**Title**

SOCRATIC: A social approach to network coding rate control

**Authors**

Garcia-Luna-Aceves, J.J.
Wood, S.
Sadjadpour, S.

**Publication Date**

2014-12-08

# SOCRATIC:
# A Social Approach to Network Coding Rate Control

Samuel Wood*†, Hamid Sadjadpour*†, J.J. Garcia-Luna-Aceves*†
*UC Santa Cruz, †SUNS-tech, Inc.
{sam, hamid, jj}@suns-tech.com

***Abstract*—Tactical and emergency-response networks require efficient communication without a managed infrastructure. Recent work demonstrates that applying information-centric paradigms to the tactical edge can provide performance benefits over traditional address centric approaches. We propose SOCRATIC (SOCial RATe control for Information Centric networks), an approach that unifies replication and network coding to disseminate content by taking advantage of social content and context heuristics. SOCRATIC replicates network encoded blocks according to a *popularity index* metric that is shared during neighbor discovery. The number of encoded blocks that is relayed to a node depends on its own interest in a data object and its social popularity, i.e., how often and for how long the node meets other nodes. These blocks are subsequently replicated towards the subscriber if a stable path exists. We evaluate an implementation of SOCRATIC through network emulation of a tactical scenario and demonstrate that it can achieve better performance than traditional socially agnostic approaches.**

## I. INTRODUCTION

Disaster-response and tactical networks require efficient delay tolerant communication without any fixed infrastructure guarantees. Our research focuses on efficient content dissemination in tactical scenarios consisting of several small groups (called squads) of up to 10 nodes each that are persistently connected within several hops. Inter-squad communication is typically non-contemporaneous and can require message ferrying through members of another squad, or using a UAV. Nodes in these networks that are more social or have more interests in common are more likely to be co-located. SOCRATIC (SOCial RATe control for Information Centric networks) is a replication approach to content dissemination in information-centric, delay-tolerant networks (ICDTN [1]) that exploits social content and context information to improve network performance. Unlike previous approaches, SOCRATIC replicates a bounded number of network coded blocks according to the satisfaction degree of the data object and the potential relay, along with other social-context heuristics. We have implemented and evaluated SOCRATIC as an extension of the ICEMAN [2] system.

Recent work demonstrates that the information centric networking (ICN) paradigm can improve content delivery and reduce latency in volatile networks [3], [4]. The ICN approach changes the communication abstraction provided by the network layer from a socket between a pair of location-based addresses to a data object referenced by name. In ICN, applications communicate with the network layer through an API based on publish-subscribe primitives. This approach enables communication over non-contemporaneous paths and supports ubiquitous caching: subscriber applications only specify *what* content they desire (not *where*), and the network decides *how* to deliver this content to the subscriber. Data objects are cached opportunistically at intermediate relays along the path to a subscriber, according to a cache eviction policy. Similarly, publisher applications only specify *what* content they want to share, along with metadata to define a name that is used by the subscribers to reference the data object. The main challenge for ICDTNs is to efficiently deliver data objects to subscribers from network caches, possibly the publisher. Given that global knowledge is unavailable in ICDTNs, nodes must make local decisions based on limited network knowledge and intermittent neighbor connectivity. SOCRATIC adopts ICEMAN's naming scheme where an application subscribes to data objects by specifying a human readable list of attributes and a matching threshold: this scheme generalizes exact match and computes a *satisfaction degree* for an $\langle$ node, data object $\rangle$ pair.

As an alternative to delay-tolerant forwarding techniques, replication is used to improve dissemination performance in networks with non-contemporaneous paths. With replication, the publisher stores replicas of a data object at other relay nodes that need not be interested in the data object, with the intent that these relays will come into contact with a subscriber. The challenge with replication in this context consists of selecting relays so that the data object delivery has high probability and the network usage is low. SOCRATIC uses social heuristics to replicate data objects to nodes that have a higher probability of meeting the subscriber.

