CARE: Content Aware Redundancy Elimination for Challenged Networks

Udi Weinsberg[1]   Qingxi Li[2]   Nina Taft[1]   Athula Balachandran[3]   Vyas Sekar[4]

Gianluca Iannaccone[5]   Srinivasan Seshan[3]

ABSTRACT

This paper presents the design of a novel architecture called CARE (Content-Aware Redundancy Elimination) that enables maximizing the informational value that challenged networks offer their users. We focus on emerging applications for situational awareness in disaster affected regions. Motivated by advances in computer vision algorithms, we propose to incorporate image similarity detection algorithms in the forwarding path of these networks. The purpose is to handle the large generation of redundant content. We outline the many issues involved in such a vision. With a Delay-Tolerant Network (DTN) setup, our simulations demonstrate that CARE can substantially boost the number of unique messages that escape the disaster zone, and it can also deliver them faster. These benefits are achieved despite the energy overhead needed by the similarity detectors.

Categories and Subject Descriptors: C.2.2 Network Protocols: Routing protocols

Keywords: Challenged Networks, Informational Value, Network Architecture, Image Similarity.

1. INTRODUCTION

Network infrastructures have evolved significantly in the last decades to provide high bandwidth, reliable, and highly available infrastructures that form the foundation of many everyday activities. Nevertheless, we have witnessed several real-world events and scenarios – natural disasters (earthquakes, fires, tornadoes), military on-field deployments, flash crowd events (e.g., SuperBowl) – where such foundational assumptions about connectivity, throughput, and reliability are shattered [9, 22]. While these scenarios may differ in

1) Technicolor, 2) University of Illinois at Urbana-Champaign, 3) CMU, 4) Stony Brook University, 5) Red Bow Labs

their spatial and temporal magnitudes, a common characteristic is that the communication infrastructure is intrinsically damaged or limited in capacity; we refer to these collectively as *challenged networks*.

In parallel, we see a technology trend where everyday computing devices, such as smartphones, tablets and laptops have made it extraordinarily easy for people to generate a great deal of content, such as photos and videos. Furthermore, evidence from user studies [13] shows that the nature of events that result in challenged networks often causes anxious users to generate even more redundant content then usual, exactly at a time when the communication infrastructure is damaged.

Such challenging scenarios require us to radically rethink the role that the communication infrastructure needs to serve. We argue that in such challenged environments the fundamental role of a network is not to optimize any specific performance or availability metric (e.g., bandwidth or latency). Rather, the goal of the network should be to maximize the *informational value* that it offers its participants. Having thus recast the role of the network, we need to revisit abstractions for the different network entities such as hosts and routers.

Our observation here is that this informational value ultimately relates to what the users really care about rather than the bits that the network delivers. Consequently, we believe that we need to look to other domains of computer science (e.g., machine learning, NLP, computer vision) that allow us to obtain a *semantic understanding* of the text, image, and video content generated by users in challenged networks. Furthermore, by integrating these as first-order mechanisms in our networking "stacks", we can de-prioritize or eliminate semantically redundant content to optimize the amount of *useful* information delivered. We refer to this as the CARE (Content-Aware Redundancy Elimination) paradigm.

As a specific example of such a challenged network where users can generate redundant content, we focus in this paper on photos taken within a disaster. Specifically, we observe a recent upsurge in the use of online social networks, as well as the emergence of new web applications, that aggregate data (e.g., flood levels, location of help, wounded victims, etc.) and share it via annotated maps [1, 11, 12]. These can both assist victims and enable ordinary citizens to involve them-

selves in the emergency response effort. These emerging *situational awareness* (SA) applications assume that those inside the disaster zone can actually connect to the Internet. All too often this is not the case because the usual communication infrastructure can be compromised by the disaster [5, 4]. This motivating application helps us to start understanding the challenges involved in realizing our CARE paradigm idea in practice, namely the tradeoffs, and the scenarios under which this paradigm brings benefits.

Our focus on photos enables us to leverage the many advances in the field of computer vision over the last decade, specifically methods that accurately identify whether two images are similar based on their content [18, 20, 19]. We propose the idea of using image similarity detectors in the forwarding path of networks. We believe that this paradigm has broad uses in challenged networks and applications such as participatory sensing, massive photo sharing, storage and syncing. Furthermore, the advent of software routers may even offer a plausible roadmap for implementing such algorithms in routers.

