by a wide range of network devices. In SNMP, the entities that perform management operations are called

managers and those entities that are managed are called agents. The basic functionality of SNMP consists of synchronous queries from managers to agents, query responses from agents to managers, and asynchronous SNMP trap messages that agents send to notify managers of occurred events and condition changes of the managed entity.

IP-based network management tools do not perform well in all networks. In its basic form, SNMP heavily relies on the use of *get*, *get-next*, and *get-bulk* requests from managers to agents and on responses to these requests that are sent from the agents back to the managers. However, when operating in DTNs or other challenged networks, continuous end-to-end connectivity may not exist or cannot be guaranteed. Requests may get lost or they may be so delayed that the response gets outdated before it reaches the manager. Thus, SNMP and other network management protocols that use synchronous messaging and request-response model perform poorly in DTN environment [8]. Instead, the management should use asynchronous mechanisms that are based on intelligent push of management data from agents to managers and on configurable autonomous behaviour of agents.

In this paper, we study the network management of hierarchically structured organizations that operate in challenging environments and in which the network topology follows, or is based on, the organizational hierarchy and patterns. This is typical especially to the military, emergency response agencies, and to crisis management operations. The network management of these organizations significantly differs from management of a flat well-connected network. This paper aims to show how to manage networks of these hierarchical organizations that operate in DTN environment, how the hierarchy affects the management, and what is the performance of the proposed hierarchical management methods compared to an equivalent nonhierarchical solution.

This paper approaches the network management in DTNs from the perspective of a hierarchical network. The paper presents the impact of the hierarchy to (1) management centralization, (2) network quality in different parts of the network, (3) roles and responsibilities of the managing and managed nodes, and (4) configurability and messaging between the nodes. The paper also introduces a use case that shows how network management can be done in a hierarchical military organization in a given reference scenario using the exiting technical solutions and the concepts introduced in this paper. Based on the use case, a performance comparison of hierarchical and nonhierarchical management solutions is presented.

The rest of the paper is organized as follows: In the next section, we go through related work. In Section 3, we discuss the characteristics of hierarchical network management, present a node categorization, and show how the categories are tied to an organizational hierarchy. The next section presents the roles and responsibilities of the nodes and introduces a concept of *Management responsibility stack*. Section 5 is focused on the messaging and configurability, and Section 6 presents a use case that applies the concepts of this paper to a hierarchical organization of a reference military scenario. In Section 7, we provide summary and conclusions.

2. Background and Related Work

Various solutions to network management in DTN environment have been studied and proposed. Peer-to-peer (P2P) technologies, that have been used in management of traditional networks [9, 10], have been applied to DTN context as well [11, 12]. The solutions aim at distributed management and autonomous self-management. Peoples et al. have studied DTN self-management of deep space network [13]. The solution is based on the use of the context-aware broker (CAB) middleware that makes decisions based on policies and contextual data that is autonomously gathered by the managed node. However, the aforementioned solutions do not address management hierarchy.

Pierce-Mayer and Peinado have implemented the DTN-O-Tron node management system [14, 15] for DTN Interplanetary Overlay Network (ION) environment where Contact Graph Routing (CGR) [16, 17] is used. Their approach is similar to the case study presented in Section 6 of this paper in the sense that it is based on DTN management drafts and on-going work in IETF, namely, on DTN Management Protocol (DTNMP) that is a former version of Asynchronous Management Protocol [18] that we use in our use case. However, DTN-O-Tron is nonhierarchical, focused on CGR context, uses additional middleware, and relies on a centralized database that is not applicable in our context.

Papalambrou et al. have implemented a shell script based DTN monitoring solution on the DTN2 reference implementation in a network of three nodes [19]. The implementation is based on static topology with predefined routing tables and a fixed set of monitoring parameters. Thus, the solution is not scalable and cannot be used in a network with mobile nodes. Torgerson has focused on Network Monitor and Control (NM&C) system and management tools in Interplanetary Overlay Network (ION) context [20]. Kumar et al. have studied DTN configuration management and defined Configuration Network Management Protocol (CNMP) [21] that experiments extending NETCONF [22] to DTN environment. However, the results are very preliminary, and the solution does not tackle the fundamental problems of DTN management as it relies on concepts, such as resource locking and acknowledgments of operations, that require bidirectional message exchange.

Ferreira et al. have studied [23, 24] the usage of SNMP in Vehicular Delay-Tolerant Network (VDTN) architecture [25]. Unlike in traditional DTN stack, in the VDTN architecture, the bundle layer resides below the transport and network layers of the OSI model. Also, the bundle layer consists of separate control and data planes. In the study, the IP packets that contain SNMP messages in UDP datagrams were encapsulated into DTN bundles, i.e., the approach is based on carrying IP packets over DTN. SNMPv3 with a customized MIB and out-of-band signaling for the connection establishment was used. The approach was demonstrated in laboratory environment with a manager and a set of static relay nodes and moving vehicles but does not

tackle the fundamental problem related to request-response messaging of SNMP in DTNs. Similarly, the monitoring and management tool of Dias et al. [26] is based on control plane data of the out-of-band signaling of the VDTN architecture.

Another approach is to use in DTNs a management protocol that is tailored to the environment with high delays and packet loss and to guarantee interoperability with Internet management protocols, such as SNMP. Salvador, Macedo, and Nogueira present the HiErarchical MANagement (HE-MAN) architecture [27, 28] for Vehicular Delay-Tolerant Network (VDTN) environment. In the architecture, nodes that are close to each other are clustered together and managed locally using SNMP. For remote monitoring, SNMP over the Bundle Protocol is used in publish-subscribe fashion. The solution is targeted for VDTN applications with very strict delay constraints and requires high-speed and low-delay communication links between the nearby nodes. Thus, it is not suitable for environments where such links do not exist.

Campbell has studied [29] the possibility of using an SNMP gateway on the edge of high and low latency parts of the network. In the study, the gateway gathers data from the DTN to a local database. At the same time, the gateway acts as an SNMP agent towards the low latency network so that the data can be queried from the gateway using SNMP messages. Further, the gateway sends asynchronously SNMP *trap* notification messages to selected SNMP managers in the low latency network. However, the solution has problems with addressing, and it assumes that all the manageable nodes in the DTN network are manually preconfigured to the setup. Thus, it is not scalable or applicable in dynamically changing networks. In the study, the management protocol used on the DTN side of the gateway is Diagnostic Interplanetary Network Gateway (DING).

DING [30] is a network management protocol for environments where traditional IP-based solutions, such as SNMP, do not perform well. In the DING protocol *subscribers* receive information from *providers* based on *subscription requests*. A *subscription request* defines the content that the subscribers want to receive from the providers, and a time interval and possibly a condition for the delivery. The specification of DING is on draft level and, as such, incomplete but can be seen as a predecessor for work related to DTN management that has been done later in IRTF and IETF.

In Section 4, we present the requirements and responsibilities related to network management of hierarchically structured DTN-like networks. Ivancic has written in 2009 an IETF draft [31] that describes requirements for DTN management. The requirements listed in the document are on high level and can be seen as general guidelines and good practices to apply for suitable parts of the specific task at hand. According to the requirements, a DTN system must be manageable both locally (through physical or real time access to the device) and remotely (over DTN). The management must be incremental and support configuration validation and rollback. The nodes must be capable of autonomous behaviour. The document also lists the parameters related to bundle and convergence layer that administrators must be able to monitor and configure and mentions administrative tools that the system should contain. A milestone for updated network management requirements has been set to February 2017 in IETF DTN working group (https://datatracker.ietf.org/wg/dtn/charter/) but has not been finished at the time of writing (5/2018) and is according to IETF mailing list (https://mailarchive.ietf.org/arch/msg/dtn/x-EMgP539vjaN7t10fROkVy2PSg/?qid=be9a200d95c97f7d332bdece1fd9d432) being moved to IRTF NMRG for a reboot.

Besides the DTN management requirements, there are several draft papers and work in progress in IETF related to the network management in DTNs. According to the documents, the network consists of actors that can implement either a role of a managing device (*manager*), managed device (*agent*), or both of them. In this paper, we use the same terminology for consistency. The IETF documents cover the architecture called Asynchronous Management Architecture (AMA) and the roles and responsibilities of its actors [32], data model used by the agents [33], the management protocol called Asynchronous Management Protocol (AMP) that is used between the actors [18], and an interface for applications to interact with the protocol [34]. The management use case shown in Section 6.1 of this paper uses AMA and AMP.

DTN management has adopted a lot of concepts, such as self-configuration, self-healing, and self-optimization, from autonomic computing (AC) [35] and autonomic management [36] of traditional IP networks. In the AMA document [32], four services that must exist in a DTN management system are defined. These services are configuration, autonomous parameterized control, reporting, and administration. The configuration service updates the data of the managed application and is used, e.g., to create new data definitions and reports on the agent. The autonomous parameterized control provides managers with an asynchronous way to change the autonomous behaviour of an agent according to predefined, preconfigured and preprogrammed functions, and parameters that are provided to agents at the execution time of the procedure. The reporting service sends information from agents to managers based on time-based or state-based conditions that are defined and set using the first two services. The administration service is used to enforce the mapping of the other three services between the managers and agents, e.g., to define the reports that can be delivered to certain managers, or configurations that are accepted by the agents.

To provide the services, the agents must be capable of producing and sending information to managers based on the predefined conditions. In AMA context, this is called *Intelligent Push of Information*. The system must use in management uniquely identifiable data elements which identification is not tied to the system configuration. Also, the system must aim at minimizing the message size instead of the processing time and be able to produce tactical data definitions (such as averages, selected samples, or data fusions) based on the existing data. The agents must be capable of autonomous operation, and the managers should configure the autonomy engine of the agents instead of directly changing the states of the agents.

The actors communicate using AMP. In AMP, all the messaging is done using three message types, namely *Register Agent, Perform Control*, and *Data Report* messages.

Register Agent messages are used to notify a manager about the presence of an agent. *Perform Control* messages are used to perform predefined operations on agents such as to add subscriptions or to define variables for which the managed device should gather values from the network. *Data Report* messages contain data that agents send to managers.

Our paper focuses on management of hierarchically structured DTN-like networks that are used, e.g., by the military and many emergency response agencies such as fire departments. The structure of these organizations is based on units of different sizes. Further, organizations of different size and type may have a different number of hierarchical levels. For instance, fire departments have less hierarchical levels than the military. Further, a fire department of a small city has less fire engines and personnel than a fire department of a big city, and thus, less hierarchical levels is needed. For example, the world's largest fire department, New York City Fire Department (http://www.nyc.gov/fdny), is divided to divisions that consist of multiple battalions. Each battalion contains several fire stations that are geographically located in different places and contain varying number of fire companies. Each company is specialized to some particular task. There are, for example, engine companies, ladder companies, and rescue companies and different special units like collapse rescue units and foam units, in different battalions. A company is made up of up to 20 firefighters and led by a captain and (three) lieutenants in his subordination. During a shift, there are three to five firefighters and an officer in a company. Smaller fire departments follow the same structure but may have less hierarchical levels. For example, a fire department of a small town may consist of only one company operating from a single fire station. From the perspective of network management, units of this kind of, or similar, organization can be managed based upon their organizational position as will be shown in the following sections.

3. Node Categorization and Network Management Characteristics

In hierarchically structured organizations, like the military and emergency response agencies, the network topology follows, or is heavily based on, the organizational structure. The users need to communicate and operate in challenging field environments where normal messaging is not possible and thus rely upon DTN or DTN-like concepts. To monitor and control the nodes, and to keep the network up-and-running, network management is needed.

There are characteristics both in hierarchical organizations and in DTNs that heavily influence the network management. On the one hand, a centralized control is wanted. The units higher in the hierarchy want to have a full overview and control over the network. In the parts of the network where they typically operate, the network connections are good and allow exchange of even large amounts of data between the nodes. The good connections together with hierarchical organization structure strongly advocate the centralization of network management to the top of the hierarchy.

On the other hand, on the bottom of the hierarchy, the nodes operate on the field in challenging environments with limited network resources and intermittent connections. In these conditions, no centralized control can be applied. Instead, the network characteristics push the design towards autonomous behaviour and decentralized solutions in the network management.

As the result, we end up with a need for a hybrid that is half-centralized and half-distributed and autonomous. Further, the management and control within the network is highly heterogeneous. In the higher levels of the hierarchy the network consists of static nodes and links with high speed and low packet loss. In terms of network management, the nodes can operate and exchange information the same way as in traditional IP-based networks. However, when the focus is moved down in the hierarchy, the nodes become more mobile, the connection quality and capacity go down drastically, and the concepts of IP network management fade away gradually. On the bottom of the hierarchy, the network is fully intermittently connected, and the network management is decentralized and based on the concepts used in DTNs. That is demonstrated below in Figure 1.

Among a hierarchically structured network with the aforementioned heterogeneous features, groups of nodes with homogeneous characteristics can be found. We have identified four different types of nodes and made a node categorization based on that as follows:

(i) *Core Nodes.* Static infrastructure that is connected to the Delay/Disruption Tolerant Network is used. Characterized by low latency and low packet loss. Suitable for IP traffic. Acts as a gateway to and from the DTN.

(ii) *Transferable Nodes.* Relatively stable nodes that form the core of the DTN. The nodes may be, for example, heavily equipped transferable trailers or vehicles such as trucks with a built-in communication center. Some of the transferable nodes are only deployable, i.e., they are not built to move independently but can be moved when necessary. These nodes are typically set up to, e.g., tents, buildings that reside on the area of operation, or shipping containers (e.g., intermodal containers), and their movement is occasional. Connections to the core network are good and almost always-on. Horizontal connections between the transferable nodes are intermittent, but when a connection between two nodes exists, the delay and packet loss are typically on a tolerable level, and there is enough bandwidth to exchange detailed network management information. Connections downwards in the hierarchy are intermittent, and the connection quality and properties depend on the communication capabilities and the operating environment of the subordinate node. However, when a connection exists, it is typically possible to exchange reports and basic level network management information between the nodes.

