NDN in Large Detached Underwater Sensing Arrays

Craig Partridge, Samuel Nelson, Vladimir Shurbanov, Dorene Ryder and Bishal Thapa
Raytheon BBN Technologies
Cambridge, MA

***Abstract*—Large underwater sensing arrays use acoustic communications links over long distances (several km) to coordinate the activities of multiple sensors. Acoustic links have low bandwidths and long propagation delays, making them a hostile environment for most communications protocols. In this paper we explore how NDN performs in this challenging environment.**

Keywords—underwater communication; content distribution networks; sensor systems

I. INTRODUCTION (*HEADING 1*)

Underwater sensing arrays typically employ regularly spaced sensors to detect and assess the underwater environment. Detached arrays lack regular communications bandwidth outside the array – in the case of underwater arrays; this means the arrays lack regular contact with entities on shore. Large detached arrays (covering thousands of km^2) also lack high quality communications internally – they must rely on low bitrate, long delay, acoustic links. These attributes tend to highlight the worst features of communication protocols. Yet the very purpose of these arrays requires communication: it is usually only possible to reliably detect information of interest by comparing results from multiple sensors.

The goal of the work we describe in this paper was to do an initial assessment of how well NDN might meet the communications needs of a large detached underwater sensing array. We created a plausible design for such a network and then evaluated operating it using NDN in simulation. .

We want to emphasize this is not a comprehensive analysis. There are too many choices about how to deploy arrays and what kinds of applications run on arrays and choices for communications both within the array and intermittently outside the array. Rather this is a walk through the design space, looking for information that may guide future experiments and designs. In this respect, this paper is a throwback – akin to the exploratory wireless networking papers of the 1970s and early 1980s.

II. EXEMPLAR SENSING ARRAY

In this section we describe our model sensor network and explain the choices that led to this architecture as being the one over which we tested with NDN and various MAC protocols.

A. Sensing Array

We envisioned a deep-water acoustic sensing array. Applications in the array seek to understand and track acoustic signatures. The array is far enough out to sea to avoid the poor propagation issues that can bedevil acoustic communications near shorelines.

The sensing array is composed of individual sensing nodes, which remain afloat at roughly the same place under the surface for an extended period. The nodes are positioned 15 km apart. We experimented with 5x5 grids, able to sense an area of 8100 km^2.

Communication within the sensing array is via acoustic communications channels. A node transmitting on the acoustic channel is incapable of hearing another transmission. Multiple audible transmissions at the same receiver cause a collision. Signals are audible at more than 15km but less than 30km, a transmitting node's neighbors all hear a transmission but its neighbors' neighbors do not. The hidden terminal problem is constantly present.

Data rates were varied in our experiments, from 0.5Kbps to 4Kbps. Bits move at 1,500 m/s, so the propagation delay between two sensors is 10s and anywhere from 5,000 to 40,000 bits are in flight between any two sensors depending on data rate.

This pattern and the communications rates are designed to be plausible ones for a sensing array that is, for instance, tracking a sea lane. Sensors are placed relatively far apart such that, in general, each sensor covers its own region – there's very little overlap. This pattern actually helps many sensing applications – something that can be heard concurrently at multiple sensors is relatively loud, which usually means it is not of interest. The bit rates are consistent with running an acoustic link over such distances.

We used the NS-3 UAN model to simulate the channel and propagation characteristics for an underwater sensor network. The channel, propagation, energy and mobility models used in the UAN model are derived from [6], [7]. We mostly used the stock implementation for our simulation runs other than minor extensions that were necessary to make it work with NDN's Forwarding Information Base (FIB) setup mechanism. We used ndnSIM as the NDN Simulator for NS3 [1].

The NS-3 UAN module is not perfect in terms of emulating the real-world underwater channel and the network environment. But it takes a strong step in the direction of trying to offer a reliable and realistic tool. The UAN module offers accurate modeling of the acoustic channel, a model of the WHOI acoustic model (one of the widely used acoustic modems) [8] and its communication performance.

