A Unified Naming and Addressing Scheme for Hybrid DTN/NDN Communication Protocols

Ronald Langrin[1], **Jeremy Blackstone**[1]

[1]Research Institute for Tactical Autonomy, Howard University, USA
ronald.langrin@rita.howard.edu, jeremy.blackstone@rita.howard.edu

Abstract

The diversity of networking paradigms such as Delay/Disruption Tolerant Networking (DTN) and Named Data Networking (NDN) presents interoperability challenges that limit seamless communication across heterogeneous networks due to their differing addressing and naming schemes. This paper proposes a consistent unified naming convention scheme with a prefix ”uni/...” inspired by DTNs URI prefix ”dtn://...” to facilitate compatibility and interoperability between DTN and NDN without modifying existing protocols while maintaining performance. Specifically, we design and implement a unified naming convention and protocol mapping scheme as containerized micro-services deployed in a middleware controller. The preliminary experimental results demonstrate that our hybrid approach effectively bridges the gap between DTN and NDN, enabling efficient and consistent communication for heterogeneous networked applications.

Introduction

In the evolving landscape of networking, various paradigms have been developed to address specific communication challenges. The two widely deployed protocols are Delay/Disruption Tolerant Networking (DTN) (Burleigh et al. 2003) and Named Data Networking (NDN) (Zhang et al. 2014). DTN is designed to provide reliable data transmission in environments with intermittent connectivity, long delays, or high error rates, such as NASA’s interplanetary communications (Burleigh 2007), applications for rural areas, and disaster zones. It utilizes the Bundle Protocol (BP) (Scott and Burleigh 2007) to store and forward messages, called bundles, across the network.

Conversely, NDN represents a shift from traditional host-centric (i.e., DTN) to data-centric networking. NDN focuses on retrieving data by names rather than by host IP addresses, enhancing efficiency, security, and scalability in content distribution. It leverages hierarchical naming and in-network caching to optimize data retrieval, making it suitable for applications like content distribution networks, Internet of Things (IoT), and real-time data dissemination.

In general, DTN and NDN operate on fundamentally different addressing and naming schemes which pose significant interoperability challenges. DTN uses Endpoint Identifiers (EIDs), typically in the form of Uniform Resource Identifiers (URIs) with a ”`dtn://`” prefix to identify communication endpoints. In contrast, NDN employs hierarchical names to identify data directly without a URI, focusing on the content rather than the URI/location.

Despite their individual strengths, this disparity complicates the development of middleware and applications that need to operate seamlessly across both DTN and NDN networks. Modifying the existing DTN and NDN protocols to accommodate a unified naming scheme is impractical due to standardization constraints and the widespread deployment of these protocols. Therefore, we adopt a controller-based solution scheme that operates at the middleware layer without any changes to the underlying protocols. This paper aims to address these challenges by:

- Designing a consistent neutral unified naming convention, dubbed ”`uni/...`”, to serve as a common prefix for resource identifiers used across both DTN and NDN.
- Implementing middleware controller services for mapping ”`uni/...`” to DTN EIDs and NDN names without modifying existing protocols.

Thus, the key contribution of this paper is to facilitate an efficient integration design and implementation of controller services for two widely adopted communication protocols to operate seamlessly in heterogeneous networked applications. This further simplifies application development and enhances the interoperability between two incompatible data sharing protocols.

The remainder of the paper is organized as follows. Section II provides a brief overview of the DTN and NDN naming and addressing schemes and the inherent challenge of combining them in a hybrid mode. Section III discusses the related work. Section IV describes the system design followed by the implementation details of the system prototype in Section V. Section VI presents a performance evaluation. Section VII concludes the paper and highlights the future work.

Background

In this section, we give a brief overview of each of the protocols: Delay/Disruption Tolerant Networking (DTN) and Named Data Networking (NDN). We describe the resource/-

---

data naming and addressing schemes, and the inherent challenges in unifying the two protocols.