(iii) *Vehicular Nodes.* A vehicle, such as a police car/fire truck/ambulance/jeep/tank, with appropriate radio transceivers. In many cases, the vehicles provide

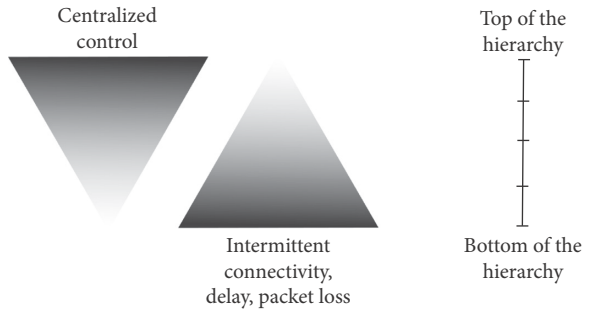


Figure 1: Change in the network quality and management centralization in relation to the hierarchical position in the organization.

users a base station which allows the users to connect to the network. Movement of the nodes is regular but not always predictable. Nodes may switch position multiple times a day or move constantly. Vehicular nodes exchange network management information with each other and forward information about each other to nodes higher in the hierarchy. The exchanged data consists of location and reachability information of the nodes that communicate and those nodes that are connected to them.

(iv) *Terminal Nodes.* Personal devices used by the end-users operating on the field, including, e.g., wearable devices and various rugged personal devices (tablets, cell phones, etc.). The devices move along with the users and are thus always on the move. Terminal nodes connect to the network via a base station that is provided by a vehicular node. The connections are almost always wireless. The network management traffic consists of delivering minimal basic data from the terminal node to the base station and receiving simple commands from nodes higher in the hierarchy. Unlike nodes on the higher levels of the hierarchy, terminal nodes do not need to exchange network management data with each other. Further, as the terminal nodes are on the bottom of the hierarchy, they do not need to control or monitor other nodes. Thus, terminal nodes only act as agents, and there are no managers on the bottom of the hierarchy. Notice that in this context, the term "Terminal" only refers to the devices used on the field in aforementioned conditions. The terminals attached to core, transferable, or vehicular nodes are categorized accordingly (based on their network access).

There is a direct relation between the node type and the node's position in the organization hierarchy. As the network topology follows the organizational hierarchy, the nodes of the same type are close to each other, i.e., on the same level or levels in the hierarchy. The core nodes are located in high levels of the organization hierarchy whereas the terminal nodes typically operate in the bottom of the hierarchy.

For example, in the organization hierarchy of the U.S. army (http://www.army.mil/info/organization/unitsandcommands/oud/, https://www.thebalance.com/u-s-army-military-organization-from-squad-to-corps-4053660), the smallest element in the structure is squad (or a section in case of an armor unit) that consists of 8–16 soldiers. The squad may be organized into smaller teams that have an assigned vehicle. 2–4 squads/sections form a platoon. For example, a mechanized infantry platoon consists of 2 sections that together contain four vehicles. A company contains multiple platoons and has a small headquarters element. A company typically consists of 15–25 vehicles. Battalions are self-sufficient units that are composed of four to six companies and are capable of independent operations to a certain extent. The similar kind of hierarchy that nests units further continues with brigades, divisions, and corps. In this kind of organization hierarchy, the soldiers on the field carry their mobile terminal devices. The vehicles have a networking capability, and they provide the users with an access point to the network. The vehicles are controlled by the company headquarter (HQ). The company HQ is connected upwards in the hierarchy to transferable nodes and further to the core network of the battalion. The relation of the node type to the position of the node in the hierarchy is demonstrated in Figure 2 in the context of the military.

4. Roles and Responsibilities in the Management Hierarchy

By definition, there are two types of actors in a management system, namely, managers and agents. An actor that has a role of a manager controls actors with a role of an agent. A node may act as both manager and agent, i.e., a node may control nodes and at the same time be controlled by other node or nodes. Further, in DTN management solutions, such as AMA, the messaging between the nodes consists of control messages sent from managers to agents, reports delivered from agent to managers, and fusions of reports sent between managers. Thus, there is a many-to-many relationship amongst managers and between managers and agents.

There are several requirements for the actors of a system. We identified three fields of requirements, namely *contextual*, *technical*, and *role-based* requirements. The contextual requirements describe the characteristics and way of usage of particular system. The technical requirements define the system-level technical solution needed to enable management in the given context. The role-based requirements define the requirements that depend on the organizational role and the position of the node in the hierarchy in the given context and show how the contextual and technical requirements are reflected to a single node of a system. Based on these requirements, the responsibilities of the actors can be defined in the corresponding fields. The fields are discussed in detail in the following subsections.

4.1. Contextual Requirements and Responsibilities. The contextual requirements are set by the operating environment, the way of usage, the policies, and the various system-level constraints. Thus, the contextual requirements are

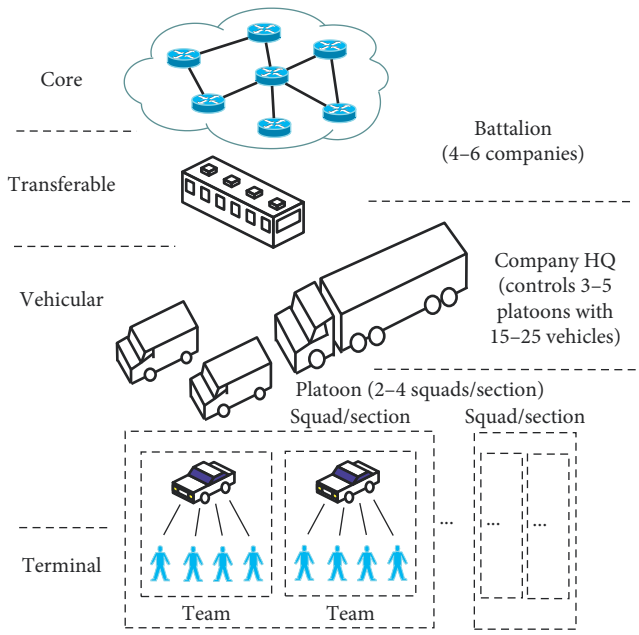


Figure 2: Military hierarchy in relation to node categories.

system-wide but may have different effect depending on a node. This difference is described by the role-based requirements. In our study, the contextual requirements are set by the intermittently connected DTN environment, the hierarchical network topology that follows the organization structure, the way the network is used, and the policy that defines how it must be administered. Based on the requirements, the contextual responsibilities can be defined as follows:

(i) The system must provide the administrators with a way to manage the network over intermittently connected and distributed network.

(ii) The management must be suitable for a hierarchically structured organization in which the network topology follows the organization structure, and the network is heterogeneous both in terms of connection quality and management centralization.

These responsibilities are common to all hierarchically structured DTN-like systems. In a specific system, such as the one discussed in the case study of Section 6, the contextual requirements and responsibilities can be defined in more detail based on the details of the organization, the operating environment, and the network equipment and connectivity of the nodes.

4.2. Technical Requirements and Responsibilities. The technical requirements are related to the technical solution that is used to enable the network management service to the administrators. In other words, they define the system-level solution that is needed in order to manage nodes of a system in the context defined by the contextual requirements. The technical requirements contain, for example, the requirements regarding the management system architecture, the management protocol, and the underlying network and its adapters. Just like contextual requirements, also technical requirements affect nodes differently depending on the role and position of a node in the organization. The role-based requirements and responsibilities describe the way the technical requirements are reflected to a single node.

Various technical solutions for the management of DTN and DTN-like environments have been proposed. Currently, the technical solutions and the roles and responsibilities of DTN management systems are being studied in IETF by Birrane as described in Section 2. In the AMA specification [32], the responsibilities of managers and agents have been defined. The agents must fully support all its Application Data Models (ADM) and locally collect and report all the data defined in them. Further, the agents must provide a configuration service that enables addition, listing, and removal of customized data, reports, macros, and other data definitions. The agents must autonomously execute the controls based on the defined conditions and determine when data must be transmitted to managers. The number of messages sent should be kept as low as possible, e.g., by wrapping multiple reports to a single message. It is allowed for an agent to act as a proxy and perform responsibilities for nodes that do not run an agent software.

The managers must be aware of the ADMs supported by the agents they communicate to and should only refer to information known by the agents. The managers must use controls to define the conditions for data report production in the agents and receive the requested reports asynchronously. Custom data and report definitions should be supported. The managers should also provide an interface to other network management protocols (e.g., SNMP). Managers may produce and exchange fusions of data with other managers.

In hierarchically structured DTN-like networks, data with different granularity are sent in different parts of the network as is described in Section 5. For that, custom data types, data definitions, and reports are needed. Further, fusions of data and relaying of messages are needed in the management. Due to the aforementioned responsibilities of agents and managers, AMA meets these technical requirements and is suitable for network management of hierarchically structured DTN-like networks.

On the protocol level, the system has a responsibility to support the architectural design, i.e., the protocol must implement the functionalities defined in the architecture. In case of AMA, Asynchronous Management Protocol (AMP) [18] is a protocol that meets the requirements of AMA. However, some other compatible protocol could be used as well.

DTN management protocols operate on top of the DTN Bundle Protocol (BP) [6] (or other similar type of protocol) that is capable of transferring management traffic over underlying heterogeneous and intermittently connected network. The BP or its equivalent resides on the Application Layer of the OSI model.

4.3. Role-Based Requirements and Responsibilities. The role-based requirements describe the characteristics of the

management that depend on the position and the organizational role of the node in the hierarchy. In other words, they define how the contextual and technical factors are reflected to a single node of the system. Further, they define the difference of two nodes on a same hierarchical level in terms of network management requirements and responsibilities. For example, there can be two completely identical vehicles acting on the same hierarchical level but only one of them acts as a relay node in the organization network. Thus, the management of the vehicles differs from each other and the role-based requirements and responsibilities describe that difference.

The technical requirements and responsibilities change both vertically between the hierarchical levels and horizontally within a level. For example, on the lowest hierarchy level, more lightweight technical solution can be used compared to the higher levels due to the absence of managers. Further, the characteristics of the network vary between the hierarchy levels, and the technical factors tied to the network quality change accordingly. Thus, certain technical solutions, such as request-response based diagnostics, can be used on a certain hierarchy level but not on another. Also, there is variation in networking equipment in the network both vertically and horizontally, which directly affects the management of the nodes. In the vertical manner, higher levels typically use equipment with more capacity due to heavier network load. Horizontally, nodes on the same hierarchical level may use different networking equipment due to their role in the organization hierarchy and network. For example, in a military context, a medic, an artilleryman, and a scout all have completely different roles in the organization even though they may operate on the same hierarchical level. Further, their need for network access and the requirements and limitations regarding that access differ between the nodes due to their roles. The role-based requirements and responsibilities describe these individual characteristics of the network management of the nodes.

In hierarchical organizations, subordinates are responsible for taking care of the tasks given by their superior. The same applies for the nodes in hierarchical network management. In terms of network management, agents are subordinate to managers. Further, the nodes higher in the management hierarchy control the nodes that are below them. Hence, the nodes have responsibilities set by both their role and their relative position in the management hierarchy.

The nodes communicate and exchange network management traffic in the hierarchy both vertically and horizontally. The direction of the communication and the responsibilities of a node depend on the management role (i.e., the actor type) of the node. Further, as a node can act either as a manager, as an agent, or as both of them, the different nodes have different management role-based responsibilities. These management role-based responsibilities in the hierarchy are shown in Table 1.

The managers control, configure, and monitor the nodes below them in the hierarchy. In case of DTNs, monitoring is typically asynchronous, passive, and based on subscriptions [32]. The managers subscribe to data reports from the agents they want to monitor. The agents report their state to the managers based on conditions defined in the subscriptions.

The responsibilities can also be looked from the perspective of the relative position of the node in the hierarchy as illustrated in Figure 3. In the horizontal manner, the managers share the responsibility of monitoring and reporting with other nodes on the same level in the hierarchy. Thus, the managers exchange information with other managers and relay the data of other managers both horizontally and upwards in the hierarchy based on the given policy.

4.4. Management Responsibility Stack. Based on the contextual, technical, and role-based responsibilities, we define the *Management responsibility stack*. The stack has five layers, namely, *Usage and Context*, *Services and applications*, *Architecture*, *Protocol*, and *Network*. The stack resides fully on the application layer of the OSI model. The stack and the relation between the layers are shown in Figure 4.

Similarly to other layered models, each layer of the stack is responsible for certain duties and tasks on its own layer. Further, each layer has a tight relation to the layer below and above it as the characteristics of one layer affect the layers next to it in the stack. From the management point of view, the responsibilities of the layers are as follows.

The *Usage and Context* layer has a responsibility to set the requirements for the system in the given context, to define the way the system must be used, and to set the bounds for the usage. In relation to the *Services and Applications* layer, the *Usage and Context* layer takes care of the contextual requirements and responsibilities by defining how the applications and services must be implemented and which kind of features are needed in the given context. Further, on node level, it defines technical level requirements for the service/application implementation in the particular node.

The *Services and Applications* layer is responsible for enabling the management functions to the administrators of the system and to provide the administrators with a management service. In relation to other layers, it has a responsibility to provide the *Usage and Context* layer a service that is tailored to the given context and use case. Further, the layer guides architecture design by defining downwards in the stack which of the available architectural solutions are feasible or how a new architecture should be designed and built to support the service. In case the architecture is predefined, the *Services and Applications* layer is responsible for adapting the service to the given architecture.

The *Architecture* layer has a responsibility to implement an architecture that enables the management of a system and to provide upwards in the hierarchy an API that makes it possible to build services and applications on top of the architecture. The *Architecture* layer is tightly coupled with the *Protocol* layer that is responsible for implementing the protocol-level presentation of the messages that are needed in the management. The *Architecture* layer defines the functions and operations used in the management architecture, and that way strongly guides the design of the underlying protocol. On

Table 1: Management role-based responsibilities in the hierarchy.

Actor	Responsibility	Traffic direction
Agent	Report	Upwards in the hierarchy
Manager	Control, configure, and monitor	Downwards in the hierarchy
	Relay management traffic	Sideways and upwards in the hierarchy

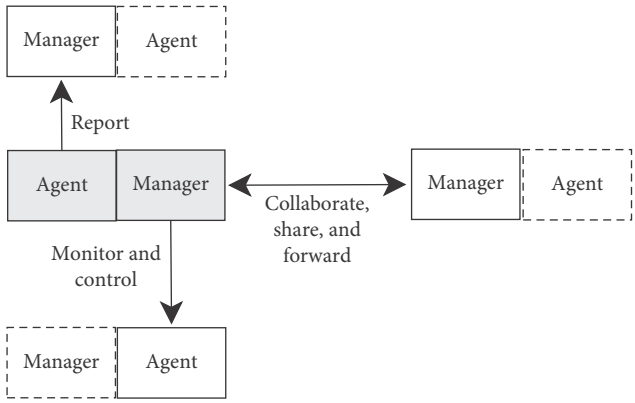


Figure 3: Responsibilities set by the management role (i.e., the actor type) and relative position in the hierarchy from the perspective of the highlighted node (in the middle on the left).