The sensing network is optionally enhanced with a mobile device. The notion is that if the sensor array detects a faint signal of interest, it may move the mobile device to a better location to sense the signal.

Including a mobile device in the example network is consistent with the increased use of Unmanned Underwater

978-1-5090-2482-7/16/\$31.00

Vehicles (UUVs) and activities such as the DARPA Upward Falling Payload program[13], that seek to make it easier to deploy and utilize UUVs. It also adds an important complication to the communications environment, namely converting the static structure of the sensor array to one in which there is at least one dynamic node. The mobile device moves at 3600 m/hour. It is modeled as moving along a path that is 15km from the edge farthest from the sensor node connected to a communications buoy (see next paragraph). The mobile device communicates using the same acoustic protocol as the sensors and communicates with the node in the sensor network that is closest to the mobile device and switches nodes as distances change.

The mobile device also provides an occasional link to shore. If running fully autonomously there is the worry that the mobile device could get into trouble (e.g. accidentally bump into the thing it is tracking), and so some level of human operator control is desirable. So we assumed that when the mobile device is active, some form of link to a human being on shore was deployed. Our canonical example was to assume the sensing array launched a buoy, attached by cable to a sensor in the array, and having a satellite terminal in the buoy. This idea is the modern equivalent of the rescue buoys with phone lines used in the 1920s on submarines. An alternative would be to imagine a temporary undersea link, e.g. the dynamic fiber optic links that the DARPA Tactical Undersea Network Architectures (TUNA) program seeks to develop[14]. The mobile device is deemed under effective control if a control loop is closed between shore and the mobile device at least once every 600s.

We placed the node connected to the buoy at one of the corners of the sensor array. The logic was that in a military sensor array, one might want to avoid having a surface buoy marking the middle of a sensor array – better to mark an edge, and leave open what direction the array extended. Also, if we were to replace the buoy with a dynamic fiber link, having the connecting node closer to shore would save up to 90 km of fiber (e.g. along the long diagonal of the grid). The data rate is 1 Mbps and the propagation delay is set to 0. The reason behind this assumption is that neither the tail circuit nor the satellite link comes anywhere close to being the bottleneck in terms of the data rate and propagation delay under our simulation scenarios.

B. *Application Traffic*

The network was modeled as always running a sensing application and optionally running an application to remotely control the mobile device.

For the sensing application, all sensor nodes generate a 7 byte report (not including protocol headers, this is just the data size) approximately every 10 seconds. These reports are sent to a single node that compares all the reports and determines if there are signals of interest. Currently this consolidator node does nothing except receive the reports – that is, it does not change its behavior if a signal of interest is found.

In the control application, a human operator sends the mobile device an 18-byte command over the tail circuit via the surface buoy. Commands are transmitted or retransmitted every 100s, 300s or 600s, depending on the simulation. The mobile device replies to a command with a maximum size message (as big as the medium permits – which varies by MAC – a topic discussed in our simulations below). If a command has not been acknowledged, it is retransmitted at the end of an interval. A control loop is considered closed if a command receives a response from the device within 600s.

III. PRIOR WORK ON NDN AND SENSORS

Named-data Networking (NDN) is an networking protocol suite in which communication is achieved by requesting idempotent named pieces of content. NDN uses content names and in-network caching to make routing and distribution decisions. NDN is receiver-driven. Receivers transmit Interest packets which are forwarded to the appropriate content source or cache, resulting in Data packets flowing back to the receivers.

Previous work related to NDN for other sensor networks include NDN for Internet of Things [2], NDN for Ad hoc wireless sensor network (WSN) [3], and NDN for real time wireless recharging framework for WSN [4]. Given the differences between radio frequency and acoustic communication networks, this work doesn't provide much guidance.

The closest work to the work presented here is [5], which assesses NDN's ability to retrieve Data in a small (12 node) highly loaded underwater sensor network running a MAC similar to the Aloha MAC below. Our results are consistent with theirs if the data rate is assumed to be 4Kbps or higher.

IV. MEDIA ACCESS LAYER

NDN expects to run over a media access layer (MAC). MAC layers for underwater acoustic networks is a well-studied field [11]. After consulting with some colleagues who have worked in the area we developed and evaluated three MACs.