DTN Naming and Addressing Scheme

DTN addresses the limitations of traditional networks in environments where continuous end-to-end connectivity cannot be assumed. At the core lies its Bundle Protocol (BP) (Scott and Burleigh 2007) that operates at the application layer, providing a store-and-forward message delivery scheme using `bundles` as the basic data unit. Nodes and services in DTN are identified using Endpoint Identifiers (EIDs), which are typically structured as URIs: `dtn://nodeID/scheme/service` where the `scheme` indicates the unique `node` identifier within the network, and an optional component specifying the `service` for the application endpoint of the node. EIDs are used for routing and delivering the `bundles` to their destinations. This host-centric nature of EIDs focuses on the endpoints of communication, and routing decisions are often made based on node availability and contact schedules.

In general, DTN's naming scheme is tightly coupled with its addressing, as the EIDs serve both as addresses and as identifiers for endpoints (IP address). With this addressing scheme, the routing protocols (i.e., Contact Graph Routing (CGR)) utilize knowledge of network contacts and schedules to make the forwarding decisions. To accommodate the intermittent network behavior, `store-and-forward` mechanism is employed that allows data/`bundles` to be held at intermediate nodes until a suitable forwarding node reachability opportunity arises. A light-weight bundle implementation is presented in (Schildt et al. 2011), and an extensive review of the protocol can be referred to in the recent survey paper in (Koukis, Safouri, and Tsaoussidis 2024).

NDN Naming and Addressing Scheme

NDN takes a data-centric approach for its naming and addressing scheme. It uses hierarchical names to identify data, structured as sequences of name components separated by slashes `/domain/category/resource` where a domain is a top-level identifier, such as an organization or service which falls under a group of related data or services in a given category, and the specific data item or content is the resources exchanged/shared. Therefore, the primary focus of NDN is on the content being requested rather than the location of the content shared a priori.

In general, names in NDN are location-independent and are used for routing, caching, and security. With this, the communication is based on two types of packets: `Interest Packets` which are sent by information/data consumers to request data by name, and `Data Packets` which are returned by information/data producers or intermediate nodes forwarding in response to Interest/subscriptions containing the requested data. Unlike DTN, NDN routers forward Interest packets based on the name prefixes and maintain a `Pending Interest Table (PIT))` to keep track of Interests awaiting data. Similarly, the data packets are cached at intermediate nodes, enabling efficient data retrieval and reducing redundant data transmissions. The details of the protocol are referred to in (Afanasyev et al. 2018) and (Saxena et al. 2016).

Protocol Unifications Challenges

The fundamental core differences between the two protocols can be loosely grouped into *Semantic and Structural* differences.

Semantic Differences As highlighted in the previous two subsections, the protocols' semantic attributes fall into two categories:

- Host-Centric vs. Data-Centric: DTN's EIDs are host-centric, focusing on endpoints, whereas NDN's names are data-centric, focusing on content.
- Addressing vs. Naming: In DTN, EIDs serve both as addresses and names, while NDN separates the concept of addresses (unnecessary) and data names.

Structural Differences Similarly, the structural differences of the two protocols fall into two categories:

- URI Schemes vs. Hierarchical Names: DTN uses URIs with a prefix `dtn://`, whereas NDN uses hierarchical names rather than URI end-points.
- Routing Mechanisms: DTN relies on store-and-forward mechanisms with knowledge of contact schedules for node reachability, while NDN uses name-based forwarding with in-network caching for efficiency.

The inherent challenges in unifying the two protocols are due to the incompatibility of their naming conventions. On one hand, the lack of a consistent naming scheme complicates the development of applications and middleware for applications to seamlessly exchange data on a heterogeneous networked applications. On the other hand, protocol modification constraints for integration are impractical due to standardization and widespread deployment. Therefore, a novel design scheme is critical to address these differences in a practical approach, thus, the focus of this paper.

Related Work

Delay/Disruption-Tolerant Networking (DTN) (Burleigh et al. 2003) was formalized by the IRTF to ensure eventual delivery over contacts that may be separated by minutes to days as defined in RFC 4838 (Cerf et al. 2007). On the other hand, Named Data Networking (NDN) (Zhang et al. 2014) has emerged from the Information-Centric Networks (ICN) community to enable receiver-driven, content-centric retrieval with in-network caching (Team 2025). The stark differences are naming, forwarding and reliability semantics for these two paradigms that have spurred several hybrid efforts to adopt the best of both worlds for improved messaging applications.