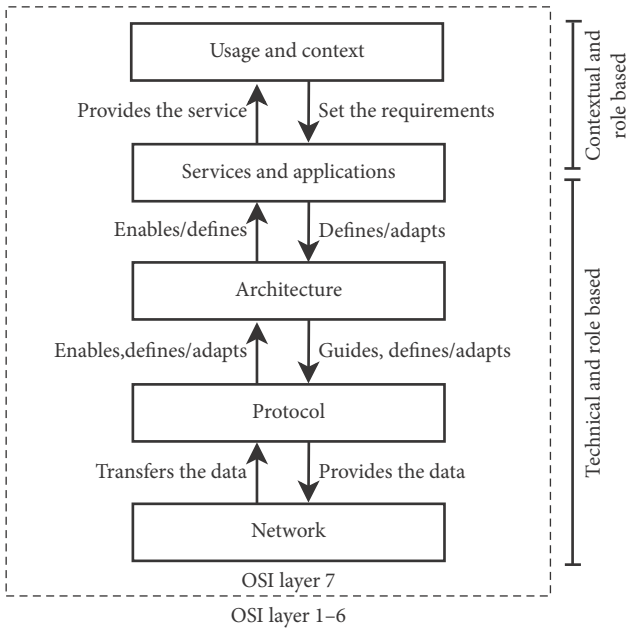


Figure 4: Network management responsibility stack and the relations between the layers. The stack resides on the application layer of the OSI model.

the other hand, the architecture also needs to adapt to the limitations set to the protocol implementation. Further, within these limitations, also the protocol design adapts to the given architecture, i.e., the adaptation process works both ways between the layers. Also, in certain cases, the characteristics of the protocol may guide the architectural decisions made. For example, if the protocol does not support asynchronous messaging, either the architecture must be designed to not rely on it, the underlying protocol must be modified, or the protocol must be changed to one having the support. In summary, the architecture and the protocol go hand in hand and have a strong interconnection so that a change in one usually affects the other.

The *Network* layer is responsible for forming a management network between the actors of a management system. In the management responsibility stack, the term "network" refers to a network on the application layer of the OSI model. Thus, the management network is an application layer logical overlay network on top of the underlying physical network infrastructure. For example, in DTNs, a management overlay network refers to intermittent interconnections between the managing and managed nodes on the DTN bundle layer. Similarly, in case of SNMP, the network consists of those nodes in a UDP/IP network that run SNMP manager or agent software. As the *Network* layer is on the bottom of the stack, it only has an interrelationship upwards in the stack with the *Protocol* layer. The *Protocol* layer must give to the *Network* layer the data of the management protocol in a format that can be carried over the network. In respect of the *Protocol* layer, the *Network* layer has a responsibility to transfer that data between the nodes.

To illustrate the management responsibility stack, the stacks of a hierarchically structured DTN and a non-hierarchical SNMP management network are shown side by side in Figure 5. We can see that, starting from the bottom, the stacks are built on the overlay network of managing and managed nodes, the management protocol, and the architecture that defines the interactions and operations between actors. The actual management service is built on top of the architecture so that it fits in the given context and use case.

5. Messaging and Configurability in Management Hierarchy

Various constraints, requirements, and objectives for DTN management messaging can be identified. In DTN-like networks, bandwidth may be low and transmission window limited, especially in the lower parts of the hierarchical network. When the connections are the bottleneck, the overhead caused by the management must be kept to its minimum. This can be achieved by either keeping the message size really small or by sending slightly bigger messages but with a lower frequency.

Second, from the network manager's point of view, an up-to-date overall picture of the network state is always desired. However, in DTN environment, no timeliness of information can be guaranteed as no status updates can be received from temporarily unreachable or disconnected network nodes. Also, due to disruptions and high packet loss, some of the messages may not be delivered to the managers. Further, too high a sending pace of management traffic may cause congestion and delay the data delivery. However, up to the point of congestion, the higher the frequency of report delivery of the nodes, the more up-to-date information the managers will get from the nodes.

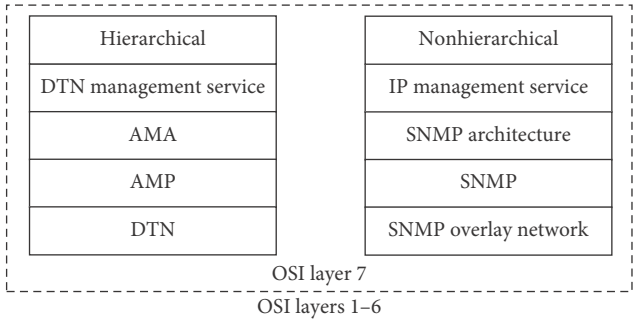


Figure 5: The network management responsibility stack of a hierarchically structured DTN (left) and a nonhierarchical SNMP management network (right).

In addition to up-to-date information, the network status should be as detailed as possible. Naturally, the more the details are wanted, the more the data must be transferred between the nodes, which leads to bigger packet sizes and increases the proportional amount of network management traffic in the network, i.e., the overhead caused by the management.

Consequently, the network management in DTNs should consist of messages that (1) contain lot of details and information about the nodes and messaging between them, (2) are sent frequently enough to give the managers up-to-date information about the state of the network, and (3) cause minimal amount of overhead to the network. However, the objectives are contradictory, and it is impossible to achieve all the three objectives at the same time. Getting up-to-date status of the network with high details increases the management overhead. If high details with small overhead are wanted, the messages must be sent more infrequently. Further, up-to-date information with small overhead can be achieved only by giving up some details from the messaging. In traditional networks, unlike in the DTN environment, the management is typically not bandwidth-bound, and the constraint for small overhead can be ignored. Thus, the problem is characteristic specifically to DTN system and we have named it the *DTN management trilemma*. The trilemma is illustrated in Figure 6 where the achieved properties of messaging can be presented as a point that is placed inside the triangle.

As opposed to management of a flat DTN, in a hierarchical organization, it is possible to make certain assumptions about messaging in different parts of the network. Within each hierarchical level, the network connections have similar properties. In the higher levels of the hierarchy, connections between the nodes are typically better. This allows more frequent report delivery from agents to managers. Also, as there is more bandwidth available, more details can be added to messages. However, this is done in the expense of increasing the packet size. On the lower levels of the hierarchy, the situation is the opposite. When the link capacity is low and transmission windows possibly short, only very few details can be added to messages to keep the message size small. Also, to keep the overhead caused by the management messages on a moderate level, reports must be delivered fairly infrequently.

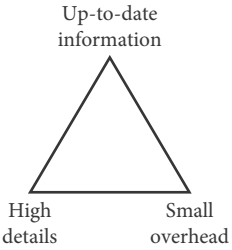


Figure 6: DTN management trilemma: only two out of three desired properties of the management messaging can be achieved simultaneously.

In addition to the connection between the hierarchical levels and the properties of DTN management messaging, a connection between the properties, the hierarchy, and the DTN management trilemma exists. When the trilemma is illustrated as a triangle, the properties of messaging on each hierarchical level can be mapped to the triangle as an area, as shown in Figure 7. In other words, the point representing messaging properties of a single node on a certain hierarchical level resides inside the respective area.

In a hierarchical network, the management traffic consists of horizontal message exchange of managers within a hierarchical level, messages that the managers send to agents that are below them in the hierarchy, and upstream data from the agents to the managers above them, as is shown in Table 1 in Section 4. There are four types of messages, namely, *subscriptions*, *reports*, *control messages*, and *diagnostics messages*. Managers send *subscriptions* to receive *reports* from agents and summary reports from other managers of the system. *Control messages* are used to perform operations on the devices. *Diagnostics messages* are used by utility tools like ping and traceroute.

The authors do recognize that the diagnostic tools are typically based on request-response model, and the use of them is in many contexts seen as a bad practice in DTN environment. However, the need for them cannot be omitted. When used correctly, they provide a powerful tool to help administrators resolve problems in the network, especially in well-connected higher layers of a hierarchical network. Yet, the diagnostics tools should be used infrequently, with care, and always manually as they require a deep knowledge about the underlying network, and the usage may easily cause congestion and overload to the low capacity links.

Network management in DTNs is always a trade-off between flexibility and efficiency. As an extreme, all the definitions related to the messaging could be hardcoded to the system. This would be really efficient as all the actors would initially know the recipients, delivery conditions, and contents of all reports, and no messaging for dynamic definitions, such as agent registrations or report subscriptions, would be needed. However, there would be no flexibility. As the other extreme, all the definitions and parameters related to the management could be defined dynamically. That would make the system fully flexible but also inefficient in terms of resource consumption as a lot a management messaging would be required. Also, advanced

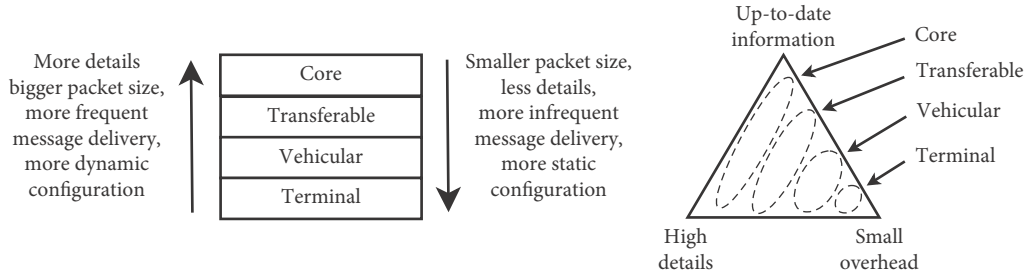


Figure 7: Relation of the management trilemma to the hierarchy.

protocol and data definitions that support the complex dynamic definitions would be needed.

In real-world deployments, some kind of compromise should be used to find a balance between flexibility and efficiency. The system should provide suitable data definitions and a protocol to allow dynamic configurability of selected system parts, but set certain things fixed to keep the complexity of the system on a tolerable level, and that way increase the efficiency.

In a hierarchical system, the details and granularity of information that is sent increase from the bottom to the top of the hierarchy as shown in Figure 7. For that, the data definitions in the bottom of the hierarchy are simple and should be fully or almost fully predefined and preconfigured to the system. From the bottom to the top of the hierarchy, the amount of dynamically configured parameters should increase proportionally. On the highest hierarchy levels, the network connections enable dynamic changes to the configurations.

Based on the gradual change of characteristics of messaging and configurability in proportion to hierarchical levels, the network management traffic can be categorized as shown in Table 2. Terminal nodes act only as agents and exchange no management information with each other. They send upwards in the hierarchy only heartbeat signal that contains very small amount of status information of the device (e.g., uptime), or no data at all. The operations performed by the managers to the agents on terminal nodes are small and simple and consist of actions such as subscriptions of reports and basic configuration changes.

Vehicular nodes gather an overview of status reports of the terminal nodes connected to them and report that, along with their own status, upwards in the hierarchy. To keep the size of the report small, only the most important details are added to the message. Due to the high mobility of vehicular nodes, the nodes need to send location and routing related data to each other and to the nodes above them in the hierarchy. Also, vehicular nodes perform basic maintenance operations to terminal nodes.

Transferable nodes move more infrequently than vehicular nodes, and thus, their management traffic contains less routing related data. Transferable nodes exchange with each other more detailed reports about their own status and the nodes below them in the hierarchy. This detailed data are also sent to core nodes upwards in the hierarchy.

Core nodes want to have an overview of the whole network and the nodes below them. The good network connections allow them to exchange all relevant data with other core nodes and that way monitor the overall state of the organization network. Core nodes receive from transferable nodes detailed summaries about the status of the network and nodes below them in the hierarchy. Core nodes may apply major configuration changes to the network, e.g., install or update software patches to the nodes, or affect the routing of the underlying core links. In the core network, both the messaging and the administrative operations highly resemble management of IP network.

Based on the hierarchy level, a default messaging category for each node category has been defined. In normal circumstances, the nodes should communicate using the default messaging category. However, if the quality of connections rapidly decreases, the managers can notify agents to switch temporarily or permanently to a lower category messaging. Also, agents may autonomously identify such a situation and make the switch without manager intervention. If the connection quality later gets improved again, the nodes should switch back to their default messaging category.

In case of a failure in network, a higher-category messaging may be used to troubleshoot the problem. This is typically done both automatically by the nodes and manually by the network administrators. For example, when the problem occurs, a node may automatically send a detailed report to a predefined manager in the network when possible. Moreover, the administrators may decide to manually use diagnostics tools to further investigate the reason and consequences of the problem. However, these diagnostics tools only function in the well-connected parts of the network, as it is difficult to distinguish failures from natural disconnections in the intermittently connected parts of the network.

6. Case Study of Hierarchical Management in a Reference Scenario

In this section, a case study of network management of a hierarchical military organization is shown in an existing reference scenario. The section consists of an introduction of the scenario and the related data set, analysis of the data, description of the management service, the requirements that were set to it, data definitions that it needs, the implementation of the management tasks using AMA and AMP, a performance comparison to a nonhierarchical management solution, and analysis of the results. The paper is concluded by reflecting the implemented management

Table 2: DTN management messaging categories.

Node category	Default messaging category	Downwards (manager ⟶ agent)	Upwards (agent ⟶ manager)	Horizontally (between managers)
Terminal	Minimal	*No traffic*	Heartbeat	*No traffic*
Vehicular	Basic	Subscription of heartbeat	Status update with most important information	Subscriptions and reports related to the status updates of neighbouring vehicular nodes
Transferable	Detailed	Subscription of the status of the vehicular node and a summary of statuses of terminal nodes	Status update with detailed information	Subscriptions and reports related to the status updates of neighbouring transferable nodes
Core	Full	Subscription of the status of the transferable node. Subscription of an overview of the other transferable nodes and the vehicular nodes below	*No traffic*	Subscriptions and reports to exchange all relevant information that is available between the managers of the core

solution against the concepts introduced earlier in this paper.