Before discussing the MACs, it is useful to provide a little intuition about why underwater acoustic MACs are difficult to design and what dimensions of MAC design we sought to cover with the MACs.

A. *Challenges*

The physics of acoustic links currently limit devices to sending or receiving at any given time. This constraint puts limits on any MAC layer.

The basic problem is that if node Q is transmitting to node R, then node R's neighbors may not transmit, lest they cause a collision at R. Note too that if the destination of Q's transmission is not known, then *all* the neighbors of Q's neighbors need to refrain from transmitting to avoid interference.

It would be useful, therefore, if there was a mechanism by which transmissions and receptions were coordinated, but that leads to the other major constraint: propagation delay.

The MAC layer can be improved if nodes coordinate their transmissions. But the propagation delay between nodes is very large: 10s or up to 40,000 bits. Carrier sense protocols (a simple way to share state) would require a 10s guard time!

Bidding for transmission times would be time-consuming and, logically, wasteful as the bid messages would be similar in size to the data messages – rather than bid, a better approach is almost assuredly to send the data. It was with these challenges in mind that we picked the sample MACs.

B. *ALOHA MAC*

We simulated an Aloha MAC with carrier sensing, in which, given a unit of data to send, if the node is not currently receiving and not currently transmitting, the node simply transmits. There are no acknowledgements. Collisions at the receiver result in data being silently lost.

The goal of the Aloha MAC was to encourage concurrent transmissions in the network, at the risk of having collisions. Note that we were running Aloha in a network with more hops than past studies recommended [11]. We expected the lower traffic loads farther from the consolidator node would balance the limitations.

Aloha is simulated with the stock NS-3 Aloha MAC enhanced to support queuing. (The stock version keeps no queue and discards new data if the node is either receiving or transmitting). We also had to fix some bugs (contact the authors for the list).

As a quick check that CSMA would do poorly and our simulator ran as expected, we ran a few test runs with the stock NS-3 CSMA MAC and it did indeed substantially underperform compared to the Aloha MAC.

C. *Time Division Multiplexing – Round Robin (TDM-RR)*

We leveraged TDMA implementation from [9] as a starting point to implement two different time division multiplex schemes for our simulation runs. The first scheme is a round robin scheme where the scheduler schedules all nodes in the network to have a dedicated slot in a round robin fashion. So, the overall TDM frame size, which is also the overall delay for any packet traversal between the first nodes in the assignment to the last, is always a slot time X number of nodes in the network. TDM-RR achieves maximum protection against collisions but no concurrency.

D. *Time Division Multiplexing – Interference Alignment (TDM-IA)*

The second TDM MAC that we implemented and evaluated via simulation is called TDM - Interference Alignment (TDM - IA).

TDM-IA takes hidden terminal and routing information into consideration and optimizes the medium access schedule by: (1) allowing concurrent transmissions while avoiding hidden terminal effects at any of the intended receiver nodes; (2) allowing hidden-terminal interference at the non-intended receiver nodes; (3) eliminating guard time by lining-up propagation delay to be an integer multiple of the time slot size, and; (4) setting the total frame size to the max in-degree of any node in the network plus one. This is the optimal frame size [10].

TDMA-IA achieves concurrency and some channel management but does not yield optimal or the most efficient TDM schedules. Note that we assume that a central scheduler exists that knows the routing table and transmission schedule of all nodes needed to create the TDM-IA schedule, and that the propagation delays are perfect integer multiples of the time slot size. These assumptions could be relaxed in future work – for instance, a distributed schedule that loosens the complete knowledge assumption is described in [12]. Figure 1 depicts a TDM-IA schedule for a 5x5 grid.