A related rethinking of network abstractions was the effort in attempting to name *content* [14, 16] and perceptual content [6]. While the broader utility of such mechanisms in an Internet-wide context is questionable [10], we believe that the challenged networking scenarios we consider offer a plausible, even if niche, application scenario where content naming has value. That said, we find that existing naming abstractions are still quite rigid; they provide a "binary" notion of whether the content is *identical* or not, be it at the byte-level [14] or multimedia-signal level [6]. We make a case for a more flexible abstraction that ties into the notion of informational value to gracefully handle the limited host and network resources.

In summary, this paper makes the following contributions:

- We propose the CARE paradigm to maximize the informational value delivered in challenged networks, present a detailed discussion of issues and tradeoffs that rise, and discuss its operating regimes.

- As a concrete example, we propose to integrate image similarity detection algorithms into the forwarding path of Delay Tolerant Networks (DTN) to enable content-aware traffic reduction. Our architecture augments a DTN stack with a capability that is compatible with existing DTN protocols.

- We present a preliminary exploration of these issues using a DTN simulator, and study a sample disaster scenario. We show that our system enables three existing DTN protocols to dramatically increase the number of unique photos that ultimately escape the disaster zone, and reduce their delivery times.

2. DATA

We use two real-world image datasets for quantifying redundancy, and study semantic redundancy vs. byte-level compression techniques:

San Diego fire (SDfire): Contains 84 pictures taken by a professional photographer who wandered around one of the affected towns both during the fire event in 2007 and afterwards.The pictures depict a variety of scenes including burning homes, damaged homes and cars, firefighters, policemen, etc. This dataset serves as an example to the data and redundancy that could be generated by a single person.

Haiti earthquake: Contains 415 pictures taken during and after the Haiti earthquake in January 2010 by the volunteers of a medical assistance organization called Team Rubicon [2]. The photos cover a wide range of subjects including wounded people, damaged buildings, vehicles, etc., captured by a team of roughly 10 people.

2.1 Quantifying Redundancy

To objectively evaluate the extent of redundancy in our data and to evaluate how well similarity detection performs, we need a notion of ground truth regarding the similarity of these images. We consulted experts who have experience in disaster events to help us with manual labeling of the data. We worked with members of our city's Disaster and Fire Commission, and the local Amateur Ham Radio Club . We built a tool that allows the labeler to view 2 images side by side and rate the similarity on a scale from 0 (entirely dissimilar) to 5 (most similar). After our consultants labeled the data, we mark two images as similar if the average score of all labelers is over 3.

To quantify the amount of redundancy in our datasets, we use the notion of a minimum set cover, which quantifies the minimum number of non-similar photos that need to escape the disaster zone in order to cover all of the information about events within it. Using this definition and our labeled data, we find that the redundancy in *SDfire* and *Haiti* datasets is 53% and 22%, respectively. This means, for example, that by sending less than half the photos in the *SDfire* data, we can convey all of the unique information.

2.2 Do Byte-Level Methods Suffice?

A natural question is whether existing byte-level compression methods can capture the redundancy in these datasets. We consider two methods – gzip-based compression and chunk-level compression [21, 7]. For gzip, since each image is already in a compressed encoding, we consider compressing the entire set of images (i.e., tar + gzip), and use the default window of 15 bits. For the chunk-based compression, we vary the chunk size to explore the tradeoff between chunk size and redundancy.

Method	Haiti	SDfire
tar+gzip	7.7%	5%
Chunk-based, 64B	2.2%	0.9%
Chunk-based, 512B	0.67%	0.04%
Chunk-based, 2KB	0.6%	0%
Ideal/Content-Aware	22%	53%

Table 1: Redundancy elimination gained using different methods on the Haiti and SDfire datasets.

Table 1 shows that byte-level approaches cannot capture the redundancy in the two image datasets from disaster scenarios – compression technique fails because images in JPEG format are already compressed, and byte-level fails because small photometric differences may result in significantly different byte-level encoding.

3. CARE ARCHITECTURE

One typical networking scenario that can occur in a disaster affected region is depicted in Fig. 1. Here, the network infrastructure has been damaged and users inside the disaster area can only communicate with each other via ad hoc or pairwise contact opportunities. A rescue vehicle intermittently visits the disaster area, and any node that comes in contact with this vehicle has the opportunity to transfer messages to it. The rescue vehicle can transport messages outside of the disaster area and deliver them to the public Internet. We assume that a variety of devices (cellular phones, smartphones, laptops) will be present in the disaster area. The devices will have a heterogeneous set of network capabilities (WiFi, bluetooth), compute power (ranging from low power embedded devices to multi-core processors), storage capacity (from the 100s of Mbytes of a regular phone to the 100s of GBytes of a laptop) and battery life (from a few hours to days). In practice, this means that nodes can establish point-to-point communications, can store large amounts of information for each other, and have a good amount of compute power to pre-process the data they receive.
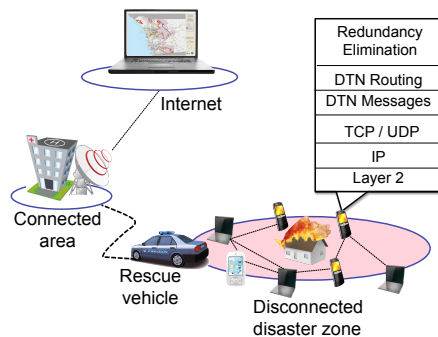

Figure 1: Example of a typical disaster scenario.