6.1. Reference Scenario. The Anglova Scenario [37] has been developed in IST-124 task group of NATO Science & Technology organization. In the scenario, a mechanized battalion performs an operation against insurgent forces in a fictitious area and is supported by a naval component and an unmanned aerial vehicle (UAV). The scenario contains a detailed mobility pattern (available at: http://www.ihmc.us/nomads/scenarios/anglova/ (visited 14.5.2018)) for the battalion for over a two-hour period since the start of the operation. The movement pattern has been developed by military experts to match a realistic operation. The positions of the nodes of the battalion are updated once a second. The scenario consists of three vignettes:

(1) Intelligence preparation of the battlefield
(2) Deployment of the forces
(3) Neutralization of insurgent and explosives and medical evacuation

We will focus on the second vignette where the battalion is deployed to the area of Anglova to perform the operation. The headquarters of the battalion is in the area called Fieldmont and the insurgent forces are located in a town called Wellport. In the operation, the battalion moves from Fieldmont to Wellport to "neutralize the insurgents, and to destroy the armaments they have collected."

During the operation, the battalion moves mainly along roads in a hilly terrain that is covered by forests. The battalion comprises four tank companies ($C_1 \ldots C_4$), one command and artillery company (C_5), and one support and supply company (C_6). The tank companies contain 24 vehicles each. The artillery company and the support and supply company contain 22 and 39 vehicles, respectively.

The operation consists of four phases. In the first phase, the battalion moves in a column away from the headquarters. In the second phase, the battalion is split up and starts to move in two separated groups along the two main roads of the area. In the third phase, the both groups are split up to the company level onto the smaller roads on the area. In the fourth phase, the companies are further split up to the platoons. After the split-up in the beginning of the phase 2, the first half of the battalion moves and splits up further slightly faster than the second half of the battalion.

The battalion Communication and Information System (CIS) is connected to National Operational WAN and the coalition network of NATO Federated Mission Networking (FMN). The vehicles of the companies $C_1 \ldots C_6$ communicate with each other and to the Coalition Headquarters (CHQ) over VHF. As the distance to the CHQ increases during the operation, the VHF connectivity may be poor. Thus, a SATCOM link or communication via UAV can be used as a backup communication channel to the CHQ and also within the battalion that is deployed.

6.2. Data Analysis. In this study, we want to implement a network management service for the battalion of the Anglova scenario. At a given moment of time in the scenario, a connection between some of the nodes does exist and with the others it does not. In this study, we observe the scenario in the time scale of a single phase of the operation. Thus, we are interested in the set of different connections that appear and disappear over time making it possible to communicate in DTN fashion. Further, we are interested in the organizational hierarchy in the different phases and want to identify the connections inside and between different hierarchical levels.

A tank company typically contains three platoons and a headquarters [38]. Depending on the army, there are three to five tanks in each platoon (e.g., in Russia three and in the US army formerly five and nowadays four). In the headquarters, there are additional two tanks and a varying number of other vehicles, such as armored personnel carriers (APC), high-mobility multipurpose wheeled vehicles (HMMWV), and cargo trucks. In addition to that, a maintenance section is normally attached to the company.

No exact information about the vehicles is given in the data set. Yet, based on the node movement and the common structure of army units, relatively precise assumptions about the vehicle types can be made. Further, it is possible to define the positions of the nodes in the organizational hierarchy; we have identified the different companies and roughly divided nodes of each company to platoons.

The data of the Anglova scenario consist latitude and longitude of each of 157 vehicles and a matrix of path losses between every pair of nodes in two different radio frequencies (50 MHz and 300 MHz). In this study, we examine the data set of frequency 50 MHz. The coordinates and the path losses are provided for each second for over two hours from the start of the operation. Based on the given path loss data, we have calculated the connectivity between the nodes during the scenario. Our calculation is based on link budget:

$$P_{RX} = P_{TX} + G - L, \tag{1}$$

where P_{RX} = Received Power (dBm), P_{TX} = Transmitted Power (dBm), G = Gain (dB), and L = Path loss (dB). Based on the information given in the paper [37], we assume the use of NATO Narrow Band Wave Form (NBWF) with the parameters $G = 0\ dB$, $P_{TX} = 37\ dBm$, and $P_{RX} = -100\ dBm$. Thus, we get a receiver's sensitivity threshold S_{RX} for the connectivity between two nodes:

$$\begin{aligned} S_{RX} &\leq P_{TX} + G - P_{RX}, \\ S_{RX} &\leq 137\ dB. \end{aligned} \tag{2}$$

To communicate in traditional IP networks, a full-path connection between the nodes must exist. If the connections are short and intermittent (i.e., they flip-flop), as in the given scenario, IP-based mechanisms perform poorly. However, when the communicating is done in store-carry-forward fashion, even an intermittent connectivity between the nodes is enough to enable messaging.

We have examined each of the four phases of the scenario separately. To take in consideration the DTN nature of the messaging, we look at the connections in each phase in a time window of 10 minutes.

The operation consists of four phases, see Figure 8. In the first phase, the battalion moves in a column away from the headquarters. The tank companies ($C_1 \ldots C_4$) move as a single group in the column. The support and supply company (C_6) is split into two parts and the first of the parts moves in the middle of the column while the remaining part is in the end of the column and leaves the HQ last. The vehicles of the command and artillery company (C_5) are divided along the column already in this early phase of the mission and are in three distinct groups. All the vehicles are on a main road and relatively close to each other and can communicate to each other. The locations of the nodes and the connections between them in the first phase 31 minutes (1860 seconds) after the start of the operation are shown in Figure 8(a).

In the second phase, the battalion is split up and starts to move in two separated groups along the two main roads of the area (Figure 8(b)). The tanks companies stay tightly coupled. The vehicles of the command and artillery company (C_5) are further detached from each other and are joining the tank companies in smaller groups or are being positioned and stopped to some strategic locations in the terrain. The vehicles are still on one of the two main roads of the area and so close to each other that the radio communication between them functions well.

In the third phase, both the main groups are split up to the company level onto the smaller roads on the area. In the fourth phase, the companies are further split up to the platoons. As the vehicles spread on a larger area in a hilly terrain, the path loss between them starts to increase. The split-up takes place at different times in different companies. For example, around 62 minutes after the beginning of the operation Company 1 is already split up to a platoon level, Company 3 is starting to split up, and the rest of the battalion is still formed in companies, as shown in Figure 8(c). In Figure 8(d), the whole battalion is split up to the platoon level.

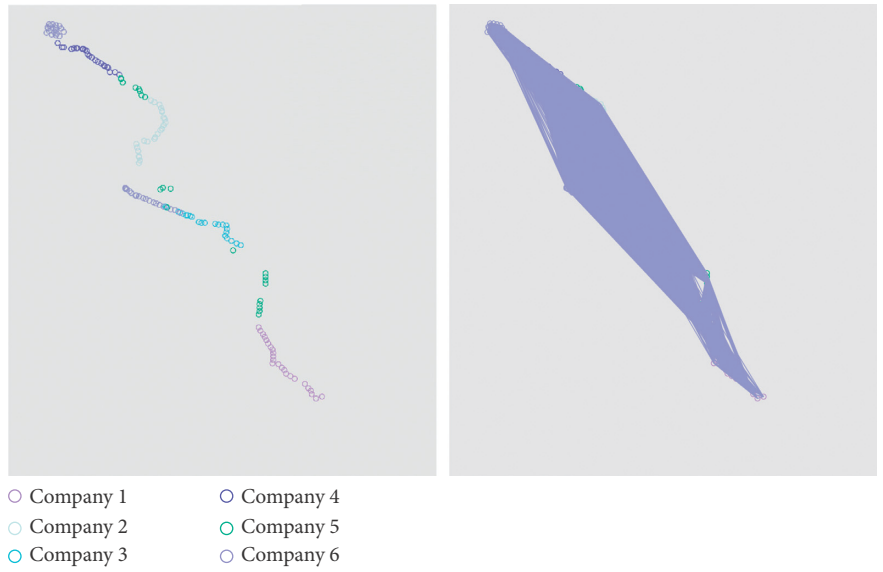
6.3. Requirements for the Management of the Anglova Scenario. We have defined a service to manage the nodes during the operation of the Anglova scenario. For clarity, the service focuses only in status monitoring of the nodes in the network. Resolving the possibly noticed failures and problems is not in the scope of the service.

In the scenario, the DTN-like nature of the communications increases as the organization is split up and the nodes get more spread around the area. Figure 9 shows the logical connections used in our network management service 7780 seconds after the start of the operation. From Figure 8(d) the actual DTN connectivity over the links of the data set can be seen. In DTN management, the reports are delivered upwards and horizontally in the hierarchy and the control traffic downwards and sideways, as described earlier in Section 4.

In the scenario, the Coalition Headquarters (CHQ) is the only core level node and acts as the connection point to the core network. Because the data set contains no information about the CHQ, no horizontal management traffic on the core level is included in our management service. On transferable level, there are five central communication points, formed by nodes of command and artillery company (C_5), that are positioned behind the tank companies (to the northern side) and have a functioning network connectivity. Presumably, these nodes are battalion level command points that control the companies below them in the hierarchy. These nodes have an access to the CHQ and act as messaging relay points for transmission, including network management traffic. Two of these transferable nodes, T_1 and T_2, are connected to the nodes below in the hierarchy, and T_3 relays their management traffic upwards in the hierarchy. The nodes $T_2 \ldots T_5$ exchange network management data directly with the CHQ.

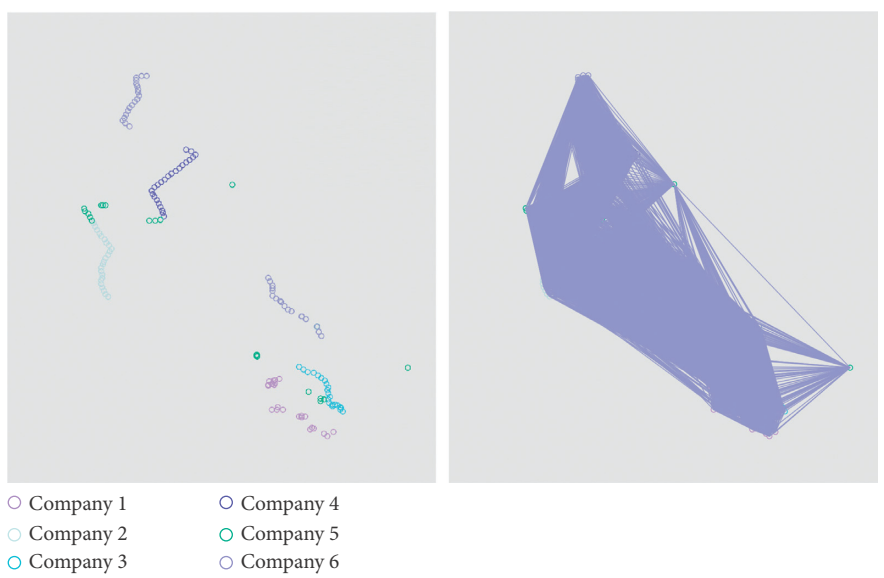
Below the transferable nodes, there are 152 vehicular nodes. The vehicular level nodes contain the nodes of tank companies $C_1 \ldots C_4$, nodes of C_6, and those nodes from C_5 that are not on the transferable level. The vehicular nodes of C_5 are divided geographically into two groups that contain 8 and 9 vehicles. Similarly, the vehicles of C_6 are in groups of 19 and 20 nodes. The vehicular nodes of C_5 and C_6 are connected to transferable nodes T_1 and T_2 as shown in Figure 9.

In the data set, there is no information about the terminal level nodes, i.e., the nodes that are in the hierarchy below the



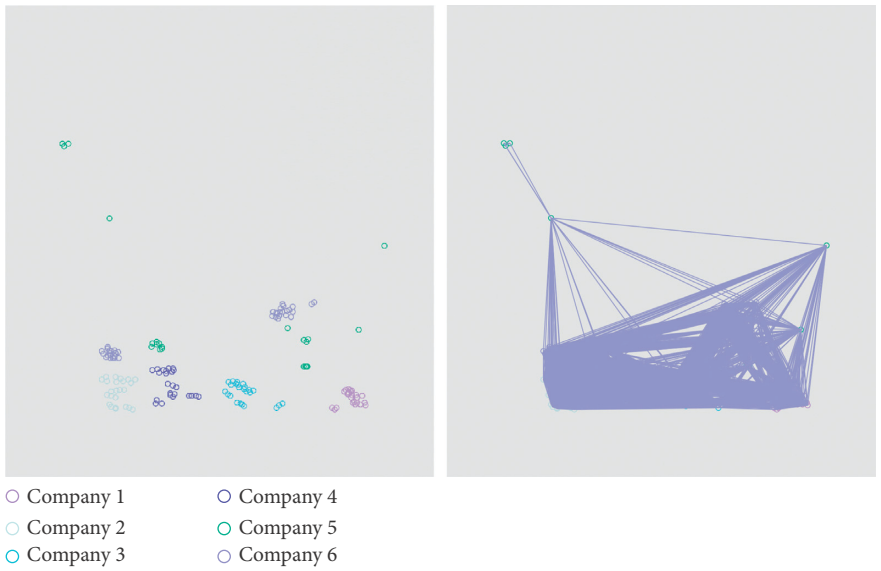
(a)

(b)



(c)

Figure 8: Continued.



(d)

Figure 8: The locations of the nodes and the connections between them in the different phases of the Anglova scenario. (a) Phase 1: the battalion moves in a single column away from the HQ ($t = 1860$ s). (b) Phase 2: the battalion splits up over the two main roads on the area ($t = 2900$ s). (c) Phase 3: the battalion splits up onto many roads grouped in companies. (d) Phase 4: the battalion further splits up to the level of platoons ($t = 7780$ s).