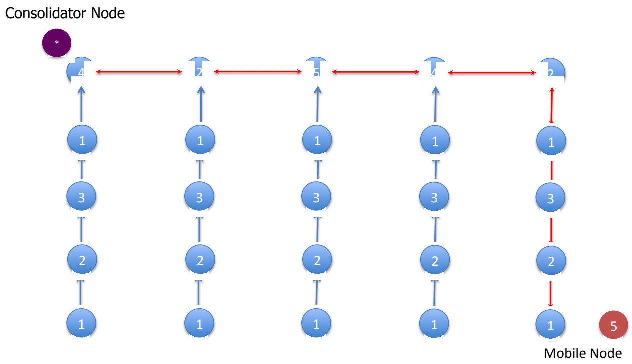

Figure 1: TDM-IA schedule with node label representing the slot assigned to the node in the grid and arrows representing the direction of the traffic flow

V. SIMULATIONS

We ran hundreds of simulations, generating thousands of data point. This section briefly presents two results on routing and header sizes that are, we believe, unsurprising but necessary to understanding the other simulations. Then the section focuses on more interesting results.

A. *Routing*

Given the presence of the mobile node and the intermittent link to shore, our initial expectation was the network might need to run a routing protocol to setup FIBs and that the overhead of routing would be an issue. We ran each of simulation setups described below with the Optimized Link State Routing (OLSR) protocol and compared that with static routing, where the set of static routes include routes for the mobile device traffic. The static routes used the shortest paths from any source to the destination.

We independently simulated the impact of routing on our NDN experiments without integrating OLSR with the ndnSIM. We did not try any other routing protocols. Running our experiments with NDN's Neighborhood Link-State Table and Hyperbolic routing protocols is in our future work [1].

OLSR overloaded the network. The applications struggled to get any useful traffic exchanged. We could have tried to optimize OLSR for this environment. For instance, we could have sharply increased various time intervals and sought to shrink routing message sizes. We chose not to do so because our back of the envelope calculations suggested these improvements might not be enough. Our assessment was that part of the problem was OLSR's need to exchange state with all nodes, which created traffic loads on all links that was hard to manage. Furthermore, we expect much of the sensor grid topology to remain static. Only the position of the mobile node will change and at a very slow pace. Letting the mobile node broadcast its position and update its link state information with one-hop neighbors seemed sufficient to maintaining up-to-date routes. Hence, we use static routes to run all the experiments below.

B. Protocol Headers

Recall that application data is generally small. Sensing messages are just 7 bytes. So protocol header sizes could heavily influence traffic load. When MAC headers were added to NDN headers, the network overhead went up by 38 bytes. That represented overhead exceeding 500% for sensing messages.

We ran some tests to see how burdensome the header overhead was. The answer was crippling. We were unable to close control loops to the mobile device – in many simulations, the control loop never completed even once.

So we posited the existence of a protocol header reduction technique that might eliminate virtually all the MAC and NDN overhead. Given the limited number of applications and the limited network topology, this assumption is ambitious but not implausible. The constrained environment lends itself to header reduction. We emulate the best use case scenario using zero length headers.

With these inputs, we reran the control simulations and all protocols successfully closed their control loops at least some of the time. All experiments below use reduced headers.

C. Format of the Charts

There are two types of simulations. *Sensing* simulations look at the challenges of retrieving sensing information for the sensing application, without the application controlling the mobile device. *Control* simulations evaluate the ability to keep a control loop working with the mobile device, while the sensing application is also running in the network.

In all the simulations, the X-axis is the data rate in bps, ranging from 500 bps to 4000 bps.

In sensing simulations, the left axis in green is the goodput bitrate across all nodes that send the sensed data to the consolidator node, with the solid line being the target goodput (all data is sent and received with no losses and no duplicates) and the dashed line being the achieved goodput. The right axis, a log scale in blue, plots the delay times between the sensing node's first transmission and when the consolidation node finally receives the data, with crosses for individual measured times and the dashes representing the median delay. Note that the time values cluster, so a single cross may represent many individual times.

In the control loop simulations, the left side in green depicts how many times a control loop was completed. The solid lines show the fraction of time the loop was closed at least once in any 600s interval. The dashed line shows the percentage of times the total number of request-response was completed within 600s. The distinction is about assessing risk – a simulation may show at least one control loop is always completed, but that the completion rate was only 60%, suggesting a longer simulation might have a failure at least once. The right side measures the median round-trip delay (dashed line) with a solid blue line showing the 600s mark.