Authors in (Liu and Fujita 2024) use Disruption-Tolerant Networking Implementations of the Bundle Protocol 7 (DTN7) at the network layer, but use brokers to periodically broadcast advertisement packets to maintain communication paths and facilitate efficient data forwarding, drawing inspiration from Named Data Networking (NDN) techniques. (Liu and Fujita 2025) conducts extensive simulations using `ns-3` showing a significant reduction in message delivery delays while improving delivery rates for high-mobility scenarios. These works are able to improve network performance metrics for DTN by attempting to leverage the key design principles and characteristics of NDN. However, they do not actually allow for effective cross-protocol communication or hybrid networks.

Adaptive hybrid schemes have been demonstrated in several studies. Authors in (Tyson, Bigham, and Bodanese 2013) introduced an Information-Centric DTN (ICDTN)– a conceptual model that grafts ICN naming onto DTN routing to exploit cache diversity in disrupted environments. An NDN approach for DTN Routing introduced in (Li et al. 2017) provides an extensive review of DTN integration with ICN, justifies hybrid protocol adoption and highlights the performance advantage of name-based routing in DTN environments.

Authors in (Sarros, Demiroglou, and Tsaoussidis 2021) propose a hybrid DTN and NDN to improve data retrieval from intermittently-connected devices for IoT and sensor networks applications in remote areas. In (Demiroglou, Mamatas, and Tsaoussidis 2023), the authors extended this work for their SDN-controlled smart-city platform. They introduced an NDN-over-DTN (NoD) tunneling stack that encapsulates NDN packets in BPv7 bundles that relies on DTN custody transfer for reliability and dynamically switching between NDN, DTN, and NoD based on link quality. While effective for basic file sharing, it retains DTN's hop-by-hop semantics and lacks a mechanism for demultiplexing multiple concurrent content sessions.

Furthermore, their approach is unidirectional (NDN→DTN). It does not preserve NDN security metadata end-to-end. Along the same lines, trust-aware extensions in doctoral theses (Sarros 2022) and (Demiroglou 2024) implement multi-protocol systems in a variety of single hop and multi hop environments with dynamic networking conditions and extend NDN with DTN transport and a reputation-based trust layer but require modified DTN nodes.

Unlike most of these studies, our hybrid approach advances the state of the art by (i) offering bi-directional translation, (ii) maintaining a per-session state so that applications experience a continuous dialog despite underlying disruptions, and (iii) employing a policy engine that selects NDN or DTN on a per-request basis using real-time link telemetry—an area that has not been sufficiently addressed in existing hybrid schemes.

System Design

This section describes the proposed system design scheme and the rationale behind our design choices followed by the system prototype implementation details and experimental evaluations in the next sections respectively.

Design Principles

To bridge the gap between DTN and NDN without modifying existing protocols, we propose a unified addressing scheme based on the following principles:

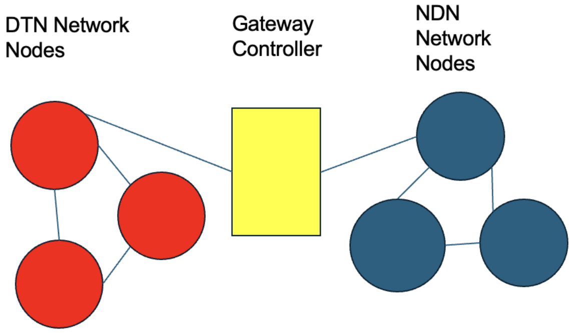

Figure 1: High-Level System Architecture. The Hybrid Middleware Gateway Controller Components Bridging DTN nodes (left) and NDN nodes (right).

1. Adopt a protocol-agnostic neutral naming scheme, referred to as `uni`, without including a scheme prefix.
2. Adopt a hierarchical naming structure compatible with NDN and representable within DTN EIDs.
3. Enable the middleware to map unified names to DTN EIDs and NDN names to transparently handle the differences.
4. Ensure that DTN and NDN protocol specifications and node identification methodologies remain unaltered.
5. Design the scheme to scale with network size and minimize performance overhead.