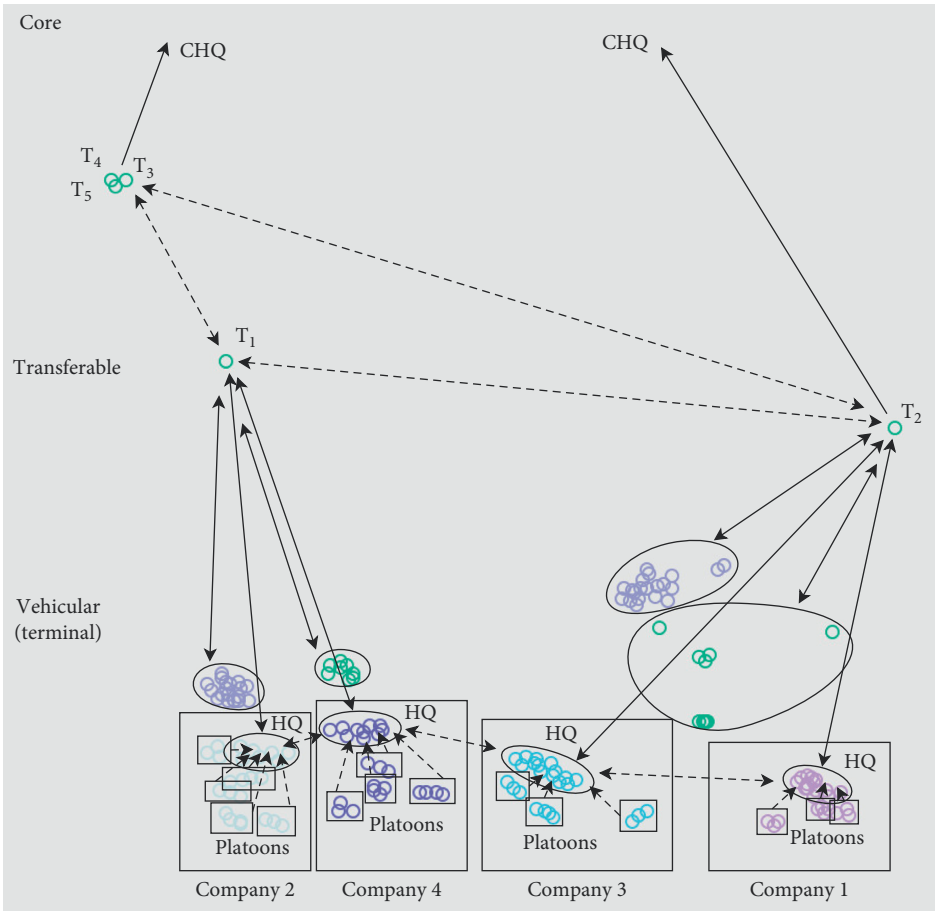


Figure 9: Network management traffic through the hierarchical organization of the Anglova scenario at moment $t = 7780$ s. The solid lines show the traffic upwards and downwards in the hierarchy and the dashed line the horizontal data exchange within the organization.

vehicular nodes. For examplificatory purposes, we assume that in each of the tank companies ($C_1 \ldots C_4$), there are 48 terminal nodes that are in the scope of network management. We assume that all these terminal nodes are connected to the company HQ. Further, we assume that in C_5 and C_6, there are two terminal nodes connected to each vehicular level node of the company. Thus, the number of terminal nodes in C_5 and C_6 are 34 and 78, respectively, and the total number of terminal level nodes in the scenario is 304.

We have defined the required outcomes of the management on the different layers of the organization hierarchy as follows:

(i) *CHQ (Core).* An overview of networking of the battalion is wanted as a part of monitoring of the whole operation.

(ii) *Control and Relay Points of* C_5 *(Transferable).* An overview and status updates of each company is needed to monitor the companies and to make a report to the CHQ. Additionally, horizontal data traffic to exchange status reports between selected transferable nodes (T_1, T_2, and T_3).

(iii) *Tank Company HQ (Vehicular).* Status updates of all the platoons are needed to make sure that the vehicles that belong to the company are reachable and up-and-running. Each company HQ also wants to exchange overview of the company level connectivity with other company HQs in case a backup connection via them is needed, or messages from those companies must be relayed, e.g., due to jamming or loss of certain nodes. All the terminal nodes of the company are monitored by the Company HQ.

(iv) *Platoons of Tank Companies (Vehicular/Terminal).* Reporting responsibility upwards in the hierarchy (to the Company HQ). No horizontal management data exchange.

(v) *Nodes of* C_5 *and* C_6 *(Vehicular/Terminal).* The vehicular nodes have a direct reporting responsibility upwards in the hierarchy. No horizontal management data exchange is made. Terminal nodes of the companies are monitored by the vehicular nodes that belong to the same company.

6.4. Definition of Data Types. The management of the defined use case can be performed using Asynchronous Management Protocol. This subsection describes in detail the AMP message structures that are needed to understand AMP messaging used in our management service, to be able to define messages needed in the management, and to calculate the amount of management data sent during the operation.

We have examined the case based on the draft version 3 of the protocol specification [18]. The predefined OIDs of the controls identified by AMP Agent are defined in AMP Agent Application Data Model (ADM). We used version 0.3 of the Agent ADM which is described in [33]. In AMP, all the management is performed using configurations of ADMs, and communicating with three kinds of messages, namely, *Register Agent*, *Data Report*, and *Perform Control* messages.

In AMP, the messages consist of a one-byte header, a body, and optionally a trailer that contains an access control list. In our implementation, no trailer is attached to the messages. Further, neither positive nor negative acknowledgments (ACK/NACK) for the messages are requested. Messages between the AMP actors are delivered in *Message group* format. *Message group* packs one or more messages together so that they can be delivered as atomic units by an encapsulating protocol used in the communication between the actors. A *Message group* consists of a Self-Delimiting Numeric Value (SDNV) [39] that tells the number of messages in the group, a 5-byte timestamp, and the message data. In our case, only a single message is delivered between the actors at time and the SDNV value takes 1 byte. Thus, the message group and the message header add together an overhead of 7 bytes to each *Register Agent*, *Data Report*, and *Perform Control* message that is sent.

A *Register Agent* message is used to inform a manager about the existence of an agent. The message contains the address of the registering Agent as a Binary Large Object (BLOB), i.e., as the raw bytes of data and its length as an SDNV. If we assume that the system uses 32bit addresses, such as IPv4 addresses or alike, the size of one *Register Agent* message is 5 bytes. Thus, with a message header and message group overhead, the total size of an AMP message containing a registration is 12 bytes.

A *Data Report* message carries report data between the actors. The message consists of a 5-byte timestamp, the recipient address (5 bytes similarly as in case of a *Register Agent* message), number of report entries attached to the message as an SDNV (in this case 1 byte), and the actual report data. On top of that, additional 7-byte overhead of the AMP message group and message headers is added. When each report entry is sent separately, there is 18 bytes of header data, in addition to the actual report data, in each *Data Report* message.

A *Perform Control* message is used to run preconfigured controls on the receiving actor, and can be used, e.g., to create report templates or to add Time-Based Rules to an agent. A *Perform Control* message consists of a timestamp that tells when to run the controls or the macros that are defined in the *Managed Identifier* (MID) collection (MC) of the message. The timestamp is from one to five bytes depending on its value. The controls and macros are identified and parameterized by the Object Identifiers (OIDs) that are defined in the MID of the message. According to the AMP Agent ADM document [33], the Agent ADM Root has a ASN.1 BER-encoded OID 0x2B0601020303 (ID 1.3.6.1.2.3.3) which has a nickname "[8]." Further, Agent Reports (OID 0x2B060102030303) and Agent Controls (OID 0x2B060102030304) have nicknames "[3]" and "[4]," respectively. The nicknames allow the use of compressed parameterized OIDs in the identification of controls and reports. The size of the MID collection of a *Perform Control* message is dependent on the OIDs of the controls that are attached to the message. Thus, each *Perform Control* message consists of 8–12 bytes of fixed data

(depending on the given timestamp) and the data of the controls given in the MID collection.

In the Agent ADM, a set of Externally Defined Data (EDD) is defined. The EDD consist of values that the agent must collect from the node and the underlying network and its adapters. Typically, this requires support from the agent application and the firmware of the node where the agent is run. However, the EDD defined in the specification does not consist of all the needed values for the management of a hierarchical organization in DTN or DTN-like environments. Thus, we have defined an AMP Agent ADM EDD extension for hierarchical management (Table 3). The extension must be implemented and included in the agent and its firmware, so that agents can collect the related data for the reports. The defined EDD extension must be known by all the actors of the system.

In the Agent ADM specification, only one report is predefined. However, in the management of a hierarchical organization, reports with different information granularity are used on different levels of the hierarchy. Thus, custom reports for the management must be defined. The reports use the EDD extension of Table 3. The custom reports needed in hierarchical management are shown in Table 4.

Unlike the EDD that must be predefined to all the actors, the reports are defined only to actors who need them by using *Perform Control* messages. The data that a report contains are defined in a report template. Report templates are created by sending a *Perform Control* message that contains a *AddRptTpl* control (OID [4].9) with appropriate parameters. A *AddRptTpl* control consists of an MID that is used to identify the template defined in the message and an MID collection that defines the contents of the template. Size of a *AddRptTpl* control is $9 + N \times 5$ bytes where N is the number of values in the report (definition of each value in the template takes 5 bytes). We have precalculated and attached the size of each custom template to its own column in Table 4. The *AddRptTpl* control is sent inside a *Perform Control* message that increases the overall size by 20 bytes.

After the definition of the report templates, the reports can be subscribed. To subscribe to the reports, a *GenerateRpts* control (OID [4].9) is encapsulated to a *AddTimeRule* control (OID [4].E). The *AddTimeRule* control is sent in a *Perform Control* message, and the total size of the message is 59 bytes including the AMP message header and message group overhead. The reports are delivered as report entries which contain the values that are defined in the template. The size of an entry of each custom report type, i.e., the size of the report data gathered by the agent, is shown in the last column of Table 4.

For consistency, we have defined the EDD and the report templates so that their OID values are under the Agent ADM root (1.3.6.1.2.3.3) but with some gap to the existing value base. However, the OID values shown in the tables should not be used in real-world deployments because there is a chance of collision with values possibly defined in the future versions of AMP specification. Instead, unique OIDs from a namespace that belongs to the deployment should be selected.

6.5. Implementation. The messaging needed in the monitoring can be divided to (1) preconfiguration and (2) the delivery of monitoring data that is sent from the agents to the managers during the operation. The initial preconfiguration can be done before the troops are deployed. In the preconfiguration, the agents register themselves to the managers using *Register Agent* messages. All agents register themselves to the managers above them in the hierarchy. Additionally, some agents need to send another registration horizontally in the hierarchy to allow data exchange with managers on the same level of the hierarchy. After the registration, the managers define the custom reports needed in the monitoring by using *Perform Control* messages. Managers also use *Perform Control* messages to subscribe to the reports they want to receive from the agents by setting *Time-Based Rules* (TRLs) for report delivery. In the TRLs, agents are requested to generate and send reports to managers once in every 10 minutes. The request is sent only once and the reports are delivered as *Data Report* messages from the agents to the managers periodically throughout the operation. The high-level AMP messaging that is needed to implement a monitoring service of our use case is shown in Figure 10.

As described in the previous subsection, there are 304 terminal nodes, 152 vehicular nodes, five transferable nodes, and one core node within the scope of our management service. The core node acts only as a manager, and the terminal nodes only as agents. Also, the transferable nodes T_4 and T_5 communicate only to the CHQ and act only as agents. The transferable nodes T_1, T_2, and T_3 exchange management data both vertically and horizontally in the hierarchy, and act as both managers and agents. Also, the vehicular nodes of the tank company HQs and all the vehicular level nodes of companies C_5 and C_6 have both the role of a manager and an agent. The rest of the vehicular level nodes of the tank companies $C_1 \ldots C_4$ have no terminal nodes attached to them, so they do not need to run manager software. Thus, in each tank company ($C_1 \ldots C_4$), there is one manager and 23 agents on vehicular level. In addition to that, there are 48 agents on the terminal level. In C_5, there are 17 managers and 17 agents on vehicular level, and 34 agents on the terminal level. In C_6, the number of both managers and agent on vehicular level is 39, and there are 78 agents on the terminal level. Overall, there are a total of 304 agents on the terminal level, 60 managers and 152 agents on the vehicular level, three managers and five agents on the transferable level, and one manager on the core level. Hence, there are a total of 64 managers and 461 agents in the system.

During the preconfiguration phase, agents on all levels register themselves to the manager above them in the hierarchy. For that, 461 *Register Agent* messages are needed. To enable horizontal communication within a hierarchy level, the agent of each company HQ sends another registration to the manager of every other company HQ, and on the transferable level, the agents nodes $T_1 \ldots T_3$ register themselves to the managers of each other (additional 17 registrations). Thus, the total number of *Register Agent* messages that are needed is 478.

Table 3: AMP Agent ADM EDD extension for hierarchical management.

Name	OID	Description	AMP data type	Size in bytes	Primary usage
Uptime	[1].50	Uptime of the node running the agent	TS	4 (or less)	Reports that are sent upwards in the hierarchy
Battery left	[1].51	Remaining battery life (%) of the device running the agent. Value "0" means unknown	UINT	4	Reports from *terminal* nodes to *vehicular* nodes
Num neighs	[1].52	Number of neighbours (unique nodes) this nodes has been communicating with	UINT	4	In horizontal reports sent on *vehicular* level
Num agents	[1].53	Number of agents registered to (the manager run on) this node	UINT	4	In reports on *vehicular* level
Agent statuses	[1].54	List of (id, timestamp) pairs that tell the time when each of the agents that is registered to this node was seen last time	TBL(UINT, TS)	$7 + N \times 9$, where N is the number of agents	In reports on *vehicular* and *transferable* level
Bundles sent	[1].55	Bundles sent since last reboot	UINT	4	In reports on *transferable* and *core* level
Bytes sent	[1].56	Bytes sent since last reboot	UINT	4	In reports on *transferable* and *core* level
Bundles received	[1].57	Bundles received since last reboot	UINT	4	In reports on *transferable* and *core* level
Bytes received	[1].58	Bytes received since last reboot	UINT	4	In reports on *transferable* and *core* level
Messages in queue	[1].59	Number of messages in the queue waiting for delivery	UINT	4	In reports on *transferable* and *core* level
Disk space total	[1].60	Total disk space (in bytes) on the device running the agent	UVAST	8	In reports on *transferable* and *core* level
Disk space free	[1].61	Free disk space (in bytes) available on the device running the agent	UVAST	8	In reports on *transferable* and *core* level
Stats fusion	[1].62	Table that contains a row for each other agent (identified by id) that is on the same hierarchy level and which basic statistics ([1.52], [1].53) are available	TBL(UINT, UINT, UINT)	$9 + N \times 13$, where N is the number of agents	In reports that are sent from *vehicular* nodes to *transferable* nodes
Messaging details fusion	[1].63	Table that contains a row for each other agent (identified by id) that is on the same hierarchy level and for which messaging details, disk space usage, and information about the registered agents (i.e., [1].55, [1].56, [1].57, [1].58, [1].59, [1].60, [1].61) are available	TBL(UINT, UINT, UINT, UINT, UINT, UVAST, UVAST)	$19 + N \times 41$, where N is the number of agents	In reports that are sent from *transferable* nodes to *core* nodes
Full agent details fusion	[1].64	Table that contains a row for each agent (identified by id) registered to this node and contains all the information that the agent sent from itself to this node (i.e., [1].50, [1].55, [1].56, [1].57, [1].58, [1].59, [1].60, [1].61, [1].54, [1].63)	TBL(UINT, UINT, UINT, UINT, UINT, UVAST, UVAST, TBL(UINT, TS), TBL(UINT, UINT, UINT, UINT, UINT, UVAST, UVAST))	$25 + \sum_{n=1}^{N} (71 + J_n \times 9 + K_n \times 41)$ where N is the number of agents registered to the core node, J_n is number agents registered to the nth agent of the core node, and K_n is the number of neighbouring agents (i.e., on the same hierarchy level) of the nth agent of the core node.	In horizontal reports sent on *core* level

Next, the report templates are defined. Each manager sends to the agent below it in the hierarchy the template of the report it wants to subscribe. The template definition is sent as a *Perform Control* message that consists of a *AddRptTpl* control. For the monitoring task of the Anglova scenario, 304 templates of *Terminal status report*, 104 templates of *Intervehicular report*, 60 templates of *Vehicular status report*, 6 templates of *Intertransferable report*, and 4

Table 4: AMP Agent Hierarchical Management Data (HMD) Report Templates.

