ICEMAN: A Practical Architecture for Situational Awareness at the Network Edge

Samuel Wood*†, James Mathewson*†, Joshua Joy‡, Mark-Oliver Stehr¶,
Minyoung Kim¶, Ashish Gehani¶, Mario Gerla‡, Hamid Sadjadpour*†, J.J. Garcia-Luna-Aceves*†
*UC Santa Cruz, †SUNS-tech, Inc., ‡UC Los Angeles, ¶SRI International
{sam, james, hamid, jj}@suns-tech.com, {jjoy, gerla}@cs.ucla.edu, {stehr, mkim, gehani}@csl.sri.com

***Abstract*—Situational awareness applications used in disaster response and tactical scenarios require efficient communication without support from a fixed infrastructure. As commercial off-the-shelf mobile phones and tablets become cheaper, they are increasingly deployed in volatile ad-hoc environments. Despite wide use, networking in an efficient and distributed way remains as an active research area, and few implementation results on mobile devices exist. In these scenarios, where users both produce and consume sensed content, the network should efficiently match content to user interests without making any fixed infrastructure assumptions. We propose the ICEMAN (Information CEntric Mobile Ad-hoc Networking) architecture which is designed to support distributed situational awareness applications in tactical scenarios. We describe the motivation, features, and implementation of our architecture and briefly summarize the performance of this novel architecture[1].**

I. Introduction

The immense global adoption of commercial off-the-shelf mobile phones and tablets has lead to inexpensive devices with sufficient performance, size, weight, and power (SWAP) characteristics for deployment at the network edge of tactical and disaster response scenarios [11], [10]. The predominate reason for deploying these devices is to support applications that increase situational awareness for the user (the warfighter or emergency responder). Increased situational awareness is paramount in scenarios where fixed infrastructure is limited to non-existent. In this case, the network must communicate opportunistically by using whatever resources are available, to provide the most recent information as early as possible to situational awareness applications. For example, a blue force tracking application provides the most recent GPS coordinates of each squad member's position, to each squad member, to avoid friendly fire. Applications must support an assortment of sensing and communication hardware to efficiently produce and consume content to increase situational awareness. Despite the increase in sensing and communication hardware capabilities in mobile devices, efficient communication to and from applications on the volatile network edge remains as a challenging research problem, and large engineering effort.[2]

[1]This work was supported in part by SRI International and by the Defense Advanced Research Projects Agency (DARPA) and SPAWAR Systems Center Pacific (SSC Pacific) under Contract N66001-12-C-4051. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

[2]It is sometimes referred to as the "last tactical mile" problem.

This paper highlights problems with applying existing network architectures to moderate sized networks running situational awareness applications at the tactical edge, and introduces a new architecture, ICEMAN (Information-CEntric Mobile Ad-hoc Networking) aimed at supporting such networks. Our approach is practical: we evaluate our implementation (which builds on Haggle [24]) on hardware with similar SWAP characteristics as those found on the tactical edge. ICEMAN adopts an Information Centric Networking (ICN) philosophy [12] where the network provides a data object publish/subscribe abstraction to applications. We use attribute-based naming, where users express queries as a set of attribute-value pairs and a matching threshold. Among other mechanisms, ICEMAN supports UDP broadcast, network coding, and utility-based content caching to increase data object delivery and reduce delivery latency. Due to the modest size of our target networks, ICEMAN pushes more intelligence into the network layer to increase performance.

The main contributions of this paper are: 1) a description of a complete ICEMAN architecture that integrates multiple content-dissemination, utility-based caching, and transport mechanisms to provide a publish/subscribe API with attribute-based content naming; 2) a description of content and context-based policies which utilize these mechanisms to achieve efficient communication at the tactical edge.

The paper is organized as follows. Section II describes related architectures, and discusses their differences in assumptions and design. Section III describes ICEMAN in detail. A brief summary of our evaluation can be found in Section IV followed by the conclusion in Section V.

II. Related Work

We summarize a few representative architectures related to ICEMAN and highlight their different assumptions and approaches.

Information centric networking (ICN) encompasses several approaches that share the same content-centric philosophy. The paradigm that distinguishes ICN from other approaches is the principle that the network should provide a host-to-content abstraction, as opposed to the traditional host-to-host abstraction. Indeed, in most ICN proposals there is not an explicit mechanism to communicate with a specific host. ICN architectures share three key design principles [12]

that are also used in ICEMAN: (i) publish/subscribe-type primitives, (ii) universal caching, and (iii) content-oriented security model. Unlike other ICN architectures, subscribers in ICEMAN specify constraints on how to match the data object to the interests. Using these constraints, each node can construct a ranked list of the best data objects that match the subscriber's interest.