Unified Naming Convention

Our unified name format follows a hierarchical structure without a scheme prefix: `uni/domain/category/resource` where the `uni` prefix is the logical indicator of the unified naming convention, not included in the actual name used in protocols. The `domain` represents a high-level organizational unit that is grouped as related services or data types specified in a `category` that specifies the particular data item or service for a given `resource`. Unlike the prefix naming schemes in DTN, the `uni` prefix is a conceptual identifier used in the middleware to distinguish unified names but is not part of the name sent over the network.

Name and Address Mapping

Mapping DTN EIDs To map the unified name to a DTN EID, we adopt a scheme addition approach where we prepend the `dtn://` EID scheme to the unified name `uni` to form the path component of the EID. For example, a Unified Name of `uni/sensors/temperature/room101` mapped to DTN EID would be `dtn://sensors/temperature/room101`

Mapping NDN Names To map the unified name `uni` to an NDN name, we adopt a hierarchical name directly as the NDN name, starting with a slash. For example, the unified name of `uni/sensors/temperature/room101` mapped to NDN Name would be `sensors/temperature/room101`

Interoperability Considerations To address the unified naming and mapping issues without altering the underlying protocols, we consider the following capabilities implemented as micro-services in the middleware (discussed in the next section) introduced in (Langrin and Blackstone 2025):

- Name Resolution: Translates unified names to DTN EIDs and NDN names.
- Protocol Interface: Interfaces with DTN and NDN protocols using their standard addressing and naming schemes.
- Data Forwarding: Forwards data between DTN and NDN networks when necessary.
- Compatibility Assurance: The DTN nodes receive standard `dtn://` EIDs and NDN nodes use hierarchical names as usual, unaffected by the unified naming convention.
- Security Considerations: The middleware enforces authentication for name resolution requests, and enables data integrity to insure that the mapping process does not introduce vulnerabilities.

System Implementation

To illustrate the practicality of our proposed solution scheme, we implement the services as a middleware gateway controller component that interfaces with NDN & DTN network nodes via its respective APIs, thereby, acting as a convergence layer between the two application paradigms. We adopt the middleware introduced in (Langrin and Blackstone 2025). The core components of the middleware are a *Session Manager, Protocol Translator, and Data Synchronizer*. Each of these components are implemented as python functions as-a-service packaged deployed in docker container images for ease of integration and deployment. In addition to this, these functions are also implemented in a Jupyter Notebook (Jupyter 2025) for large scale testing.

In this work, we integrate the middleware with a protocol-specific formatting micro-services where the middleware formats the unified name according to the requirements of DTN and NDN protocols. This enables no changes to protocols where the existing DTN and NDN nodes receive addresses and names in their standard formats. Figure 1 illustrates high-level architecture of the middleware and the naming and address mapping micro-services.

Building on the unified addressing scheme following the aforementioned design principles, the unified naming scheme is designed with the following requirements:

- Consistency: Provide a consistent naming convention across both DTN and NDN.
- Protocol Agnosticism: Use a neutral naming convention without protocol-specific schemes.
- Hierarchical Structure: Maintain compatibility with NDN's naming conventions and DTN's EID structures.
- Middleware Implementation: Handle all mappings within the middleware layer, requiring no changes to protocols.
- Ease of Integration: Simplify application development by providing a uniform naming approach.

To illustrate this, we consider an Environmental Monitoring application where the Unified Name is `uni/env/sensors/humidity/greenhouse`. The DTN EID Mapping would be `dtn://env/sensors/humidity/greenhouse`, and the NDN Name Mapping would be `/env/sensors/humidity/greenhouse`. Similarly, for an Educational Content Distribution, a Unified Name `uni/edu/courses/cs101/lecture`. The DTN EID Mapping would be `dtn://edu/courses/cs101/lecture2` and the NDN Name Mapping would be `/edu/courses/cs101/lecture2`.

Advantages of the Unified Naming Scheme

Key advantages include:

- Simplified application development where the developers can use the unified naming convention without worrying about protocol-specific details, reducing complexity and potential errors.
- Seamless interoperability where the applications and services can operate across DTN and NDN networks, with the middleware handling the necessary translations.
- Scalability in which the hierarchical structure supports efficient organization and retrieval of resources, scaling effectively with network growth.

By implementing the solution entirely at the middleware layer, we avoid the need for any changes to existing DTN and NDN protocols.