Name	OID	Description	Primary usage	Definition	Size of *RptTpl control* (bytes)	Size of *Report entry* (bytes)
Terminal status report	[3].50	Battery life and uptime of the terminal node	Upward in the hierarchy from terminal nodes to vehicular nodes	[1].50, [1].51	19	8
Intervehicular report	[3].51	Simple statistical data about the vehicular node (number of agents registered to the manager run on the node, and total number of neighbours of the node)	To share data horizontally between vehicular nodes	[1].52, [1].53	19	8
Vehicular status report	[3].52	Basic information about the vehicular node and the nodes below it in the hierarchy (uptime, status of each agent registered to this node, fusion of statistics received from neighbouring vehicular nodes)	Upward in the hierarchy from vehicular nodes to transferable nodes	[1].50, [1].54, [1].62	24	$20 + N \times 9 + M \times 13$, where N is the number of agents registered to the node, and M is the number of agents from which a *Intervehicular report* has been subscribed
Intertransferable report	[3].53	Detailed data about the transferable node (messaging statistics, disk space usage, information about the agents registered to the node)	To share data horizontally between transferable nodes	[1].55, [1].56, [1].57, [1].58, [1].59, [1].60, [1].61, [1].54	49	$43 + N \times 9$, where N is the number of agents registered to the node
Transferable status report	[3].54	Detailed information about the transferable node and the nodes below it in the hierarchy	Upward in the hierarchy from transferable nodes to core nodes	[1].50, [1].55, [1].56, [1].57, [1].58, [1].59, [1].60, [1].61, [1].54, [1].63	59	$66 + N \times 9 + M \times 41$, where N is the number of agents registered to the node, and M is the number of agents from which a *Intertransferable report* has been subscribed
Intercore report	[3].55	All relevant information about the core node and detailed summary about the nodes below it in the hierarchy	To share data horizontally between core nodes	[1].55, [1].56, [1].57, [1].58, [1].59, [1].60, [1].61, [1].64	49	$61 + \sum_{n=1}^{N} 71 + J_n \times 9 + K_n \times 41$ where N is the number of agents registered to the core node sending the report, J_n is the number of agents registered to the nth agent of the core node, and K_n is the number of agents from which the nth agent of the core node has subscribed an *Intertransferable report*

Note. The sizes of the *RptTpl control* and the resulting report entry do not contain the overhead of the AMP messages carrying the payload, i.e., the header data of *Perform Control* and *Data report* messages, respectively.

templates of *Transferable status report* are needed. After the template definition, the managers subscribe to the defined reports by sending to the agents a *Perform Control* message with a *AddTimeRule* control. The rule makes the agents run a *GenerateRpts* control once in every ten minutes throughout the operation, and as a consequence to produce and deliver a *Data Report* to the manager. Summary of all management messages sent by the monitoring service of the Anglova scenario is shown in Table 5.

6.6. Results and Comparison. In this subsection, the results and performance of the monitoring service of the Anglova scenario are examined. Table 5 shows the number of different management messages sent between the actors, the sizes of the messages, and the number of bytes generated by the management messaging. We can see that the total amount of AMP management messaging is 176837 bytes. 53220 bytes (30.1%) of this is generated in preconfiguration phase and can be sent before the deployment of the troops. The remaining 123617 bytes (69.9%) is sent during the operation. The amount and the type of messaging in different node categories are shown in Figure 11.

On the terminal and transferable levels, around 35.3 and 33.7 kilobytes of data (19.9% and 19.0% of all data) is transferred, respectively. On the terminal level 89.7% and on

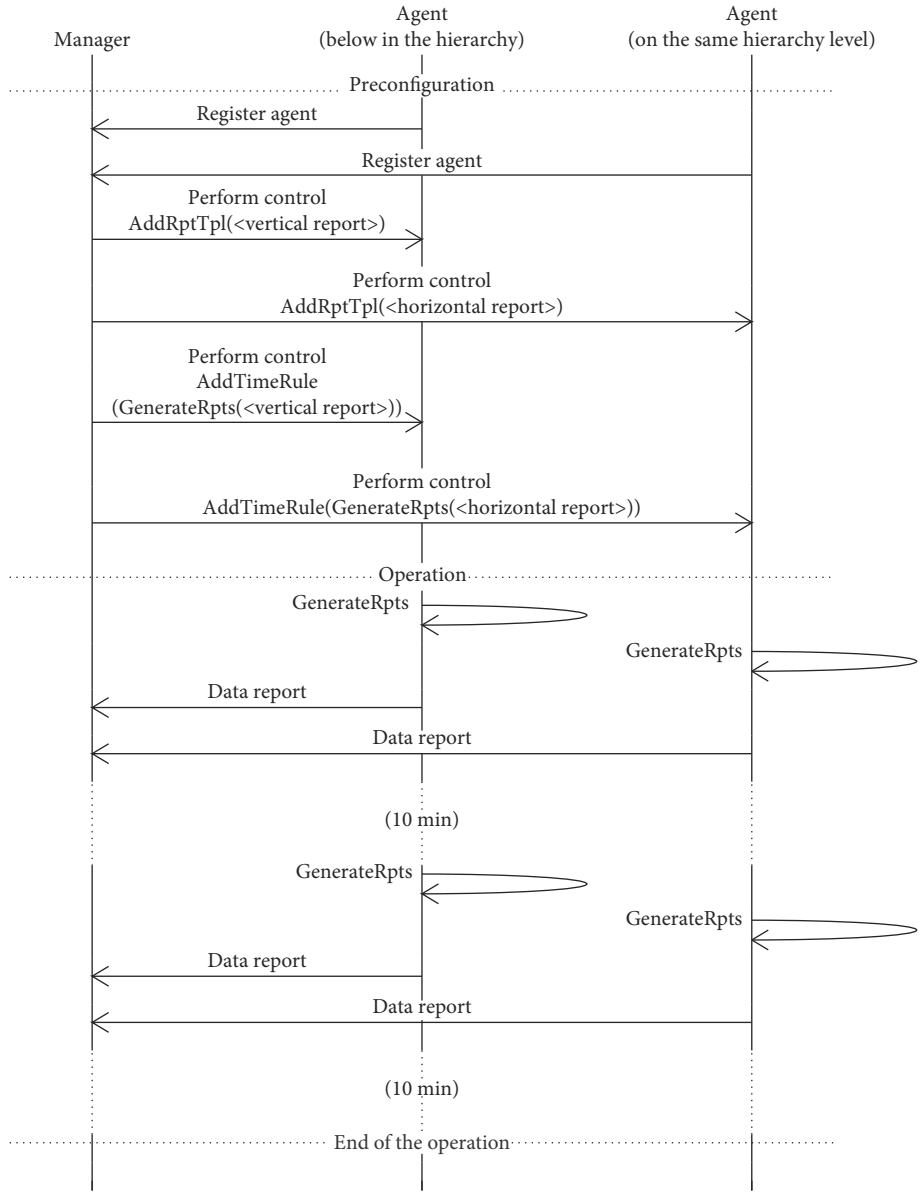


Figure 10: AMP messages sent by our management service.

the transferable level 79.0% of the data consist of data reports sent during the operation, and the rest of the data are registration and definitions sent during the preconfiguration phase of the operation. On the core level, only 552 bytes of data is transferred, which is natural, because there is only one core node and no horizontal data exchange takes place on the core level. Thus, all 552 bytes of data consists of report template definitions and additions of time rules sent to transferable nodes.

Interestingly, on the vehicular level, over 107 kilobytes of data is sent, which is 60.7% of all the management traffic sent during the operation. This high amount of data sent by the vehicular nodes is caused by the organization structure. There are 152 vehicular nodes that manage 304 terminal nodes below them in the hierarchy. Even though the details are kept to a minimum and the messages are relatively small in size, the aggregated amount of data is high due to the large number of nodes on the vehicular level. Also, proportionally large amount (39.1%) of preconfiguration data is needed. On the terminal level, the reports that are sent upwards in the hierarchy are smaller, there is no horizontal data exchange, and the nodes never act as managers, which keeps the amount of preconfiguration data significantly smaller.

Despite that the majority of data transferred were sent by the vehicular nodes, the amount of data sent by a single node was clearly smallest on the terminal level and increased almost linearly between the node categories. On the terminal level, each node sent on average 116 bytes of data during the operation. On the vehicular and transferable levels, the amount of data sent by a node was on average 706 and 6731 bytes, respectively (in case of core nodes the comparison is not relevant due to the absence of horizontal traffic).

To examine the performance improvement gained from the hierarchical solution, also a nonhierarchical implementation of the network management service of the Anglova

Table 5: Hierarchical DTN-level management messaging of the Anglova scenario.

Description	End points	Message (OID)	Total number of messages	Message size (bytes)	Bytes total
Registration	Agent ⟷ manager	*Register Agent*	478	12	5736
Addition of report templates using *Perform Control* messages and *AddRptTpl* control	Vehicular level manager ⟷ terminal level agent	Template for *Terminal status report* ([3].50)	304	39	11856
	Manager of company HQ of C_n ⟷ agent of company HQ of C_n, where $n \in \{1, 2, 3, 4\}$	Template for *Intervehicular report* ([3].51)	92	39	3588
	Manager of company HQ of C_n ⟷ agent of company HQ of C_m, where $n, m \in \{1, 2, 3, 4\}, n \neq m$	Template for *Intervehicular report* ([3].51)	12	39	468
	Transferable level manager ⟷ agent of company HQ of $C_1 \ldots C_4$/Vehicular level agent of $C_5 \ldots C_6$	Template for *Vehicular status report* ([3].52)	60	44	2640
	Manager of transferable node T_n ⟷ agent of transferable node T_m, where $n, m \in \{1, 2, 3\}, n \neq m$	Template for *Intertransferable report* ([3.53])	6	69	414
	Core level manager ⟷ transferable level agent of $T_2 \ldots T_5$	Template for *Transferable status report* ([3].54)	4	79	316
Addition of time-based rules using *Perform Control* messages, and *AddTimeRule* and *GenerateRpts* controls	Vehicular level manager ⟷ terminal level agent	TRL for *Terminal status report* ([3].50) delivery	304	59	17936
	Manager of a company HQ of C_n ⟷ agent of a company HQ of C_n, where $n \in \{1, 2, 3, 4\}$	TRL for *Intervehicular report* ([3].51) delivery	92	59	5428
	Manager of a company HQ of C_n ⟷ agent of a company HQ of C_m, where $n, m \in \{1, 2, 3, 4\}, n \neq m$	TRL for *Intervehicular report* ([3].51) delivery	12	59	708
	Transferable level manager ⟷ agent of company HQ of $C_1 \ldots C_4$/Vehicular level agent of $C_5 \ldots C_6$	TRL for *Vehicular status report* ([3].52) delivery	60	59	3540
	Manager of transferable node T_n ⟷ agent of transferable node T_m, where $n, m \in \{1, 2, 3\}, n \neq m$	TRL for *Intertransferable report* ([3].53]) delivery	6	59	354
	Core level manager ⟷ transferable level agent of $T_2 \ldots T_5$	TRL for *Transferable status report* ([3].54) delivery	4	59	236
Delivery of reports defined in TRLs as *Data Report* messages	Agent of a terminal node ⟷ manager of a company HQ of $C_1 \ldots C_4$/Manager of a vehicular node of $C_5 \ldots C_6$	*Terminal status report* ([3].50)	304 × (1/10 min)× 130 min = 3952	8	31616
	Agent of a vehicular node of a platoon of C_n ⟷ manager of a company HQ of C_n, where $n \in \{1, 2, 3, 4\}$	*Intervehicular report* ([3].51)	92 × (1/10 min) × 130 min = 1196	8	9568
	Agent of a company HQ of C_n ⟷ manager of a company HQ of C_m, where $n, m \in \{1, 2, 3, 4\}, n \neq m$	*Intervehicular report* ([3].51)	12 × (1/10 min) × 130 min = 156	8	1248
	Agent of a company HQ of $C_1 \ldots C_4$ ⟷ manager of a transferable node	*Vehicular status report* ([3].52)	4 × (1/10 min) × 130 min = 52	518	26936
	Agent of a vehicular node of $C_5 \ldots C_6$ ⟷ manager of a transferable node	*Vehicular status report* ([3].52)	56 × (1/10 min) × 130 min = 728	38	27664
	Agent of the transferable node T_1 ⟷ manager of the transferable node T_2/T_3	*Intertransferable report* ([3].53)	2 × (1/10 min) × 130 min = 26	304	7904
	Agent of the transferable node T_2 ⟷ manager of the transferable node T_1/T_3	*Intertransferable report* ([3].53)	2 × (1/10 min) × 130 min = 26	322	8372
	Agent of the transferable node T_3 ⟷ manager of the transferable node T_1/T_2	*Intertransferable report* ([3].53)	2 × (1/10 min) × 130 min = 26	43	1118
	Agent of the transferable node T_2 ⟷ manager of the core node	*Transferable status report* ([3].54)	1 × (1/10 min) × 130 min = 13	427	5551
	Agent of the transferable node T_3 ⟷ manager of the core node	*Transferable status report* ([3].54)	1 × (1/10 min) × 130 min = 13	148	1924
	Agent of the transferable node T_4/T_5 ⟷ manager of the core node	*Transferable status report* ([3].54)	2 × (1/10 min) × 130 min = 26	66	1716
Total					176837

Note. The given message size is the size of the type-specific AMP message including all its headers and payload.