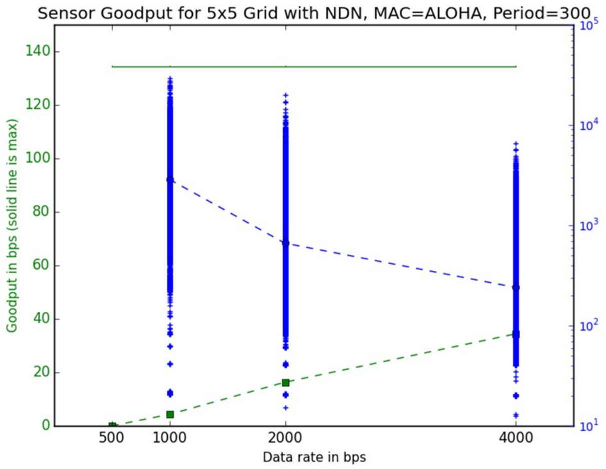

Figure 2: 5x5 Grid with NDN Sensor Traffic and Aloha MAC

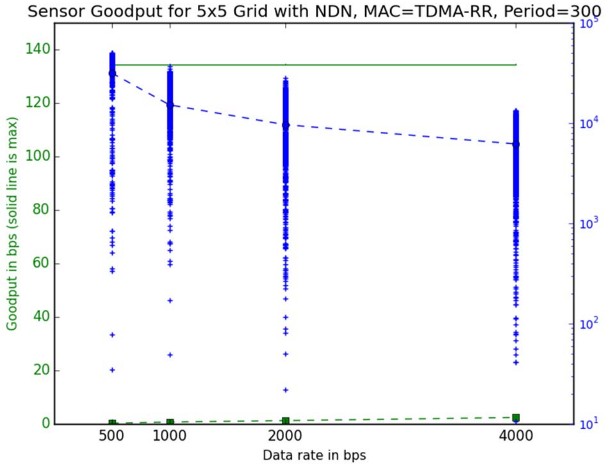

Figure 3: 5x5 Grid with NDN Sensor Traffic and TDM-RR

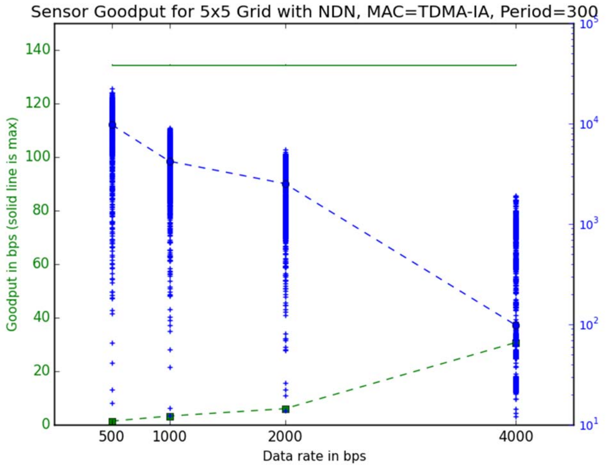

Figure 4: 5x5 Grid with NDN Sensor Traffic and TDM-IA

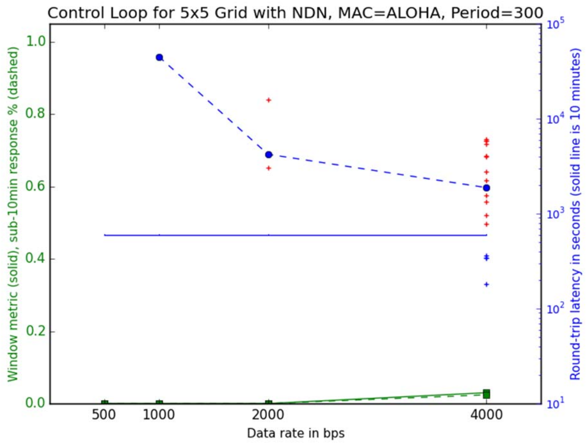

Figure 5: NDN Control Loop with Aloha MAC

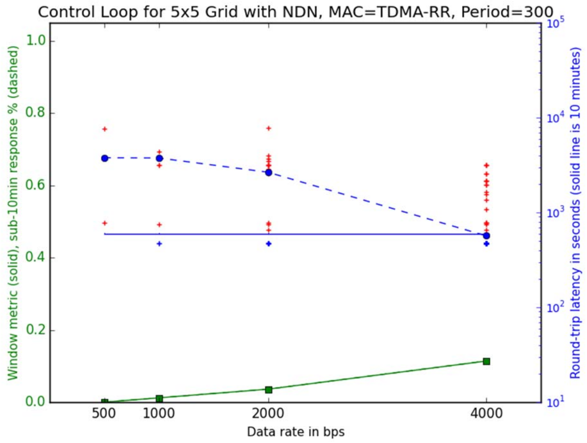

Figure 6: NDN Control Loop with TDM-RR MAC

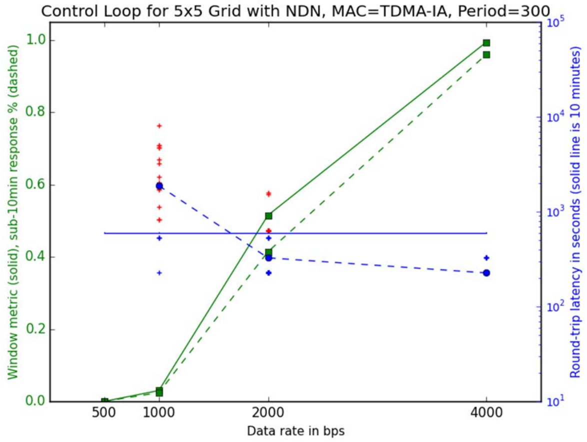

Figure 7: NDN Control Loop with TDM-IA MAC

D. Sensing Traffic (no control loop)

Figures 2-4 present the goodput and latency performance graphs for NDN network simulation when only the sensor traffic is flowing with each of the MACs. Interests are sent every 300s to all nodes (following the practice in other studies, Interests repeat for the prior Data if not received, and new Data if the past Interest has been satisified). Intuitively we would expect to see delays (right side data) varying between 10s and around 200-250s, as the longest path has a minimum delay of 160s (8 hops each way) with low goodput and TDM-RR doing sharply worse because of the elimination of concurrent transmissions.