An orthogonal approach to replication that is adopted in error-prone networks with limited connectivity is *network coding*. This technique breaks a data object into multiple smaller blocks that are encoded and disseminated independently, so that receivers can reconstruct the original data object using a smaller amount of network resources than naive fragmentation[1] approaches [5].

In contrast with traditional replication approaches that operate on a data object level, SOCRATIC intelligently replicates a bounded number of network-coded blocks that are subsequently replicated. SOCRATIC enforces a type of rate control that limits the total number of generated network-coded blocks and imposes a dynamic limit for each relay.

---

[1] In the fragmentation approach, the data object is divided into smaller fragments and then each fragment is transmitted toward the destination.

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

In SOCRATIC, a publisher for a data object decides to replicate a bounded number of coded blocks to each encountered neighbor based on the neighbor's *popularity index*. Nodes that frequently visit new neighbors, visit neighbors for long periods of time, or have more interest in the data object, have a higher popularity index and receive more blocks. Using this criteria, SOCRATIC can achieve better network performance than approaches to replication that are socially-agnostic. Section II discusses related content dissemination work in volatile networks. Section III details our approach. Section IV evaluates an implementation of SOCRATIC in several emulated tactical scenarios. We conclude the paper in Section V.

## II. RELATED WORK

We briefly describe related DTN approaches to content dissemination. Epidemic dissemination [6] is one baseline for DTN routing protocols: in its simplest form, nodes exchange cache summaries and replicate each missing data object so that their caches are identical. To improve epidemic dissemination, PRoPHET [7] reduces the number of replicas by sharing encounter history summaries among nodes and by transitively building an encounter probability matrix; this results in replication of content to nodes that have a high probability of meeting the destination. Bounded replica approaches such as Spray and Wait [8] enforce an explicit constraint on the number of data object replicas to finely control network usage; relays may further replicate the data object so long as the bound is not exceeded (e.g., using *forwarding tokens*), and relay selection is based on first encounter. Utility-based DTN approaches (e.g., [9], [10]) select relays and data objects for replication according to a generic utility function, and may also support bounded replicas or a utility threshold. Typical utility functions aim to reduce delivery latency and network usage by selecting nodes based on the content (e.g., data object creation time), context (e.g., node mobility as in most-social-first) or both (e.g., last node to see the data object's destination first, or minimize delay). Hui et al [11] identify the importance of using network centrality and social community knowledge when routing in packet switched networks, and propose a method that exploits both criteria in their technique called BUBBLE. Similar to this work, SOCRATIC estimates a node's centrality using a metric that encompasses the node degree over time. Unlike BUBBLE, SOCRATIC is designed for an ICN and uses the explicit subscriptions to infer social community, and includes link duration in the estimate of network centrality.

The discussed DTN approaches are agnostic of a data object's interests and are sender driven. Our research investigates DTN and ICN, which is receiver driven and has separate interest dissemination and content dissemination phases. SocialCast [12] is a utility-based replication scheme for DTN pub-sub systems that uses interests and a social heuristic to compute a single value for the utility of storing a replica at a particular node, using utility forecasting. In this scheme, data objects are forwarded only to nodes with a greater utility. The DIRECT protocol [13] replicates data objects towards the direction of the interest originator. Interests are propagated epidemically across connected components, and data objects traverse the reverse path of the interest propagation. Periodic purging and refreshing of interests ensure that stale paths are discarded and that new paths are discovered. Recent work [2] has demonstrated the value of combining network coding, ICN [14] and DTN [5], using interest-based routing. In the ICEMAN system, network coding can achieve significant performance benefits over fragmentation [15]. As an extension of ICEMAN, SOCRATIC uses DIRECT on network coded blocks when a path to the subscriber exists. Unlike existing approaches, SOCRATIC sets a global bound on the number of network encoded blocks for each data object, and a dynamic bound for each ⟨ relay node, data object ⟩ pair. To the best of our knowledge, SOCRATIC is the first proposal to set this dynamic bound based on an interest satisfaction metric. Furthermore, none of the existing social ICN and DTN replication approaches operate on network encoded blocks.