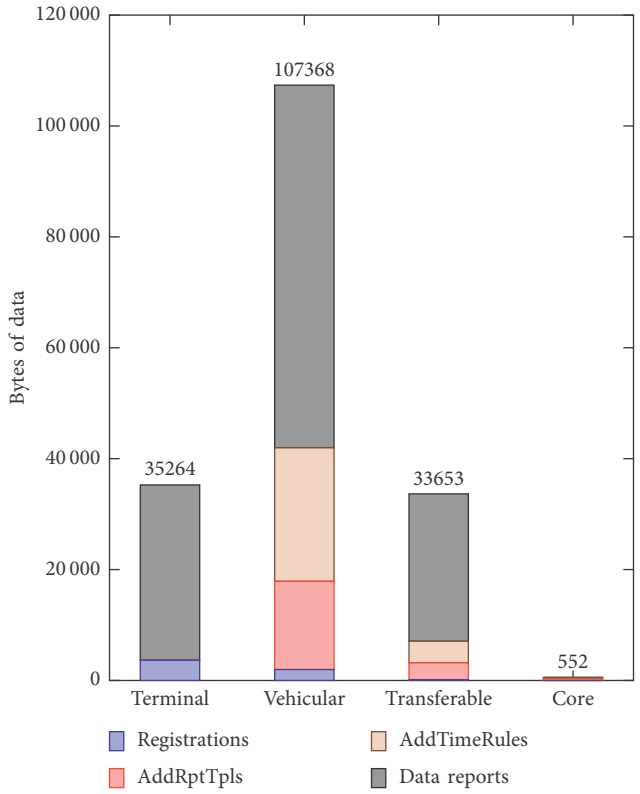


Figure 11: DTN management data sent by nodes in different categories.

scenario was made. For that, the same monitoring service was implemented in a nonhierarchical manner using the same technology (AMA and AMP) as in the implementation of the hierarchical service, which makes the results directly comparable with each other.

Regardless of the flat management hierarchy, the nodes still belong to the organization hierarchy, and the need to monitor them remains the same. In the flat management hierarchy, all the status reports are sent directly between the agent and the manager that needs the information, i.e., the subscriber who is the recipient of the status reports is no longer the next node above in the hierarchy. Thus, there is also no horizontal data exchange for the hierarchy-based summary reports of the nodes below in the hierarchy. The agents register themselves to all the managers that need to be aware of their statuses. To allow messaging that is needed to fulfill the required outcomes that were defined in Section 6.3, all the nodes need to send a registration to the CHQ. Further, the vehicular nodes of $C_1 \ldots C_4$ need to register themselves to the HQs of the companies $C_1 \ldots C_4$ and to the transferable nodes $T_1 \ldots T_3$. The vehicular nodes of C_5 and C_6 need to register themselves to $T_1 \ldots T_3$. The terminal nodes of $C_1 \ldots C_4$ need to send a registration to the HQs of the companies $C_1 \ldots C_4$ and to the transferable nodes $T_1 \ldots T_3$. The terminal nodes of C_5 and C_6 need to register themselves to the vehicular node that is monitoring them (in the same company) and also to the transferable nodes $T_1 \ldots T_3$. The resulting 3089 registration messages make the managers in different positions of the organization hierarchy aware of the agents they need to monitor in order to get the information required by the monitoring service.

The report templates and the resulting reports used by the nonhierarchical service are similar to *Terminal status report* (OID [3].50), *Vehicular status report* (OID [3].52), and *Transferable status report* (OID [3].54) of Table 5 except that no summary of the nodes below in the hierarchy (OIDs [1].62 and [1].63) is added to the reports. For identification purposes, OIDs [3].60, [3].62, and [3].64 were assigned to these nonhierarchical status reports, respectively. The management data that are sent by the nonhierarchical monitoring service of the Anglova scenario are shown in Table 6.

In the nonhierarchical management service, a total of 1,854,972 bytes of management data is transferred between the nodes. 339,965 bytes (18.3%) of that data is sent in the preconfiguration phase and 1,515,007 bytes (81.7%) during the operation. 243,136 bytes (13.1%) of the data is sent by the terminal nodes, 1,269,248 bytes (68.4%) by the vehicular nodes, 297,235 bytes (16.0%) by the transferable nodes, and 45,353 bytes (2.4%) by the core node. Similarly, as in the hierarchical solution, most of the data are sent on the vehicular level. However, in the hierarchical solution, the terminal nodes send 4.8% more data than transferable nodes whereas in the nonhierarchical solution the transferable nodes send 22.3% more data. Further, the core node sends more report templates and subscriptions in the nonhierarchical solution. Figure 12 shows the amount of data sent by both solutions.

The results show that the amount of data sent in the nonhierarchical management service is approximately 10.5 times (1049%) as much as that of the hierarchical management solution. The increased amount of data sent is a direct consequence of the absence of the hierarchical methods. In order to provide the status of the network to the nodes that need it on the different levels of the organization, the statuses are sent multiple times (point-to-point) instead of utilizing the hierarchy-based summary reports. Also, the nonhierarchical model requires more preconfiguration data to enable the messaging between the nodes. Even though there is proportionally less preconfiguration data transferred, the absolute amount of that data is approximately 6.4 times larger in the nonhierarchical service compared to the hierarchical solution. However, as mentioned above, in the nonhierarchical solution, the data definitions and reports do not contain the complex structures required by the hierarchy-based summary messages. In that sense, the hierarchical methods increase the complexity of system but help cut down the amount of data sent between the nodes significantly.

7. Summary and Conclusions

In this paper, we studied network management of organizations that are hierarchically structured and operate in DTN or DTN-like environments. In many of these of organizations, the topology of the network follows, or is based on, the hierarchical structure of the organization. Examples of such organizations include, e.g., the military and different emergency agencies.

Table 6: DTN-level management messaging of the Anglova scenario when the management service is implemented in a nonhierarchical manner.

Description	End points	Message (OID)	Total number of messages	Message size (bytes)	Bytes total
Registration	Agent ⟷ manager	*Register Agent*	3089	12	37068
Addition of report templates using *Perform Control* messages and *AddRptTpl* control	Manager of company HQ of C_n ⟷ agent of a terminal node of C_n, where $n \in \{1, 2, 3, 4\}$	Template for *Terminal status report* ([3].60)	768	39	29952
	Vehicular level manager of $C_5 \ldots C_6$ ⟷ agent of a terminal node of $C_5 \ldots C_6$	Template for *Terminal status report* ([3].60)	112	39	4368
	Manager of $T_1 \ldots T_3$ ⟷ agent of a terminal node of $C_1 \ldots C_6$	Template for *Terminal status report* ([3].60)	912	39	35568
	Core level manager (CHQ) ⟷ terminal level agent	Template for *Terminal status report* ([3].60)	304	39	11856
	Manager of company HQ of C_n ⟷ agent of a vehicular level node of a platoon of $C_1 \ldots C_4$/agent of company HQ of C_m, where $n, m \in \{1, 2, 3, 4\}, n \neq m$	Template for *Vehicular status report* ([3].62)	380	39	14820
	Manager of $T_1 \ldots T_3$ ⟷ agent of a vehicular node of $C_1 \ldots C_6$	Template for *Vehicular status report* ([3].62)	456	39	17784
	Core level manager (CHQ) ⟷ vehicular level agent	Template for *Vehicular status report* ([3].62)	152	39	5928
	Core level manager ⟷ transferable level agent of $T_1 \ldots T_5$	Template for *Transferable status report* ([3].64)	5	74	370
Addition of time-based rules using *Perform Control* messages, and *AddTimeRule* and *GenerateRpts* controls	Manager of company HQ of C_n ⟷ agent of a terminal node of C_n, where $n \in \{1, 2, 3, 4\}$	TRL for *Terminal status report* ([3].60) delivery	768	59	45312
	Vehicular level manager of $C_5 \ldots C_6$ ⟷ agent of a terminal node of $C_5 \ldots C_6$	TRL for *Terminal status report* ([3].60) delivery	112	59	6608
	Manager of $T_1 \ldots T_3$ ⟷ agent of a terminal node of $C_1 \ldots C_6$	TRL for *Terminal status report* ([3].60) delivery	912	59	53808
	Core level manager (CHQ) ⟷ terminal level agent	TRL for *Terminal status report* ([3].60) delivery	304	59	17936
	Manager of company HQ of C_n ⟷ agent of a vehicular level node of a platoon of $C_1 \ldots C_4$/agent of company HQ of C_m, where $n, m \in \{1, 2, 3, 4\}, n \neq m$	TRL for *Vehicular status report* ([3].62) delivery	380	59	22420
	Manager of $T_1 \ldots T_3$ ⟷ agent of a vehicular node of $C_1 \ldots C_6$	TRL for *Vehicular status report* ([3].62) delivery	456	59	26904
	Core level manager (CHQ) ⟷ vehicular level agent	TRL for *Vehicular status report* ([3].62) delivery	152	59	8968
	Core level manager ⟷ transferable level agent of $T_1 \ldots T_5$	TRL for *Transferable status report* ([3].64) delivery	5	59	295

Table 6: Continued.

Description	End points	Message (OID)	Total number of messages	Message size (bytes)	Bytes total
	Agent of a terminal node of $C_n \leftrightarrow$ manager of company HQ of C_n, where $n \in \{1, 2, 3, 4\}$	*Terminal status report* ([3].60)	768 × (1/10 min) × 130 min = 9984	8	79872
	Agent of a terminal node of $C_5 \ldots C_6 \leftrightarrow$ vehicular level manager of $C_5 \ldots C_6$	*Terminal status report* ([3].60)	112 × (1/10 min) × 130 min = 1456	8	11648
	Agent of a terminal node of $C_1 \ldots C_6 \leftrightarrow$ manager of $T_1 \ldots T_3$	*Terminal status report* ([3].60)	912 × (1/10 min) × 130 min = 11856	8	94848
	Terminal level agent $\leftrightarrow$ core level manager (CHQ)	*Terminal status report* ([3].60)	304 × (1/10 min) × 130 min = 3952	8	31616
	Agent of a vehicular level node of a platoon of $C_1 \ldots C_4 \leftrightarrow$ manager of company HQ of C_n, where $n \in \{1, 2, 3, 4\}$	*Vehicular status report* ([3].62)	368 × (1/10 min) × 130 min = 4784	11	52624
Delivery of reports defined in TRLs as *Data Report* messages	Agent of company HQ of $C_n \leftrightarrow$ manager of company HQ of C_m, where $n, m \in \{1, 2, 3, 4\}, n \neq m$	*Vehicular status report* ([3].62)	12 × (1/10 min) × 130 min = 156	2594	404664
	Agent of a vehicular level node of a platoon of $C_1 \ldots C_4 \leftrightarrow$ manager of $T_1 \ldots T_3$/Core level manager (CHQ)	*Vehicular status report* ([3].62)	368 × (1/10 min) × 130 min = 4784	11	52624
	Agent of company HQ of $C_1 \ldots C_4 \leftrightarrow$ manager of $T_1 \ldots T_3$/Core level manager (CHQ)	*Vehicular status report* ([3].62)	16 × (1/10 min) × 130 min = 208	2594	539552
	Agent of a vehicular level node of $C_5 \ldots C_6 \leftrightarrow$ manager of $T_1 \ldots T_3$/Core level manager (CHQ)	*Vehicular status report* ([3].62)	224 × (1/10 min) × 130 min = 2912	29	84448
	Transferable level agent of $T_1 \ldots T_3 \leftrightarrow$ core level manager (CHQ)	*Transferable status report* ([3].64)	3 × (1/10 min) × 130 min = 39	4151	161889
	Agent of the transferable node $T_4/T_5 \leftrightarrow$ core level manager (CHQ)	*Transferable status report* ([3].64)	2 × (1/10 min) × 130 min = 26	47	1222
Total					1854972

Note. The given message size is the size of the type-specific AMP message including all its headers and payload.

In the paper, we described the gradual change in management centralization and network quality from the top to the bottom of the hierarchy. We introduced a node categorization of *core*, *transferable*, *vehicular*, and *terminal* nodes and showed the relation between the node categories and the organization hierarchy. We described how the organization and position of a single node within it affect the role and the responsibilities of the node. Further, we identified three fields of, namely, *contextual*, *technical*, and *role-based*, requirements and responsibilities for each node and defined *management responsibility stack* that describes how the parts of a layered system are interconnected in terms of the responsibilities related to network management. We also described how the hierarchical structure of the organization and the network affect the messaging and configurability of the nodes of a system.

To tie the theory to practice, we defined and implemented a monitoring service for the Anglova scenario. The Anglova scenario is a fictitious military scenario that has been developed by military experts to match a realistic military operation. In the Anglova scenario, a battalion performs a military operation in hilly terrain covered by forests. The battalion consists of four tank companies, a command and artillery company and a support and supply company, which all together contain 157 vehicles. For the operation, detailed movement patterns and radio connectivity between the vehicles are given. We defined and implemented the monitoring service of the Anglova scenario using existing IETF AMA and AMP definitions.

Based on the given information about the companies and the organization structure, the movement patterns of the nodes, and the connectivity between the nodes, we identified the nodes that belong to each of the node categories. The management service was designed so that the Coalition HQ, that is a part of the core network, follows the overall status of the battalion performing the operation. The transferable nodes below the Coalition HQ in the hierarchy belong to the command and artillery company, and monitor the vehicular nodes connected to them. The vehicular nodes further monitor the terminal nodes. Thus, the management centralization decreases in the organization along with the hierarchy level, and the management traffic goes through the four categories of nodes. The terminal nodes deliver status data to nearby vehicles. The vehicles of the platoons of the tank companies connect to the company HQ that belongs to the same node category but is above them in the hierarchy. The company HQs and the vehicular nodes of the command and artillery company and the support and supply company connect to the transferable nodes of the command and artillery company. From the transferable nodes, there is a further connection to the core node of the Coalition HQ. Transferable and vehicular nodes also exchange data horizontally in the hierarchy to produce summaries to nodes above them in the organization.