Middleware Building Blocks

The middleware serves as an intermediary between applications and the underlying network protocols. Key components include:

- Name Resolution Service: Handles requests to map unified names to DTN EIDs and NDN names.
- Caching Mechanism: Stores frequently used mappings to improve performance.
- Protocol Interface Modules: Communicates with DTN and NDN networks using standard protocols.

Process Flow An application uses a unified name (e.g., uni/edu/courses/cs101/lecture2) to request or publish data. The middleware determines the target protocol based on network conditions or application preferences, and maps the unified name to the appropriate address format without modifying protocols. For DTN communication, the mapped EID (dtn://edu/courses/cs101/lecture2) is sent/receives bundles mediated by the middleware controller gateway. On the other hand, for NDN communication, the middleware uses the mapped name (/edu/courses/cs101/lecture2) to send or receive interest and data packets. Finally, the middleware delivers responses back to the application, maintaining the unified naming abstraction.

With this, we achieve seamless interoperability. From the DTN protocol prospective, the middleware presents standard EIDs to DTN nodes, ensuring compatibility while the middleware and uses standard NDN names, thereby, requiring no changes to NDN specifications. As a result, all necessary translations and mappings are confined to the middleware layer. Most importantly, we implemented access control for the mapping services using checksums and digital signatures to ensure that data integrity during mapping and transmission.

Performance Evaluations

We consider assessing how different network conditions affect the performance of the middleware. Specifically, we evaluated the following communication patterns of 1) within DTN and NDN independently, and 2) cross-network communication between DTN and NDN. For each pattern, we will measure latency to capture the time taken for content retrievals and the success rate for the percentage of successful content retrievals. We also measure the throughout for all communication initiated by NDN and DTN.

Experimental Setup

We developed a network topology of 50 nodes with 25 DTN nodes and 25 NDN nodes using Jupyter Notebook (Jupyter 2025). The DTN nodes consist of 10 source nodes which generate content in a DTN format, 10 relay nodes which forward bundles in the DTN network and 5 gateway nodes which connect to the NDN network through the middleware.

The NDN nodes consist of 10 consumer nodes which request content in an NDN format, 10 producer nodes which provide content in an NDN format and 5 Gateway nodes which connect to the DTN network through the middleware to test the middleware with one-to-many and many-to-one data dissemination. Specifically, we evaluated the following communication patterns: Communication within the DTN network (DTN to DTN), within the NDN network (NDN to NDN), and communications across networks from DTN to NDN and vice versa. We ran a simulation with a 5% failure rate to evaluate the performance of the middleware in the presence of network failures.

For DTN, we simulated the Bundle Protocol 7 (BP7)–the core protocol for DTN communications that serves the convergence layer adapters for a bundle security protocol for authentication and encryption, TCP, UDP, and other transport and routing protocols (Liu and Fujita 2025). Our DTN bundle security protocol includes support for Bundle Authentication Blocks (BABs) and Payload Confidentiality Blocks (PCBs).

For NDN, we simulated the NDN Forwarding Daemon (NFD) which is the core NDN forwarder, a content store for caching NDN data packets, a Pending Interest Table (PIT) for tracking outstanding interests, a Forwarding Information Base (FIB) for routing NDN interests and NDN TLV encoding for complete type-length-value encoding for efficient packet representation.

Finally, for the middleware we simulated an address resolver for translating between DTN and NDN addressing schemes, protocol translators for converting between bundle and interest/data formats, a content manager for storing and retrieving content across networks and gateway services for connecting DTN and NDN networks. We utilize serialization for accurate packet size calculation for network planning and optimization and the unified addressing scheme described in section IV for seamless conversion between DTN EIDs and NDN hierarchical names. We were able to simulate all of these components in Jupyter Notebook using Python classes.