## III. SOCRATIC

SOCRATIC is an extension of ICEMAN, an ICN for tactical networks that uses weighted-attribute naming (this naming expands upon [16]). We briefly describe ICEMAN's methods for network coding, opportunistic caching, and routing, and then describe the SOCRATIC extension.

The approach to naming in SOCRATIC fits tactical networks very well by pushing content discovery into the network layer. Only data objects that match an interest with a satisfaction greater than a threshold are transferred. Formally, interest predicates for node $S$ are represented as a set of weighted attribute/value pairs, $I(S) \subseteq \mathbb{A} \times \mathbb{V} \times \mathbb{N}$, where $\mathbb{A}$, $\mathbb{V}$, and $\mathbb{N}$ denote the domains for attributes, values, and weights. We say that content $C$ with weighted attributes $M(C)$ satisfies interest $I(S)$ for node $S$ with a satisfaction degree $\lambda(C,S)$ defined as in [2], where

$$\lambda(C,S) := \frac{\sum\{w_i \mid (a_i,v_i,w_i) \in I(S) \cap M(C) \times \mathbb{N}\}}{\sum\{w_i \mid (a_i,v_i,w_i) \in I(S)\}}. \quad (1)$$

We say that $C$ matches $S$ with threshold $s$, written as $C \models_s I(S)$, iff $\lambda(C,S) \geq s$. In other words, the normalized weighted sum of overlapping attributes between content $M(C)$ and interest $I(S)$ determines the satisfaction degree.

Interests are propagated epidemically and periodically purged. Content dissemination occurs either when an interest from node $S$ with threshold $s$ arrives at node $P$ with a content $C$ such that $C \models_s I(S)$, or when a data object $C$ arrives at node $R$ who has recently received an interest from node $S$ such that $C \models_s I(S)$. In both cases we configured ICEMAN to replicate $C$ to the neighbor from which $P$ or $R$ received the interests from $S$[2], effectively performing DIRECT.

After a data object $C$ was published at node $P$ and an interest from $S$ has arrived such that $C \models_s I(S)$, $P$ will generate network coded blocks $C_0, C_1, \ldots C_k$ to replicate towards $S$

---

[2] A cache summary Bloom filter is included with the interests in a *node description* data object, to avoid redundant transmissions.

via relay $R$ until either: 1) $R$ and $P$ become disconnected, or 2) $R$ has received enough blocks to reconstruct $C$. We have limited the number of encoded blocks that the publisher can generate and transmit in order to avoid flooding the network with the content. Each block $C_i$ is cached and routed just like any other data object, and inherits the attributes of $C$: $M(C_i) \supset M(C)$.

### *A. Popularity Index*

SOCRATIC extends ICEMAN as follows. To measure a node's popularity, each node $R$ maintains a log of all of its previous contacts within a window of the past $w_t$ seconds. This log is used to compute the node's average connect time $A^R_{vt}$ and the number of encountered distinct neighbors $N^R_e$. Since SOCRATIC is deployed in a managed tactical network, we assume that each node has an estimate of the network size, $N$. We use the average connect time to compute a scalar $N^R_{avt}$ given by

$$N^R_{avt} := \begin{cases} 2 & \text{if } A^R_{vt} \geq \beta_2, \\ 1 & \text{if } \beta_1 \leq A^R_{vt} < \beta_2, \\ 0 & \text{if } A^R_{vt} < \beta_1, \end{cases} \tag{2}$$

where $\beta_1, \beta_2$ are parameters that are set *a priori* according to network characteristics (15, 30 seconds in our scenarios). Node $R$ computes its popularity index as

$$P^R := \frac{N^R_e * N^R_{avt}}{N}. \tag{3}$$

Nodes exchange their popularity indices during neighbor discovery. When content dissemination occurs at node $P$ for content $C$ and interest from neighbor $R$, if content $C \not\models_s I(R)$ (content $C$ does not pass $R$'s satisfaction threshold $s$) then the SOCRATIC algorithm is activated to compute the number of network coded blocks to generate and forward to the neighbor. If $C \models_s I(R)$, then $P$ reverts to the previously described ICEMAN mechanism. With SOCRATIC, $P$ computes a satisfaction scalar $M(C, R)$ as

$$M(C,R) := \begin{cases} 2 & \text{if } 0.75 * s \leq \lambda(C,R) < s, \\ 1.5 & \text{if } 0.5 * s \leq \lambda(C,R) < 0.75 * s, \\ 1 & \text{if } 0.25 * s \leq \lambda(C,R) < 0.5 * s, \\ 0.5 & \text{if } \lambda(C,R) < 0.25 * s. \end{cases} \tag{4}$$