As in most ICN architectures, an ICEMAN node can cache any data object that it receives, and can forward this data object to any interested node on behalf of the publisher. ICEMAN does not establish secure tunnels for host-to-host content transport nor shared group keys. It secures the content directly by cryptographically enforcing access policies using attribute-based encryption [6], which, by scoping content, plays a role mathematically dual to attribute-based naming.

CCN [15] is a well-known example of an ICN architecture based on hierarchical names and prefix matching. It uses an interest-driven paradigm with the characteristic that an interest is consumed by the first piece of matching content and needs to be refreshed for each successive piece of content to maintain a TCP-like flow-balance property. A generalization of CCN that supports push-and-pull paradigms to make it more suitable for tactical MANETs has been developed [25].

Pocket switching is similar to disruption-tolerant network (DTN), and focuses on exploiting contacts between wearable wireless devices. Initial work started at Intel Research Laboratory in Cambridge, and led to the first prototype of the Haggle architecture [13]. This line of research has been advanced in the 6th European Framework Program, which has developed a wide range of routing algorithms [23], [22], [14] that can naturally deal with mobility and exploit social relationships. The European project has led to a second generation of the Haggle architecture [24]. Due to its inherent content-based foundation, we have identified Haggle as a suitable basis for ICEMAN.

While most routing algorithms for DTNs are — like IP — based on endpoint identifiers, previous work on interest-driven routing [27], [26] in the context of the DARPA DTN program allows persistent subscriptions to content under a name that will be syntactically matched (using simple prefixes or arbitrary patterns) against content stored in the network caches. Matched content travels to the subscribers on the reverse path of the interest. DTN approaches are based on semantically meaningful units of information (content is packaged in so-called bundles, defined in RFC 5050), which has been extended to include metadata in so-called extension blocks. Despite this extension, the interface to applications is based on end-point identifiers or (hierarchical) names (with syntactic matching) and is not sufficiently general to convey the common needs of applications. Descriptive destinations [5] are a noteworthy generalization that add the capability to declaratively constrain the *scope* of destinations, but does not provide a means for the dual goal of content-based access, e.g., by declaratively expressing *interest*. Finally, the notion of single-node custody (and custody transfer) developed in the context of DTN point-to-point links does not match well with the capabilities of today's wireless networks that can utilize broadcast, opportunistically overhear, and assume collective custody of content.

III. Architecture

By using a publish/subscribe paradigm, we attempt to unify two different common views of a network, namely that of a communication medium (most MANET research falls into this category) with that of a distributed data store (which is the focus of most research in peer-to-peer networking). This unification naturally leads to an architecture for integrated multi-party communication and search with in-network caching, temporal decoupling, and late binding, as exemplified by Haggle, which serves as our starting point.

As an extension and partial refactoring of Haggle, the ICEMAN architecture is an event-based architecture, in which multiple managers cooperate in a layer-less fashion to provide content-based services. It is a highly multi-threaded architecture where managers coordinate with each other asynchronously through events and manage a set of dynamically instantiated modules to perform computationally expensive operations in their own threads. For instance, data objects are managed by Haggle's data manager, which uses SQLite to store metadata and serves as a matching engine running in its own background thread with a separate task queue.

The fundamental unit of abstraction is a *data object* O associated with *metadata* $M(O)$, represented as a set of attribute/value pairs, and a *payload* $P(O)$, which is represented by a file. Each data object has a creation timestamp attribute, so that its creation time $TS(O)$ is well defined. A *data object identifier* $ID(O)$ is defined as the SHA1 hash over all this information, which is globally unique with high probability.

To provide content-based network services, two classes of data objects are disseminated: (1) *Exogenous data objects* that are directly or indirectly (e.g., using coding or encryption) used to transport *content*, i.e., application payload and associated attributes. (2) *Endogenous data objects* that support coordination and awareness between network nodes, such as *routing information objects* (if needed), and *node descriptions* for devices or applications. A *device node description* represents the cache summary of the device, while an *application node description* represents the application's interests. Node descriptions have a limited lifetime and are periodically disseminated over multiple hops. Each node maintains node descriptions of other nodes, even if they are not neighbors.