We propose a software architecture in which devices, in absence of network infrastructure, can enable a DTN stack – a sort of *disaster mode* for phones and laptops. A DTN stack gives us mechanisms to discover neighboring nodes, identify the available communication media, and to package, store and carry messages of others. In order to handle the compounding affects of image redundancy coupled with limited opportunities to transmit data outside the disaster region, we propose to use the message content to drive forwarding decisions. We thus propose a system called CARE (Content Aware Redundancy Elimination) which augments the DTN stack by adding a capability that can be incorporated on top of any DTN routing protocol. Fig. 1 illustrates

this idea in which images in a message buffer are evaluated for similarity before they are assembled into a DTN bundle and transmitted.

The DTN stack leaves open the choice of routing protocol, and while many protocols exist, we focus on a few representative algorithms. Epidemic routing [24] simply replicates messages on each contact of two nodes. Prophet [17] only forwards messages towards a node if it believes the next hop increases the chances of final delivery of the message, using their history as an indicator for likelihood of reaching the destination. Spray and Wait [3] is a quota-based approach for replication that limits the maximum number of copies of a message in the network. All of these protocols replicate messages in the hopes that one of the copies will succeed in reaching the destination, while seeking to control the number of replicas of any given message. Our proposal is complementary to these since we seek to control replicas that arise because two different messages are similar, hence we propose adding it on top of any DTN protocol.

There are two chances to carry our content-aware redundancy elimination. The first is local redundancy elimination; when the same user takes multiple photos of the same scene, we detect image similarity within the user's buffer and deprioritize copies before a transmission opportunity presents itself. The second is when two nodes meet. Before exchanging buffer contents, the two nodes swap lists of photo ids, and they only transfer those which the other does not have. We assign order to the exchange, so that after the first node receives the new photos from the second, it performs the redundancy tests locally. Once completed, it only sends the non-redundant photos to the second node, alongside with the updated ids. This process incurs more transmissions, however they are very small and potentially save both computation and transmission of photos, which are much more energy consuming.

3.1 Image Similarity Metrics

We experimented with several well-studied image similarity methods in order to explore the balance between accuracy and performance, and suggest a hybrid method that tries to leverage the advantages of each while minimizing their drawbacks.

Scale-invariant feature transform (SIFT) [18] is an algorithm that finds and extracts features that are invariant to image translation, scaling, and rotation, and some illumination changes. This method is widely used in object recognition and image stitching applications due to its accuracy, but has a relatively high complexity.

A perceptual hash (pHash) [20] is an easy-to-compute fingerprint of a file derived from the content, having the property that two similar images result in similar hashes.

GIST [19] is a method used for scene detection using perceptual dimensions, such as openness and roughness to capture the dominant spatial structure of a scene. It has the potential to be useful in disasters involving fires, floods, or tornados, since photos from such events could be of scenes with

amorphous elements (such as a spreading fire) that don't have clearly well defined edges (something that is important for methods such as SIFT).

Each of these methods individually has a threshold parameter that determines their FP and FN. We studied the accuracy of the algorithms against our ground truth labels (from Sec. 2), and found that SIFT performed best (FN of 0.48 with FP of 0.01). GIST resulted in a low FN of 0.30 but a relatively high FP of 0.15. pHash was accurate only when the photos were very similar, resulting in a very high FN of 0.83 but a low FP of 0.01.

Next, we ran the algorithms on two different machines, 1.6 GHz 2 GBytes RAM Atom-based and an 8-core 2.8 GHz 12 GBytesRAM Xeon-based. We found that on both platforms SIFT is the most expensive algorithm to run with an average execution time (over all the photos in our two datasets) being $150\times$ slower than that of pHash while GIST is $50\times$ slower than that of pHash.

Given the tradeoff between accuracy and computational overhead, we can construct a hybrid method that leverages the advantages of each algorithm, while enabling low computation complexity. For example, we start with pHash, and use it only to identify whether two images are similar. Since it has a low FN rate, when pHash identifies that the images are not simialr, we defer it to a more accurate but costly method, such as SIFT.