Let $B(C)$ be the number of blocks needed to decode $C$. $P$ computes a maximum number of blocks of $C$ to replicate to $R$, $B(C, R)$, as

$$B(C,R) := P^R * M(C,R) * B(C). \tag{5}$$

Let $\delta_P(C, R)$ be the number of generated blocks for $C$ that are successfully sent by publisher $P$ to $R$, and $\delta_P(C)$ be the total number of successfully sent blocks for $C$. Each time $P$ encounters some neighbor $R$, $P$ greedily sends blocks to $R$ to maximize $\delta_P(C, R)$ according to two constraints, namely, C1) $\delta_P(C,R) \leq B(C,R)$ and C2) $\delta_P(C) \leq \gamma * B(C)$. $\gamma$ is a replication parameter that is set *a priori* according to network characteristics (1.5 in our scenarios).

### *B. Example Scenario*

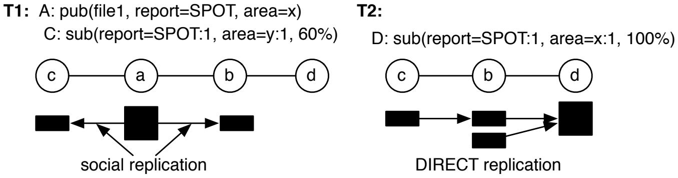

Fig. 1. An example scenario illustrating all of the mechanisms of SOCRATIC. The black square represents an decoded data object, and the rectangles represent encoded blocks of that data object.

Figure 1 illustrates a simplified scenario that demonstrates all of the mechanisms of SOCRATIC. At time T1 node *a* publishes *file1* with 2 attributes and node *c* subscribes to 2 attributes, each with a weight of 1, and a threshold of 60%, or $s = 0.6$. We assume $\gamma = 1.5$, $B(file1) = 4$, node *b* encounters a number of neighbors for durations such that $N^b_{avt} = 2$, $N^b_e = 2$, $P^b = 1$, and node *c* has $N^c_{avt} = 1$, $N^c_e = 1$, $P^c = 0.25$. Given the subscriptions, we have $M(file1, b) = 0.5$ and $M(file1, c) = 2$. Thus $a$ computes $B(file1, b) = 1{*}0.5{*}4 = 2$ and $B(file1, c) = 0.25{*}2{*}4 = 2$, and will generate 4 linearly independent encoded blocks: 2 that are sent to *c* and 2 that are sent to *b* by constraint C1 (constraint C2 does not apply since the number of blocks, 2, is less than $1.5{*}4 = 6$). Later, at time T2, *c* and *b* are connected with *d* which has an exact match interest with the original data object, and thus its encoded blocks. Since there is a stable path between *c,b,d* and $file1 \models_1 I(d)$, DIRECT is triggered which replicates these blocks to *d* who subsequently reconstructs the file. Note that decoding only occurs at the subscriber, and that the publisher is solely responsible for determining the number of blocks generated.

### *C. Discussion*

From these definitions and the example scenario, it is clear that the number of encoded blocks that are generated for a neighbor is proportional to a node's popularity and its interest commonality with the data object. $N^R_{avt}$ acts as a filter to prevent sending blocks to nodes with a very short contact time despite their popularity (i.e. a lossy interface), and gives more weight to nodes that have a longer contact time. Through $N^R_e$, nodes that meet many other nodes will receive more blocks. $M(C, R)$ exploits the observation that in tactical networks nodes with similar interests tend to be co-located.