A. Declarative Attribute-Based Naming

ICEMAN takes a declarative naming approach where subscribers identify content through weighted attribute-value pairs with a similarity threshold. This generalization makes it straightforward to represent keywords and arbitrary combinations of conjunctions and disjunctions. By enabling applications to express interest with a suitable precision, ICEMAN efficiently pushes content discovery into the network layer.

Given a predicate $I(S)$ that represents the *interest* of a possible subscriber S, what matters is if it is satisfied by

a piece of content C, written as $C \models I(S)$. A dual notion is that of a *scope* $S(C)$ that can be associated with content C, and to decide if a node N is an eligible receiver, what matters is if it is satisfied by a given node $N \models S(C)$. At the most abstract level, the objective of ICEMAN is to efficiently transport content C that is published by some node P to each node S for which both $C \models I(S)$ and $N \models S(C)$ hold. By employing attribute-based encryption (see Section III-E) scopes are framed over node attributes (e.g., representing roles) and are interpreted as content access policies that can be cryptographically enforced.

Interest predicates are represented as a set of weighted attribute/value pairs, i.e., $I(S) \subseteq \mathbb{A} \times \mathbb{V} \times \mathbb{N}$, where $\mathbb{A}$, $\mathbb{V}$, and $\mathbb{N}$ denote the domains for attributes, values, and weights, respectively. ICEMAN's naming allows applications to logically specify how to quantify (and refine) the satisfaction of an interest predicate for a given piece of content C, by the degree of similarity metric. Specifically, we say that C satisfies $I(S)$ with a threshold s, written as $C \models_s I(S)$, iff

$$\frac{\sum\{w_i \mid (a_i, v_i, w_i) \in I(S) \cap M(C) \times \mathbb{N}\}}{\sum\{w_i \mid (a_i, v_i, w_i) \in I(S)\}} \geq s$$

In other words, the normalized weighted sum of overlapping attributes between content (M(C)) and interest (I(S)) determines the degree of matching or satisfaction.

ICEMAN considers the matching threshold (s) and a bound on the number of matches as part of the interest. Data objects are retrieved, ranked, and prioritized at each node using a lexicographical ordering based on the degree of matching and the creation time stamp (freshest first). Since an application can issue multiple concurrent threshold queries/subscriptions, it is straightforward to represent arbitrary combinations of conjunctions and disjunctions by transforming them into disjunctive normal form. Unlike Haggle, which uses a different semantics for local vs. remote queries, in ICEMAN matching is uniformly defined as stated above. In our generalization, interests are represented by application node descriptions and are disseminated separately from device node descriptions.

ICEMAN periodically disseminates cache summaries to avoid redundant transmissions and further refine the set of matched data objects. Each node maintains a counting Bloom filter representing the content in its local cache. When a node description is generated and sent to a neighbor, a compact non-counting abstraction of the local Bloom filter is included. Prior to sending a data object to a neighbor, the sender will first check to see if there is a Bloom filter hit for the data object in its local view of the neighbor's Bloom filter. Additionally, the Bloom filter reduces the data base query results to only those data objects that the interested party does not already have. In other words, if a node S has interest $I(S)$ and content approximated by $BF(S)$, only data objects satisfying $I(S) \wedge \neg BF(S)$ are sent towards S. Together, the interest and the Bloom filter define the *effective interest* of a node in a concise fashion. Bloom filter abstractions are generated periodically, so that eventual consistency is maintained between the long-term local Bloom filters and their disseminated short-term abstractions. To suppress immediate retransmissions each node's local perception of the peer's short-term Bloom filter abstraction is updated optimistically, and will be replaced by the actual peer's Bloom filter abstraction when its next node description is received.

B. Content Dissemination

ICEMAN transports data objects in a hop-by-hop fashion. It dynamically selects which transport protocol to use based on the content transport policy, which can be content-based, i.e., depending on attributes and payload size. All of the transport protocols support an application-layer atomic transaction protocol, the *control protocol*, which can suppress redundant transmissions at the cost of additional control messages. Currently we support TCP, UDP unicast, and UDP broadcast. Both UDP unicast and UDP broadcast can optionally disable the control protocol, in which case only Bloom filters are used to ensure delivery.