We note that our design can accommodate additional algorithms. For example, a trivial meta-data filter can flag photos that are taken too far apart (either temporal or spatial distance) as not similar. Additionally, there are other methods for similarity between images that can be considered, ranging from simple color-histograms [23] to complicated face and object recognition. We expect that such methods can further improve the accuracy of the similarity detection, but also come at the cost of computation overhead.

3.2 Is all of this worth doing?

The idea of eliminating redundant images is intuitively appealing. However putting this in the forwarding path incurs certain costs and the question as to whether or not this is worth doing will depend upon the details of the disaster scenario, overhead, energy, accuracy, and so on. There are key research efforts needed to evaluate the implications of such a proposal and the complex tradeoffs it induces, in order to be able to answer the question "under what conditions does this make sense?". We focus on two key performance metrics for CARE. The *coverage* is defined as the number of unique messages delivered to the SA application, and we seek to increase the coverage as compared to traditional DTN forwarding. Our second metric is the latency with which these unique messages are delivered. If these images contain life-critical information (e.g., about where fire protection or medical help is needed) then early delivery could save lives or limit damage. We now discuss the costs and tradeoffs that impact the performance of our system, and we describe what's needed to understand these factors.

Overhead: Computer vision algorithms for image similarity detection create computational overhead that will use up energy. On the one hand, given that oftentimes a device's battery lifetime is less than the duration of the disaster event, the energy used by CARE will cause it to run out of battery sooner which in turn impacts coverage. On the other hand, energy is saved with CARE because redundant images are not transmitted, and saving on transmissions could prolong device lifetime. A broad range experiments are needed to understand this tradeoff, and to quantify what level of overhead we can incur while still enjoying the benefits of CARE.

Imperfection in similarity detection: All image similarity detectors can make mistakes. A false positive (FP) occurs when two images are identified as similar, but in fact are not. This is a bad mistake to make, from the point of view of disaster recovery, because it means that a unique image will get dropped. Thus FPs reduce the improvements in coverage that CARE typically brings. A false negative (FN) occurs when the algorithm fails to realize that two images are similar; the impact of this mistake is only that an opportunity for traffic reduction is lost. When unnecessary messages are forwarded, the latency of delivering unique messages could increase, thereby lowering the usual latency gains of CARE. These tradeoffs imply that some image similarity detectors may not be worth using if their FP or FN rates are high enough as to undo the usual benefits that CARE brings.

Connectivity Opportunities: At one extreme, when connectivity opportunities are plentiful, there is no real need for CARE (other than perhaps saving some energy) because all content can be delivered. At the other extreme of almost no contacts, nothing can be delivered anyway, regardless of the protocol. CARE will be useful in a range between these two extremes, a range which we plan to explore and make sure it is large enough.

Redundancy: We anticipate that more redundancy brings greater benefits, although the rate at which benefits increase with growing redundancy needs to be quantified.

4. DISASTER SCENARIO SIMULATION

We simulate a disaster event in a way that enables us to vary the many parameters in our system, so that we can understand the range of settings in which CARE brings benefits and to understand tradeoffs. We used the Opportunistic Network Environment (ONE) [15] simulator, a DTN simulator, and added an implementation of CARE on top of epidemic routing, Prophet and Spray-and-Wait. We present some initial and promising results for a sample disaster scenario.

4.1 Scenario Settings

We simulate a 12 hour-long disaster scenario in a neighborhood in Pittsburgh, USA, a city that is included in the ONE simulator, with a detailed map of all the roads and bus routes. The covered area is roughly of 10×8 miles. We consider a scenario with 60 people randomly located inside the disaster area, in which there are 3 designated "hot
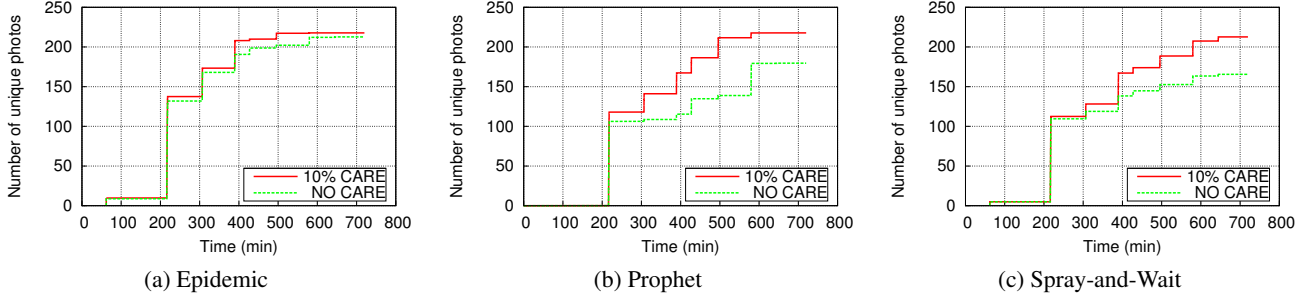
Figure 2: Number of uniquely delivered photos over simulation time using 50% redundancy