From a theoretical perspective, if $D$ denotes the number of published data objects at a particular node, and $N_e$ denotes the number of encountered neighbors, then this algorithm requires $\mathrm{O}\big(D * N_e\big)$ additional memory at the node when compared to ICEMAN. The other additional memory costs are negligible. In practice, the block counters and data objects should be aged and purged to save memory. Typically data object purging is done through a maximum hop-count TTL or timestamp which indicates when the content is no longer useful, although more complicated policies are supported such as enforcing a total order on classes of content (e.g., video frame data objects with a sequence number attribute). The parameters $w_t$, $\beta_1$, $\beta_2$, $\gamma$ are network dependent and are determined *a priori*, typically through extensive emulation.

## IV. PERFORMANCE EVALUATION

SOCRATIC is implemented as a module within ICEMAN and intercepts events such as neighbor discovery and data object send and receive. Regular ICEMAN is structured so that either data objects do or do not match an interest, however SOCRATIC uses the satisfaction degree directly to make decisions. We evaluated the implementation in the CORE [17] network emulator, where each node is associated with a Linux light-weight container running SOCRATIC. We used the CORE basic MAC which does not model layer 1 or 2 phenomena (lossless 11Mbps pipes with 50ms delay). Each node has a broadcast radius of 250m and unlimited cache capacity. Each data object is 1MB and the block size is 32KB. We compared SOCRATIC against four other content dissemination algorithms with network coding enabled: DIRECT[3], EPIDEMIC, SPRAY, and sSPRAY. DIRECT is ICEMAN as previously described, with DIRECT routing and constraint C2 (defined in Section III-A). EPIDEMIC is ICEMAN with Epidemic [6] routing and constraint C2. SPRAY is SOCRATIC without DIRECT routing and only constraint C2 (it replicates on first encounter). sSPRAY (social SPRAY) is SOCRATIC without DIRECT. The SPRAY approach is similar to social oblivious replication approaches that replicate on first encounter, while sSPRAY incorporates the social popularity index to give replication preference to more social nodes. All experiments are averaged over 4 runs, each with a different mobility seed.

### A. Disconnected Scenario

This scenario models 4 squads of 5 nodes on patrol in 4 different regions, with 2 UAVs that circle between the squads. The UAVs cannot communicate directly with each other. Figure 2 demonstrates the scenario that we have emulated. The total area for the emulation is 1300x1300 $m^2$ and the duration of each experiment is 30 minutes. There are two UAVs flying around the area with a velocity of 30 $meters/sec$ around a circle with radius of 550 meters. Each member of a squad has a mobility pattern of Random Waypoint with minimum and maximum velocity of 1 and 7 $meters/sec$, respectively. The minimum and maximum pause time for each group member is zero and five seconds, respectively. 25 data objects are published at the beginning of the test, where each data object has a single subscriber with a 100% satisfaction degree and a 100% threshold. The UAVs have 75% satisfaction degree with all of the data objects (with a 100% threshold). Publisher-subscriber pairs are chosen uniformly at random, with the constraint that the publisher and subscriber belong to different squads. This constraint implies that all inter-squad communication must occur through the UAV message ferries.

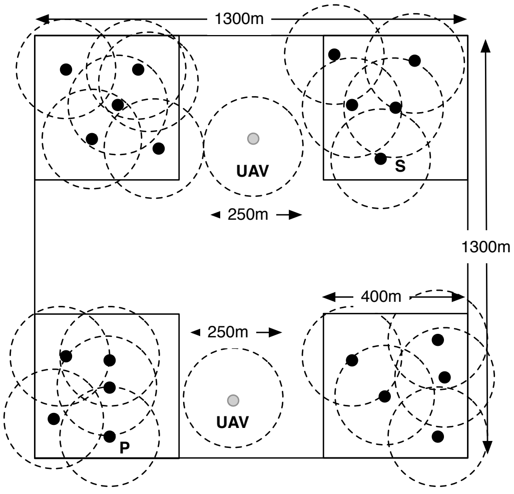

Fig. 2. Disconnected scenario. Node "P" publishes a data object that only node "S" subscribes to. "UAV" nodes ferry the data object from "P" to "S".