ICEMAN supports both proactive (push-based) as well as reactive (pull-based) dissemination algorithms, consistent with the observation in [25] that both paradigms are needed in content-based MANETs. ICEMAN dynamically selects which dissemination algorithm to use based on the content dissemination policy (e.g., depending on attributes and payload size).

1) Flooding and Replication: With *proactive flooding*, the data objects will be flooded to all nodes within the connected component of the publisher. This mechanism has been extended to *proactive replication* (epidemic propagation [29]) to push contact across newly discovered connected components.

A typical dissemination policy is to proactively flood important critical situation awareness information relevant within a squad, thus avoiding the cost of a round trip with the destination in a pull-based policy. If a message ferry is needed, then proactive replication may be a better choice to avoid additional round trip delays. It is also necessary to support one-way message ferrying.

Reactive counterparts of these algorithms are also supported, which means that content is *reactively flooded* or *reactively replicated* as soon as a matching interest is detected. Reactive replication provides an alternative method of dissemination that can deliver requested content with high probability if proactive dissemination is not feasible due to the large amount of available content.

2) Interest-Driven Routing: DIsruption REsilient Content Transport (DIRECT) is an interest-driven content dissemination protocol for DTNs developed in the DARPA DTN program. With some important changes described below, we have adopted DIRECT's interest propagation, and added DIRECT's reverse path method of content dissemination to ICEMAN. Specifically, interests are periodically epidemically disseminated with a creation timestamp and periodically purged. Upon a data object match with an interest, the data object is forwarded to the neighbor from which the interest was first received. We do not adopt DIRECT's use of CCN-style hierarchical naming and matching, nor its method of marking

queries in-active upon satisfaction to provide flow-balance between interest and content.

Unlike DIRECT, ICEMAN decouples interest dissemination from query satisfaction: ICEMAN can support both search and immediate routing of newly published information, even after the subscription has been issued. Another difference relative to DIRECT is the use of knowledge about cached content to minimize the probability of routing content that the subscriber has already obtained from other sources. This knowledge is explicitly disseminated in [27] and approximated through Bloom filters in the ICEMAN architecture. Through randomized propagation of node descriptions and hence interest, ICEMAN achieves multi-path diversity, which is especially useful together with network coding or fragmentation. Last but not least, interest-driven routing in ICEMAN can be combined with data object broadcast, which implies that data objects are pushed to and cached at overhearing nodes even if they were never requested.

3) Mobility-Driven Routing: ICEMAN supports mobility-driven routing using PRoPHET [21]. PRoPHET is a routing protocol designed for disconnected networks with non-random mobility. Experimental results in [21] show that with constrained cache sizes, PRoPHET can obtain higher delivery ratios with a modest increase (and sometimes decrease) in delay in comparison to epidemic routing. It uses a delivery predictability metric to estimate the probability that any particular destination can be reached through a particular neighbor. This delivery predictability metric is a based on each node's encounter history: nodes that meet frequently or for long durations have a high delivery predictability metric. Each node calculates its delivery predictability to every encountered node, and nodes exchange their delivery predictability vectors to transitively compute the probability of reaching a particular destination through a particular neighbor. In the context of ICEMAN, PRoPHET selects the neighbor with the highest delivery predictability when forwarding.

To serve as suitable basis for comparison, we have incorporated some of the newer ideas of the latest PRoPHET Internet Draft [9], most notably an improved transitivity rule, the periodic dissemination of routing information, and the periodic sampling of the current neighborhood to take into account contact duration. This enhanced version of PRoPHET works together with the periodic dissemination of node descriptions and matching as discussed previously. These modifications are needed, because PRoPHET is used in a content-based network where each node is a potential source, as opposed to its original use to route between two endpoints in DTNs.

C. Content- and Utility-Based Caching

Content-based caching is a feature of ICEMAN that aims to ensure that the amount of content managed by the network does not grow beyond its bounded capacity and that resources are primarily used for content that is relevant to the user. Content caching is a powerful mechanism to reduce latency and bandwidth, and mandatory if content needs to be transferred over multiple hops without a contemporaneous end-to-end path (e.g., using message ferrying or due to intermittent disruptions). Even with an end-to-end path, typical multi-hop loss rates over TCP will trigger end-to-end retransmissions, and render ICEMAN's caching-based store-and-forward solution more economic in terms of bandwidth if the content is sufficiently large. ICEMAN allows the specification of content-based caching strategies that enable fine-grained in-network purging of obsolete content, as opposed to end-to-end purging at the application level.