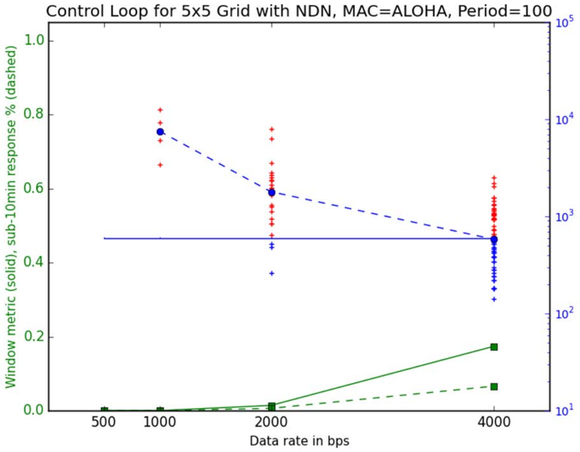

Figure 8: NDN Control Loop with Aloha MAC (Interest period = 100 sec)

This is roughly what the graphs show, with one surprise. NDN over Aloha performs similarly to NDN over TDM-IA. What is happening here is that NDN's in-network cache is making recovery from Aloha collisions sufficiently effective as to very nearly match the performance of a MAC that seeks to ensure concurrency without collisions. (Counter-intuitively, some additional simulations found that retransmitting Interests faster or slower does not change the graph much).

E. Control Loop (with background sensing traffic)

Figures 5-7 present the control loop results. Recall that the goal is to complete at least one control loop in any 600s interval. Interests were sent every 300s, allowing for 1 retransmission within the 600s interval. The control loop runs through the buoy node, which is also the consolidator node, ensuring that the control traffic and sensing traffic interact causing congestion and, for Aloha, collisions.