Figure 3 graphs the delivery vs. delay across the different dissemination approaches. Since Epidemic dissemination is usually considered a baseline for DTN routing protocols, our comparison shows that SOCRATIC outperforms this approach and delivers the most data objects within the delay constraint. It is only when the delay constraint is very high that EPIDEMIC slightly outperforms SOCRATIC because the network is less congested after the initial flood of data objects which results in faster delivery while SOCRATIC only caches the data object judiciously to save network resources. The figure also shows that SOCRATIC clearly outperforms other techniques for all level of delay constraints.

In comparing SPRAY and sSPRAY, it is clear that the popularity index captures the higher social value of the UAVs, enabling sSPRAY to replicate more blocks directly to the UAVs than SPRAY which replicates most of the blocks to its immediate neighbors who are not in contact with the subscriber. In comparing DIRECT and sSPRAY, we see the benefit of replicating blocks multiple hops along the path that the last interest was received–this tends to be through a UAV directly or an intermediate node to a UAV if it recently received the interest.

Figure 4 shows the total bandwidth utilized (for transmit (Tx) and received (Rx)) along with total data object (DO) delivery. SOCRATIC delivers 100% of data objects while other techniques deliver less that 100% except EPIDEMIC approach. This reduction in bandwidth usage enables SOCRATIC to deliver most of the data objects within less delay than using EPIDEMIC which quickly saturates the bandwidth. Moreover, SOCRATIC disseminates the information intelligently and therefore, requires less bandwidth usage for the same delivery compared to EPIDEMIC approach. Given the interference and scarcity of spectrum in dense wireless environments, SOCRATIC clearly outperforms all these techniques while

---

[3] The interest refresh was set to 20 seconds, with a 30 second purge.

caches selectively.

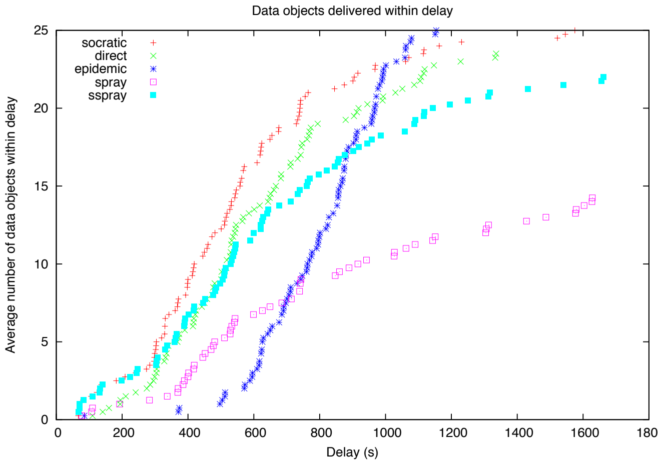

Fig. 3. Average delay distribution for the disconnected scenario.

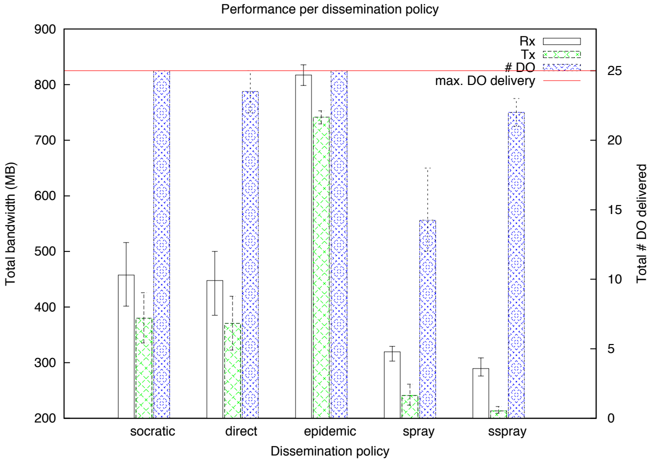

Fig. 4. Average transmit bytes, receive bytes, and data objects delivered for the disconnected scenario. The error bars represent the min and max across all runs.