Experimental Results

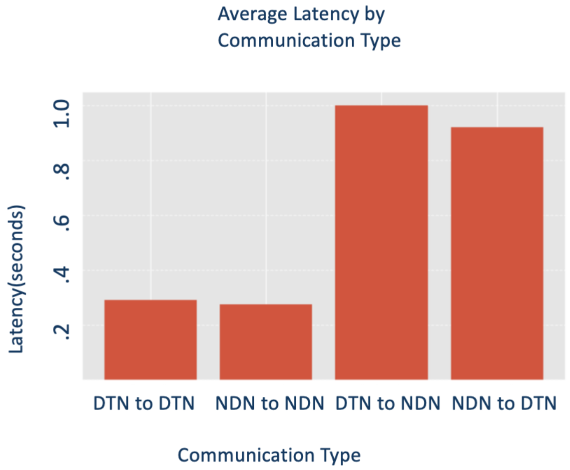

Figure 2: Average latency for same-network and cross-network communications

Figure 2 shows the average latency for each of the same-network and cross-network communication types. The x-axis represents the different communication types and the y-axis represents the latency for packets sent from source to destination on the network using a particular communication type. We find that the same-network communications(DTN to DTN and NDN to NDN) both have an average latency around .3 seconds while the cross-network communications (DTN to NDN and NDN to DTN) have an average latency closer to 1 second. This is because cross-network communications have processing overhead from three main factors.

First, we note that cross-network communication requires protocol translation overhead to convert between DTN bundles and NDN packets. This involves address translation from a DTN EID to an NDN name or vice versa as well as packet format conversion and metadata translation to and from NDN or DTN. Second, we note that cross-network communication introduces the possibility of a bottleneck on the middleware gateway nodes because all cross-network traffic must pass through them. Finally, we note that more complex routing decisions must be made when crossing network boundaries.

Alternatively, same-network communications require no protocol translation. Directing routing can be done within the same network topology and there are optimized routing algorithms for homogenous networks.

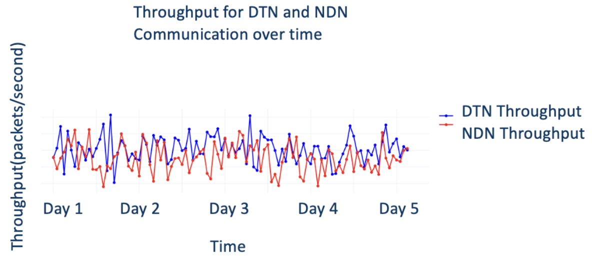

Figure 3: Throughput over time for communication initiated by DTN and NDN

Figure 3 shows how throughout changes over time for communication initiated by NDN and DTN. The x-axis represents timestamps for the network activity and the y-axis shows the throughput of packets per second. We find that NDN has a throughput of 45 packets for second with a standard deviation of 8 packets per second and DTN has a throughput of 50 packets per second with a standard deviation of 10 packets per second making the average overall throughput 47.6 packets per second.

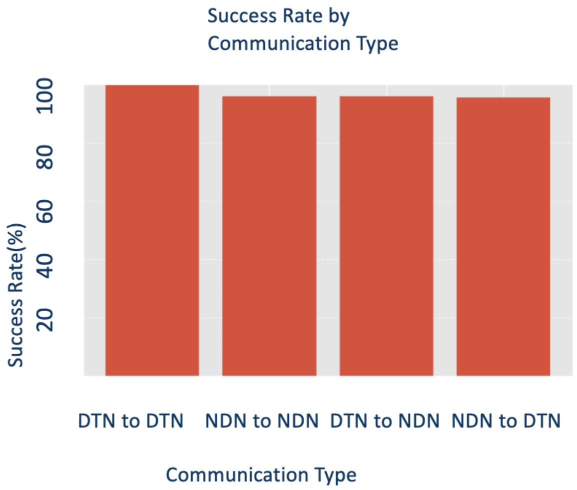

Figure 4: success rate for each of the communication types

Figure 4 shows the success rate for each of the communication types. The x-axis represents the different communication types and the y-axis represents the success rate for packets sent from source to destination on the network using a particular communication type. We find that DTN to DTN communications have a 100% success rate while both same-network and cross-network communications that involve NDN all have success rates around 96%. This is significant because it demonstrates that the effects of the unified naming scheme and middleware on the success rate are negligible.