With *time-based purging* strategies, content can be purged either by an absolute or relative expiration time (relative to reception). The user can specify (1) a *tag* to denote the class of data objects to be purged, and (2) a *metric* to determine the absolute or relative time-to-live.

Another caching strategy inspired by our earlier work [17] is *order-based replacement*. While the concept is very general, the most common use is to keep only the *freshest* piece of content, while *staler* content is discarded from the data store. The user can specify (1) a *tag* to denote the class of data objects to be totally ordered, (2) an *id* to indicate the attribute that needs to match (e.g., content originator), and (3) a *metric* to determine the ordering of the objects (e.g., content creation time). Formally, (1) and (2) define an equivalence relation $\equiv$ on matching data objects and (3) defines a total order $\prec$ over all data objects in each equivalence class. This total-order replacement strategy only keeps the maximal element in each equivalence class that has been received. Multiple total replacement strategies can be composed in a prioritized fashion, for instance to define a lexicographical ordering, which is generally a partial order.

Content-based caching is further generalized to a *utility-based caching* pipeline which builds on the work in [28], [7] and frames the cache replacement and decision problem as a utility maximization problem. A caching policy defines a utility function which assigns a real number between 0 and 1 to each data object in the cache. This utility function is a composition of multiple utility functions that are content and context sensitive (they vary in time and space). Data objects that do not meet a minimum threshold (as specified by the policy) are immediately evicted. Once the cache exceeds a certain watermark capacity, the pipeline choses which data objects to evict in order to bring the cache capacity under the watermark. This eviction selection is posed as a 0-1 knapsack problem where the watermark capacity is the bag size, the data object payload size is the cost, and the computed utility is the benefit. By specifying suitable utility functions various combinations of popularity-based and cooperative caching strategies can be expressed in this framework.

D. Network Coding and Fragmentation

In a MANET environment, *network coding* can take advantage of the broadcast nature of transmissions as well as node mobility [19]. To overcome intermittent connectivity and to allow content dissemination in a decentralized setting, ICEMAN can perform network coding at the level of data objects (depending on content size and other factors) rather

than individual packets. ICEMAN specifically exploits the capability of network coding to mitigate the last coupon collector problem. In our targeted applications, groups may merge and split dynamically. When groups merge they can exchange innovative blocks which will expedite the reconstruction of the transmitted content.

In addition to network coding, ICEMAN supports *randomized informed fragmentation* to support scenarios where network coding is not needed or the overhead incurred by network coding is too high. We call it informed, because the sender examines the receivers Bloom filter and selects a random subset of fragments from the peers set of missing fragments. Randomizing the selection subset across multiple nodes increases the likelihood that different fragments are received by a node concurrently from different sources. Network coding can be combined with fragmentation, in which case the fragments are also known as generations. Multiple generations are needed when content is too large to be solely network coded due to the overhead of the associated vectors.

Blocks and fragments are cached and disseminated by intermediate nodes. Both coded blocks and uncoded fragments remain unchanged; i.e. different from random- linear network coding, ICEMAN does not perform mixing of blocks at intermediate nodes, but peers can become new seeds of innovative blocks upon reconstruction.

E. Security

ICEMAN leverages any underlying link- or network-layer security mechanisms, but our work focuses on providing an independent layer of security that secures the content directly. End-to-end security properties, namely non-repudiation and confidentiality are based on digital signatures and attribute-based encryption [6]. In both cases, we use protocols that support multiple certification authorities. Our current architecture secures the payload, while security for metadata is a challenging research topic left for future work.

Nodes have their signing keys certified by one or more authorities. Each node only accepts content from a neighbor if they share a certification authority. This prevents an attacker (insider or outsider) from polluting the network without exposing his (assumed) identity. Simultaneously, the availability of multiple authorities ensures that trust can be flexibly and robustly bootstrapped.

Publishers can limit access to content by specifying access policies framed over node attributes. Policies are specified with a range of operators, including conjunction and disjunction, allowing expressive authorization, and can combine attributes from multiple authorities [20]. Similarly, nodes can receive their attributes from multiple authorities. During publication, content is encrypted with a policy, ensuring that access control is enforced cryptographically with an end-to-end guarantee of confidentiality despite the flexible access specification. Hybrid encryption is used to optimize performance, with AES [1] used for the content, and multi-authority attribute-based encryption in the Charm framework [4] used to encrypt the AES keys.