The overall result is that NDN performs poorly over both Aloha and TDM-RR, but performs very well over TDM-IA. Aloha's increased concurrency of transmissions does not help the several hop, two-party, control application. TDM-RR does not exploit enough parallelism in its transmission scheduling algorithm compared to the TDM-IA. However, NDN's in-network caching is still helping. Figure 8 shows NDN over Aloha but with a 100s transmission of Interests (so 5 chances to retransmit) and performance improves.

VI. CONCLUSIONS

The first observation is that to operate in this environment, NDN needs to be able to compress its headers and minimize routing traffic. Though the routing protocol is not part of the NDN architecture, a routing protocol is needed to route Interests and that overhead plays a major role in impeding the NDN operation in a constrained environment.

The more subtle observation is that NDN's in-network caching and receiver driven approach interacts well with MAC protocols that support concurrency in the network and that NDN can compensate for collisions. This result suggests that NDN allows MAC protocol designers a wider range of design options when working in the challenging environment of large underwater acoustic networks.

ACKNOWLEDGMENT

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and SPAWAR Systems Center Atlantic under Contract No. N65236-14-C-2816. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Distribution Statement "A": Approved for Public Release, Distribution Unlimited, Case 26735.

This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

REFERENCES

[1] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN Simulator for NS-3". NDN Technical Report NDN-0005, 2012. http://named-data.net/techreports.html.

[2] W. Shang, A. Bannis, T. Liang, Z. Wang, Y. Yu, A. Afanasyev, J. Thompson, J. Burke, B. Zhang, and L. Zhang. "Named Data Networking of Things". IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI), 2016. Invited Paper.

[3] M. Meisel, V. Pappas, and L. Zhang, "Ad Hoc Networking via Named Data". ACM MobiArch 2010.

[4] J. Li, C. Wang, F. Ye, and Y. Yang, "NETWRAP: An NDN Based Real Time Wireless Recharging Framework for Wireless Sensor Network". IEEE MASS 2013.

[5] M. Kuai, T. Haque, X. Hong, and Q. Yu, "A Named-Data Networking Approach to Underwater Monitoring Systems". ACM WUWNET 2015.

[6] A. F. Harris, and M. Zorzi, "Modeling the underwater acoustic channel in ns2". International Conference on Performance Evaluation Methodologies and Tools (Nantes, France, October 22 - 27, 2007). ValueTools, vol. 321. ICST (Institute for Computer Sciences Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium.

[7] O. Tremblay, L. -A. Dessaint, and A. -I. Dekkiche, "A Generic Battery Model for the Dynamic Simulation of Hybrid Electric Vehicles". Ecole de Technologie Superieure, Universite du Quebec, 2007.

[8] L. Freitag, M. Grund, I. Singh, J. Partan, P. Koski, K. Ball, and W. Hole, "The WHOI micro-modem: an acoustic communications and navigation system for multiple platforms". IEEE OCEANS05 Conference, 2005.

[9] T. Henderson. http://code.nsnam.org/tomh/ns-3-simple-wireless/.

[10] S. Gandham, M. Dawande, and R. Prakash. "Link Scheduling in Sensor Networks: Distributed Edge Coloring Revisited". IEEE INFOCOM 2005.

[11] S. Climent, A. Sanchez, J. V. Capella, N. Meratnia, and J. J. Serrano. "Underwater Acoustic Wireless Sensor Networks: Advances and Future Trends in Physical, MAC and Routing Layers". http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3926587

[12] G. Jakllari, M. Neufeld, R. Ramanathan. "A Framework for Frameless TDMA using Slot Chains," *Proc. IEEE MASS,* Las Vegas, Oct. 2012

[13] J. Krolik, "Upward Falling Payloads (UFP)", http://www.darpa.mil/program/upward-falling-payloads

[14] J. Kamp, "Tactical Undersea Network Architectures (TUNA)," http://www.darpa.mil/program/tactical-undersea-network-architectures