Conclusion

We presented a consistent, unified, naming and addressing scheme for hybrid DTN/NDN communication protocols to seamlessly exchange information without modifying the underlying protocols. We discussed the details of our design and a system prototype implementation integrated into a middleware controller to illustrate the practicality of our hybrid scheme. Finally, we presented preliminary performance evaluations to show the efficacy of our solution approach. For future work, we consider extending our naming scheme to support additional networking paradigms (i.e., ZMQ/MQTT) and conduct extensive simulations and practical application use cases to evaluate performance under various network conditions.

Acknowledgments

This material is based on work supported by the Air Force Research Laboratory (AFRL) under the Air Force Contract No. FA955023D0001. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the AFRL.

DISTRIBUTION STATEMENT A: Approved for Public Release, Distribution unlimited: PA CLEARANCE: AFRL-2026-0396

References

Afanasyev, A.; Burke, J.; Refaei, T.; Wang, L.; Zhang, B.; and Zhang, L. 2018. A brief introduction to named data networking. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, 1–6. IEEE.

Burleigh, S. 2007. Interplanetary overlay network: An implementation of the dtn bundle protocol.

Burleigh, S.; Hooke, A.; Torgerson, L.; Fall, K.; Cerf, V.; Durst, B.; Scott, K.; and Weiss, H. 2003. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6): 128–136.

Cerf, V.; Burleigh, S.; Hooke, A.; Torgerson, L.; Durst, R.; Scott, K.; Fall, K.; and Weiss, H. 2007. RFC 4838: Delay-tolerant networking architecture.

Demiroglou, V. 2024. *NDN-and DTN-based solutions for intermittent IoT communications*. Ph.D. thesis, Δημοκρίτειο Πανεπιστήμιο Θράκης (ΔΠΘ). Σχολή Πολυτεχνική. Τμήμα Ηλεκτρολόγων ....

Demiroglou, V.; Mamatas, L.; and Tsaoussidis, V. 2023. Adaptive NDN, DTN and NoD deployment in smart-city networks using SDN. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 1092–1097. IEEE.

Jupyter, P. 2025. Jupyter Notebook. Technical report.

Koukis, G.; Safouri, K.; and Tsaoussidis, V. 2024. All about Delay-Tolerant Networking (DTN) contributions to future internet. *Future Internet*, 16(4): 129.

Langrin, R.; and Blackstone, J. 2025. A Hybrid DTN/NDN Middleware Gateway Controller for UAV Applications. In *MILCOM 2025-2025 IEEE Military Communications Conference (MILCOM)*, 1590–1595. IEEE.

Li, Y.; Li, Y.; Wolf, L.; Lingren, A.; and Wang, J. 2017. A named data approach for DTN routing. In *2017 wireless days*, 163–166. IEEE.

Liu, X.; and Fujita, S. 2024. Resilient Pub/Sub Multicast in Dynamic MANETs with High Node Mobility. In *2024 Twelfth International Symposium on Computing and Networking Workshops (CANDARW)*, 8–14. IEEE.

Liu, X.; and Fujita, S. 2025. Reliable Low-Latency Multicasting in MANET: A DTN7-Driven Pub/Sub Framework Optimizing Delivery Rate and Throughput. *Information*, 16(6): 508.

Sarros, C.-A. 2022. Extending named data networking with delay-/disruption-tolerant and reputation-based trust mechanisms.

Sarros, C.-A.; Demiroglou, V.; and Tsaoussidis, V. 2021. Intermittently-connected IoT devices: Experiments with an NDN-DTN architecture. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 1–9. IEEE.

Saxena, D.; Raychoudhury, V.; Suri, N.; Becker, C.; and Cao, J. 2016. Named data networking: a survey. *Computer Science Review*, 19: 15–55.

Schildt, S.; Morgenroth, J.; Pöttner, W.-B.; and Wolf, L. 2011. IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation. *Electronic Communications of the EASST*, 37.

Scott, K.; and Burleigh, S. 2007. RFC 5050: Bundle protocol specification.

Team, W. C. D. 2025. WC3 Group. Information Centric Networking. Technical report.

Tyson, G.; Bigham, J.; and Bodanese, E. 2013. Towards an information-centric delay-tolerant network. In *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 387–392. IEEE.

Zhang, L.; Afanasyev, A.; Burke, J.; Jacobson, V.; Claffy, K.; Crowley, P.; Papadopoulos, C.; Wang, L.; and Zhang, B. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3): 66–73.