As a basic mechanism to reject unwanted traffic, signatures are not only used on exogenous data objects, but on all other endogenous data objects. The overhead of signing is low due to our choice of a relatively large fragment/block size. This approach is reasonable under a closed-system model, where it is difficult for an attacker to generate new valid identities that are accepted by the original nodes of the network. Taking control of existing nodes is the only practical way to do so. The capability to exclude nodes from the network is a stepping stone for an architecture that attempts to maintain network availability by a notion of trust that evolves over time.

IV. Summary of Evaluation

We conducted an extensive evaluation of ICEMAN through CORE/EMANE [3] emulation to understand the performance characteristics of different policies. We modeled a tactical network consisting of 30 nodes (3 squads of 10) with different classes of situational awareness traffic. We found that different dissemination, transport and caching policies have significantly different performance characteristics (in terms of total data objects delivered and delay). A combination of content-based policies was necessary to achieve the best performance (e.g., epidemic broadcast for node descriptions and interest driven routing and network coding for large data objects and high channel contention). Combinations of hard- and soft-constraint utility-based caching policies that intelligently rank data according to network context achieved higher performance than only using hard-constraint policies such as time-based purging and order-based replacement. Battery lifetime results on Nexus S phones demonstrated the feasibility of ICEMAN on current hardware, where CPU intensive policies such as network coding achieved higher performance than alternative policies. Similarly, security performance tests demonstrate that policy caching can achieve significant performance improvements, making efficient attribute-based encryption feasible on mobile devices.

V. Conclusion

We have introduced a new ICN architecture where scope and interest are dual concepts associated with publishers and subscribers, respectively, and uniformly expressed in an attribute-based framework. The design of our ICEMAN architecture emphasizes compositionality in the sense that all features seamlessly interoperate with each other. Without architectural changes, our system supports any combination of the discussed caching, transport, and dissemination mechanisms. All features are independently configurable and for backward compatibility and performance comparisons we support the original feature set of Haggle.

The utility-based caching framework is a first step towards a unified utility-based architecture that formulates content dissemination, caching, and resource management policy selection as an online utility maximization problem.

We also plan to add a higher-level distributed monitoring and optimization component to maximize content availability based on an analysis of the tradeoff space of policies and

parameters. By quantifying their benefit and cost, ICEMAN can potentially improve the overall system utility, for instance using an approach similar to cross-layer optimization [16]. Distributed monitoring plays another role in the detection of unexpected behaviors such as using an excessive amount of resources. It can also detect violations of properties (expected invariants) and their combinations that could indicate compromised devices or attacks. An adaptive trust management component could utilize this information to exclude misbehaving nodes from the network or require additional confirmation.

Attribute-based naming is a first step towards a logic, but there is much more potential in the declarative approach to content-based networking by further increasing the expressiveness of queries and subscriptions. For instance, predicate-based naming with OWL/RDF [18] has recently been implemented on top of ICEMAN in the context of the DARPA CBMEN [8] Program by the Drexel university team. ICEMAN has a transport architecture that can support other transport mechanisms, such as NORM [2] which is currently being integrated with our architecture in the scope of same progam.

With ICEMAN we are exploring a new area of the networking space that is quite different from existing research on MANETs and peer-to-peer networks. The need for a higher level of abstraction and increased expressiveness means that data objects have a much higher constant overhead than packets in IP; ICEMAN operates at a higher time scale and a level of content-granularity to amortize the cost. On the other hand, the transition to a higher level of abstractions seems essential to solve the problems that face traditional approaches by being too distant from the actual needs of applications. More interestingly, it opens opportunities for new mobile applications of the future, where the network architecture can provide services and optimize resources based on what the content represents and how it is used.

REFERENCES

[1] Federal information processing standards, publication 197: Advanced encryption standard. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[2] B. Adamson, C. Bormann, M. Handley, and J. Macker. Negative-acknowledgment (nack)-oriented reliable multicast (norm) protocol. *Internet Society Request for Comments RFC*, 3940, 2004.

[3] J. Ahrenholz, C. Danilov, T.R. Henderson, and J.H. Kim. Core: A real-time network emulator. In *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pages 1–7. IEEE, 2008.