## V. CONCLUSION

This paper introduces a new replication approach called SOCRATIC that unifies replication and network coding to efficiently disseminate content through delay-tolerant networks (DTNs). The main idea behind SOCRATIC is to cache network encoded blocks in nodes that are more relevant in terms of their interest in the original data object, and nodes that have a higher probability of meeting another node that has interest in this content by computing a popularity index. The popularity index computes a value that is proportional to the interest commonality of node to the data object, average contact time for the node and the average number of nodes that the node meets. SOCRATIC takes advantage of the distributed nature of random linear network coding [18] and controls the number of encoded blocks that a cache receives. This judicious distribution of the encoded blocks allows the network to utilize minimum network resources (bandwidth) while maximizing the data delivery. Emulation results have shown that this technique outperforms several replication approaches proposed in the DTN literature.

This work can be extended to wireless mobile ad hoc networks (MANETs) where nodes are mainly connected in the network. In such networks, the popularity index can be redefined to consider the unique characteristics of MANETs.

## REFERENCES

[1] G. Tyson, J. Bigham, and E. Bodanese, “Towards an information-centric delay-tolerant network,” in *Proceedings of IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN)*, 2013.

[2] S. Wood, J. Mathewson, J. Joy, M.-O. Stehr, M. Kim, A. Gehani, M. Gerla, H. Sadjadpour, and J. Garcia-Luna-Aceves, “ICEMAN: A system for efficient, robust and secure situational awareness at the network edge,” in *MILCOM*. IEEE, 2013.

[3] S.-Y. Oh, D. Lau, and M. Gerla, “Content centric networking in tactical and emergency MANETs,” in *Wireless Days, IFIP*. IEEE, 2010, pp. 1–5.

[4] V. Kawadia, N. Riga, J. Opper, and D. Sampath, “Slinky: An adaptive protocol for content access in disruption-tolerant ad hoc networks,” in *ACM MobiHoc 2011 International Workshop on Tactical Mobile Ad Hoc Networking*, 2011.

[5] U. Lee, J.-S. Park, J. Yeh, G. Pau, and M. 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*. ACM, 2006, pp. 1–5.

[6] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Technical Report CS-200006, Duke University, Tech. Rep., 2000.

[7] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[8] K. Psounis and C. S. Raghavendra, “Spray and wait: An efficient routing scheme for intermittently connected mobile networks,” in *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005.

[9] A. Balasubramanian, B. Levine, and A. Venkataramani, “DTN routing as a resource allocation problem,” in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, 2007, pp. 373–384.

[10] T. Spyropoulos, T. Turletti, and K. Obraczka, “Routing in delay-tolerant networks comprising heterogeneous node populations,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1132–1147, 2009.

[11] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay-tolerant networks,” *Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[12] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, “Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks,” *Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 748–760, 2008.

[13] I. Solis and J. Garcia-Luna-Aceves, “Robust content dissemination in disrupted environments,” in *Proceedings of the Third ACM Workshop on Challenged Networks*. ACM, 2008, pp. 3–10.

[14] M.-J. Montpetit, C. Westphal, and D. Trossen, “Network coding meets information-centric networking,” *arXiv preprint arXiv:1201.2387*, 2012.

[15] J. Joy, Y.-T. Yu, M. Gerla, S. Wood, J. Mathewson, and M.-O. Stehr, “Network coding for content-based intermittently connected emergency networks,” in *Proceedings of the 19th Annual International Conference on Mobile computing and Networking*. ACM, 2013, pp. 123–126.

[16] E. Nordström, P. Gunningberg, and C. Rohner, “A search-based network architecture for mobile devices,” *Department of Information Technology, Uppsala University, Tech. Rep*, vol. 3, 2009.

[17] J. Ahrenholz, “Comparison of CORE network emulation platforms,” in *MILCOM*. IEEE, 2010, pp. 166–171.

[18] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, , and B. Leong, “A random linear network coding approach to multicast,” in *IEEE Transactions on Information Theory*, 2006, pp. 4413–4430.