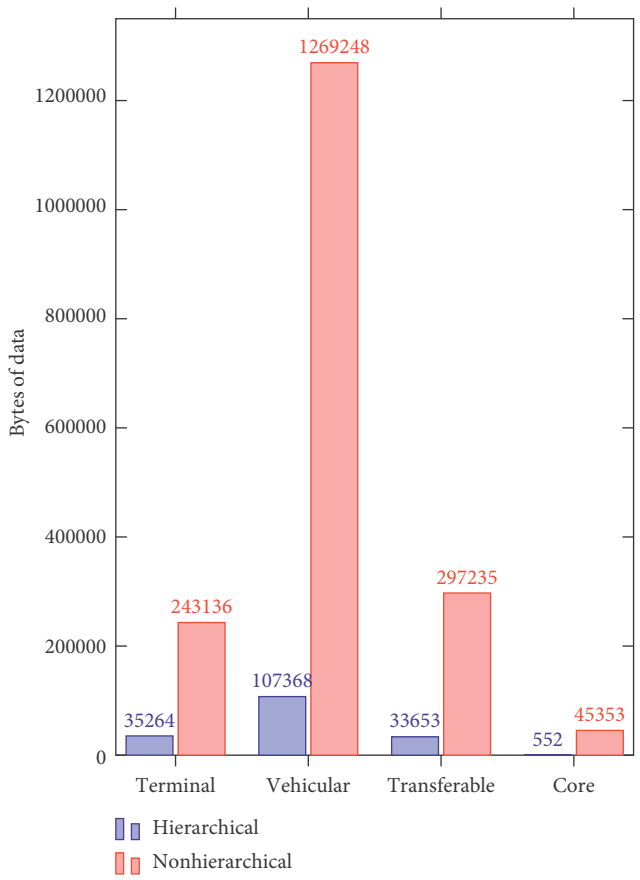


FIGURE 12: Data sent in the hierarchical and nonhierarchical DTN management service implementations of the Anglova scenario.

In the management service, each node has contextual, technical, and role-based requirements and responsibilities. The contextual requirements and responsibilities are system-wide and set by the environment and context for which the service is designed for and the way the service is used and the objectives of that usage. The management service of the Anglova scenario must function in the overlay network formed on top of the available connections between the vehicles of the battalion moving in the terrain of the scenario. Further, the management service must fulfill the required outcomes defined in Section 6.3. Each individual node must meet the technical requirements and responsibilities set by the devices and the radio hardware of the node and AMP and AMA which were used in the service implementation. The role-based requirements and responsibilities reflect the differences in the responsibilities between the nodes on the same hierarchical level. For example, in the scenario, transferable nodes $T_2 \ldots T_3$ communicate with the CHQ, whereas transferable node T_1 only relays traffic to them and has no direct connection to the CHQ. Similarly, on the vehicular level, the HQs of tank companies $C_1 \ldots C_4$ communicate with transferable nodes T_1 and T_2. However, due to their differing role in the hierarchy, the rest of the vehicular nodes of the tank companies have no connections to the nodes of the transferable level.

To adapt to the demands of hierarchical organization, the network management messaging of the Anglova scenario follows the messaging categories defined in Table 2 in Section 5. For that, an extension to EDD of the Agent ADM was made by defining data types needed in the management of a device that is part of a hierarchically structured organization operating in DTN environment. Based on the extension, custom report types for horizontal and vertical communication on each hierarchy level were defined. Finally, the messaging between the managers and the agents was configured to happen so that each actor of the system was able to meet its requirements both vertically and horizontally in the hierarchy as illustrated in Figure 3 in Section 4.

The results show that in the hierarchical management service of the Anglova scenario, a total of 177 kilobytes of data was sent. Approximately, 30.1% of the data consisted of preconfiguration data needed by the management service, and the remaining 69.9% were data reports sent during the operation. To see the performance of our hierarchical solution, a nonhierarchical AMA/AMP-based implementation of the same management service was made. In the nonhierarchical solution, 1855 kilobytes of data was sent and 18.3% of that data was preconfiguration data. The results show that the hierarchical methods require proportionally more preconfiguration and in that way increase the complexity of the system. However, compared to nonhierarchical management, they improve the performance of network management significantly: in case of the Anglova scenario, 90.5% less data was sent when the hierarchical methods were used.

The latency and the delivery rate of the messages are fundamental and widely known problems in DTN systems. In the context of network management service, they cause additional uncertainty and make it difficult to distinguish delayed message delivery caused by natural disconnections from possible network failures. The monitoring service of the Anglova scenario uses the delivery rate of 10 minutes. However, no timeliness or successful delivery of the messages can be guaranteed. As future research, the DTN management solutions should find a way to solve, or mitigate, the impact of the problem. However, currently no such solution exists and also our hierarchical management methods omit the issue.

As the details and message size increase in each node category from the bottom to the top of the hierarchy, the authors expected that the amount of data sent would be shaped accordingly as well. However, it turned out that in the hierarchical monitoring service of the Anglova scenario, the majority of the data were transferred by the vehicular nodes. Further, it turned out that the amount of data sent on the bottom of the hierarchy on the terminal level were about the same as on the transferable level. This seems to be a result of the hierarchical structure of the organization. In a hierarchical military organization, such as the one of the Anglova scenario, there are a large number of vehicular nodes that manage even larger number of terminal nodes below them in the hierarchy. Even though the messages sent between the nodes are relatively small in size and scarce in granularity, the aggregated amount of total data sent in the node categories is relatively high. Further, the large number of nodes and the small message size result in large

proportional amount of data definitions in the preconfiguration phase on the vehicular level.

It would be interesting to see if this kind of traffic shape is specific to the Anglova scenario only, to all military operations, or to all hierarchical organizations in general. However, based on a single scenario, strong conclusions on the traffic shape cannot be drawn. Yet, when observing the topic further in the future, the results shown in this paper can be used as a reference point.

Data Availability

Section 6.1 of this paper uses mobility pattern and pathloss parameters of third-party data of the Anglova Scenario [37] developed as part of the NATO IST-124 Research Task Group on Heterogeneous Networks: Improving Connectivity and Network Efficiency. The data are available at http://www.ihmc.us/nomads/scenarios/anglova/. There are no additional data related to the rest of the sections of this paper.

Disclosure

The research was performed as part of the employment of Aalto University.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors would like to thank Aleksi Marttinen from Aalto University for his help with the link budget calculation and radio parameter estimation for the analysis of the Anglova Scenario in Section 6.2.

References

[1] K. Scott, T. Refaei, N. Trivedi, J. Trinh, and J. P. Macker, "Robust communications for disconnected, intermittent, low-bandwidth (DIL) environments," in *Proceedings of Military Communications Conference (MILCOM)*, pp. 1009–1014, Baltimore, MD, USA, November 2011.

[2] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, pp. 27–34, ACM, Karlsruhe, Germany, August 2003.

[3] A. McMahon and S. Farrell, "Delay- and disruption-tolerant networking," *IEEE Internet Computing*, vol. 13, no. 6, pp. 82–87, 2009.

[4] V. Cerf, S. Burleigh, A. Hooke et al., "Delay-tolerant networking architecture," RFC 4838, 2007.

[5] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 828–836, 2008.

[6] K. Scott and S. Burleigh, "Bundle protocol specification," RFC 5050, 2007.

[7] R. Presuhn, "Version 2 of the protocol operations for the simple network management protocol (SNMP)," RFC 3416, 2002.

[8] E. Birrane and H. Kruse, "Delay-tolerant network management: the definition and exchange of infrastructure information in high delay environments," in *Proceedings of Infotech@Aerospace*, American Institute of Aeronautics and Astronautics, St. Louis, MO, USA, March 2011.

[9] L. Z. Granville, D. M. Da Rosa, A. Panisson, C. Melchiors, M. J. B. Almeida, and L. M. R. Tarouco, "Managing computer networks using peer-to-peer technologies," *IEEE Communications Magazine*, vol. 43, no. 10, pp. 62–68, 2005.

[10] J. C. Nobre, C. Melchiors, C. C. Marquezan, L. M. R. Tarouco, and L. Z. Granville, "A survey on the use of P2P technology for network management," *Journal of Network and Systems Management*, vol. 26, no. 1, pp. 189–221, 2018.

[11] J. C. Nobre, P. A. P. R. Duarte, L. Z. Granville, and L. M. R. Tarouco, "Delay-tolerant management using self-∗ properties and P2P technology," in *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pp. 728–731, Ghent, Belgium, May 2013.

[12] J. C. Nobre, P. A. P. R. Duarte, L. Z. Granville, L. M. R. Tarouco, and F. J. Bertinatto, "On using P2P technology to enable opportunistic management in DTNs through statistical estimation," in *Proceedings of IEEE International Conference on Communications (ICC)*, pp. 3124–3129, Sydney, Australia, June 2014.

[13] C. Peoples, G. Parr, B. Scotney, and A. Moore, "Context-aware policy-based framework for self-management in delay-tolerant networks: a case study for deep space exploration," *IEEE Communications Magazine*, vol. 48, no. 7, pp. 102–109, 2010.

[14] J. Pierce-Mayer and O. Peinado, "DTN-O-Tron: a system for the user-guided semi-autonomous generation and distribution of CGR contact plans," in *Proceedings of IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pp. 1–4, Aachen Germany, December 2015.

[15] J. Pierce-Mayer and O. Peinado, "DTN network management," in *Proceedings of 14th International Conference on Space Operations*, Deajeon, Korea, May 2016.

[16] S. Burleigh, "Contact graph routing," Internet-Draft, Version 1, July 2010, https://tools.ietf.org/html/draft-burleigh-dtnrg-cgr-01.

[17] G. Araniti, N. Bezirgiannidis, E. Birrane et al., "Contact graph routing in DTN space networks: overview, enhancements and performance," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 38–46, 2015.

[18] J. Mayer and E. Birrane, "Asynchronous management protocol," Draft Version 3, Internet-Draft, June 2016.

[19] A. Papalambrou, A. G. Voyiatzis, D. N. Serpanos, and P. Soufrilas, "Monitoring of a DTN2 network," in *Proceedings of Baltic Congress on Future Internet and Communications*, pp. 116–119, Riga, Latvia, February 2011.

[20] J. L. Torgerson, "Network monitor and control of disruption-tolerant networks," in *Proceedings of 13th International Conference on Space Operations (SpaceOps)*, Pasadena, CA, USA, May 2014.

[21] S. Kumar, A. Mishra, and G. C. Robert, "Configuration management for DTNs," in *Proceedings of IEEE Globecom Workshops*, pp. 589–594, Singapore, December 2010.

[22] R. Enns, M. Bjorklund, A. Bierman, and J. Schönwälder, "Network configuration protocol (NETCONF)," RFC 6241, 2011.

[23] B. F. Ferreira, J. N. Isento, J. A. Dias, J. J .P. C. Rodrigues, and L. Zhou, "An SNMP-based solution for vehicular delay-tolerant network management," in *Proceedings of IEEE*

Global Communications Conference (GLOBECOM), pp. 250–255, Anaheim, CA, USA, Dec 2012.

[24] B. F. Ferreira, J. J. P. C. Rodrigues, J. A. Dias, and J. N. Isento, "Man4VDTN–a network management solution for Vehicular Delay-Tolerant Networks," *Computer Communications*, vol. 39, pp. 3–10, 2014.

[25] V. N. G. J. Soares, F. Farahmand, and J. J. P. C. Rodrigues, "A layered architecture for vehicular delay-tolerant networks," in *Proceedings of IEEE Symposium on Computers and Communications (ISCC)*, pp. 122–127, Sousse, Tunisia, July 2009.

[26] J. A. F. F. Dias, J. J. P. C. Rodrigues, J. F. de Paz, and J. M. Corchado, "MoM–a real time monitoring and management tool to improve the performance of vehicular delay tolerant networks," in *Proceedings of Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 1071–1076, Vienna, Austria, July 2016.

[27] E. M. Salvador, D. F. Macedo, J. M. Nogueira, V. D. D. Almeida, and L. Z. Granville, "Hierarchy-based monitoring of vehicular delay-tolerant networks," in *Proceedings of 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pp. 447–452, Las Vegas, NV, USA, January 2016.

[28] E. M. Salvador, D. F. Macedo, and J. M. S. Nogueira, "HEMAN: hierarchical management for vehicular delay-tolerant networks," *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 663–685, 2017.

[29] G. L. Campbell, "An SNMP gateway for delay/disruption tolerant network management," McClure School of Information and Telecommunication Systems Technical, a Project Report adviced by Hans Kruse, August 2010. http://www.its.ohiou.edu/kruse/publications/Campbell_SNMP_Gateway.pdf.

[30] H. Kruse, S. Ostermann, G. Clark, and G. Campbell, "DING protocol - a protocol for network management," February 2010. https://tools.ietf.org/html/draft-irtf-dtnrg-ding-network-management-02.

[31] W. Ivancic, "Delay/disruption tolerant networking-network management requirements," Internet-Draft, version 00, 2009. https://tools.ietf.org/html/draft-ivancic-dtnrg-network-management-reqs-00.

[32] E. Birrane, "Asynchronous management architecture," Draft Version 3. Internet-Draft, June 2016.

[33] E. Birrane, "Asynchronous management protocol agent application data model," Draft Version 02. Internet-Draft, June 2016.

[34] E. Birrane, M. Sinkiat, and S. Jacobs, "AMP manager SQL interface," Internet-Draft, September 2015.

[35] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[36] N. Samaan and A. Karmouch, "Towards autonomic network management: an analysis of current and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 22–36, 2009.

[37] N. Suri, A. Hansson, J. Nilsson et al., "A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks," in *Proceedings of IEEE International Conference on Military Communications and Information Systems (ICMCIS 2016)*, pp. 1–8, Brussels, Belgium, May 2016.

[38] United States Army, *United States Army Field Manual 17-15*, Headquarters, Department of the Army, County, VA, USA, 1996.

[39] W. Eddy and E. Davies, "Using self-delimiting numeric values in protocols," RFC 6256 (Informational), 2011.