[4] Joseph Akinyele, Matt Green, and Avi Rubin. Charm: A framework for rapidly prototyping cryptosystems. Technical report, Johns Hopkins University, 2011.

[5] P. Basu, R. Krishnan, and D. W. Brown. Persistent delivery with deferred binding to descriptively named destinations. In *Proc. of IEEE Military Communications Conference*, 2008.

[6] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. *28th IEEE Symposium on Security and Privacy*, 2007.

[7] N. Chand, RC Joshi, and M. Misra. Cooperative caching in mobile ad hoc networks based on data utility. *Mobile Information Systems*, 3:19–37, 2007.

[8] Defense Advanced Research Projects Agency (DARPA). Content-based mobile edge networking. http://www.darpa.mil/Our_Work/STO/Programs/Content-Based_Mobile_Edge_Networking_(CBMEN).aspx, 2012.

[9] A. Lindgren et. al. Probabilistic routing protocol for intermittently connected networks. *Internet Draft draft-irtf-dtnrg-prophet-10*, 2012.

[10] W. Finn. Improving battlefield connectivity for dismounted forces. *Defense Tech Briefs*, pages 6–10, 2012.

[11] S. Frink. Secure cell phone technology gets ready for deployment. *Military and Aerospace Electronics*, pages 10–19, 2012.

[12] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Information-centric networking: seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, page 1. ACM, 2011.

[13] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 244–251, New York, NY, USA, 2005. ACM.

[14] Stratis Ioannidis, Augustin Chaintreau, and Laurent Massoulié. Distributing content updates over a mobile social network. *SIGMOBILE Mob. Comput. Commun. Rev.*, 13:44–47, June 2009.

[15] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.

[16] Minyoung Kim, Je-Min Kim, Mark-Oliver Stehr, Ashish Gehani, Dawood Tariq, and Jin-Soo Kim. Maximizing availability of content in disruptive environments by cross-layer optimization. In *28th ACM Symposium on Applied Computing (SAC)*, 2013.

[17] Minyoung Kim, Mark-Oliver Stehr, Jinwoo Kim, and Soonhoi Ha. An application framework for loosely coupled networked cyber-physical systems. In *8th IEEE International Conference on Embedded and Ubiquitous Computing (EUC-10), Hong Kong, December, 2010*.

[18] Joseph B. Kopena and Boon Thau Loo. Ontonet: Scalable knowledge-based networking. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering Workshop*, ICDEW '08, pages 170–175, Washington, DC, USA, 2008. IEEE Computer Society.

[19] Uichin Lee, Joon-Sang Park, Joseph Yeh, Giovanni Pau, and Mario Gerla. Code torrent: content distribution using network coding in vanet. In *Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking*, MobiShare '06, pages 1–5, New York, NY, USA, 2006. ACM.

[20] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. *30th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 6632, 2011.

[21] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):19–20, 2003.

[22] Abderrahmen Mtibaa, Martin May, Christophe Diot, and Mostafa Ammar. Peoplerank: social opportunistic forwarding. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 111–115, Piscataway, NJ, USA, 2010. IEEE Press.

[23] Mirco Musolesi, Pan Hui, Cecilia Mascolo, and Jon Crowcroft. Writing on the clean slate: Implementing a socially-aware protocol in Haggle. In *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6, Washington, DC, USA, 2008. IEEE Computer Society.

[24] Erik Nordstrom, Per Gunningberg, and Christian Rohner. A search-based network architecture for mobile devices. Technical report, Uppsala University, 2009.

[25] Soon-Young Oh, Davide Lau, and Mario Gerla. Content centric networking in tactical and emergency MANETs. In *Wireless Days*, pages 1–5. IEEE, 2010.

[26] Ignacio Solis and J. J. Garcia-Luna-Aceves. Robust content dissemination in disrupted environments. In *Proceedings of the Third ACM Workshop on Challenged Networks*, CHANTS '08, pages 3–10, New York, NY, USA, 2008. ACM.

[27] Mark-Oliver Stehr and Carolyn Talcott. Planning and learning algorithms for routing in disruption-tolerant networks. In *IEEE Military Communications Conference*, 2008.

[28] K. Obraczka T. Spyropoulos, T. Turletti. Routing in delay-tolerant networks comprising heterogeneous node populations. *IEEE Transactions on Mobile Computing*, pages 1132–1147, 2009.

[29] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, Technical Report CS-200006, Duke University, 2000.