zones" from which photos are important to the rescuers – 2 small zones of $0.4\ mi^2$ each, and a larger zone of roughly $2.5\ mi^2$. A single rescue vehicle travels back and forth between the disaster area and a communication gateway (e.g., satellite) that is located 6 miles outside the disaster area and is connected to the public Internet. The rescue vehicle drives at a speed of 25–54 Km/hr and spends roughly the same amount of time in the disaster zone as outside it. If a person gets within 20 meters of the rescue vehicle, it can upload its buffer contents (if there is enough remaining power). Contact opportunities with this vehicle are the only way data from inside the disaster zone can reach the public Internet.

People walk at a speed uniformly distributed in the range of 3–7 Km/hr. They move in a point-of-interest map-based shortest path traversal, meaning each person selects a destination point inside the disaster area and finds the shortest path using roads to get there. Once the person reaches a destination it stops for 5 minutes, and then repeats this process to walk to another location.

At any given point in time, we assume that each hot-zone has 5 unique scenes that are of interest to people that traverse it. Whenever a person enters a hot-zone, she takes up to 5 (selected randomly) redundant photos of a scene (chosen randomly from the 5 possible scenes), each photo is of size 300KB. Different people that traverse the hot-zone can possibly take photos the same scene, resulting in cross-node redundancy. However, photos from different hot-zones and photos taken more than 15 minutes apart are never redundant. We further assume that each person will not take more than 50 photos from the same hot zone over the entire experiment lifetime.

Each person has a WiFi-enabled device, that transmits at 10 Mbps with a maximal range of 20 meters, simulating a smartphone. It has a limited buffer of 100MB used for storing photos, and a limited energy, enabling at most E_{max} hours of operation without any transmissions. When transmitting, the device consumes 2 Jouls/sec of the battery [8], which is roughly 0.015% of the battery capacity when setting $E_{max} = 12h$. The overhead of using CARE is the cost of comparing photos plus the cost of comparing meta-information regarding the photos that exist in each person's buffer. We assume the energy consumed by this overhead, on a per photo basis, is a percentage of the standard transmission energy consumed for a packet. We set this value to 10% of the transmission cost, and plan to further analyse the effect of the overhead in future work.

For Spray-and-Wait we set the quota to 12 (20% of the number of people), and for Prophet we set update interval to $60s$ and the probability computation parameters to $\alpha = 0.25$ and $\beta = 0.98$ (these parameters set the update rates of the router's probability tables [17]).

4.2 Results

We present results for the main performance metrics, namely the ability to deliver more unique messages, and to achieve lower latency, while being robust to the overhead needed by CARE. The settings we used resulted in an overall redundancy of 50% with roughly 216 unique photos. This corresponds to the redundancy that exists in our Haiti dataset. We note that this is quite a low estimation of the redundancy we expect to find in a real-world scenario.

Fig. 2 shows the number of unique messages that reach the communication gateway with and without CARE for the different routing protocols. The stairs in the plots are exchanges of messages between the rescue vehicle and the communication gateway.

The figures clearly show that all protocols benefit from adding the CARE layer, with Spray-and-Wait managing to deliver almost 30% more unique photos with CARE than without it, and Prophet managing to deliver 21% more photos. The improvement in coverage is smaller for the Epidemic protocol (5%), because there is enough bandwidth and energy for the nodes to push numerous messages through, including both unique ones and replicas. However, Epidemic with CARE manages to deliver all photos almost an hour before Epidemic without CARE.

Furthermore, we repeated these simulations with 85% redundancy and found that Epidemic with CARE delivered nearly 14% more unique photos than without CARE. The reason for this is that as the number of photos in the system increases, the flooding behavior of Epidemic routing sends

Time	CARE			Non-CARE		
	Epidemic	Prophet	S&W	Epidemic	Prophet	S&W
1h	36	13	17	25	9	14
2h	121	57	61	97	35	49
4h	**216**	201	173	207	128	140
6h	216	**216**	201	**211**	159	157
8h	216	216	210	211	176	163
10h	216	216	**211**	211	**178**	**164**
12h	216	216	211	211	178	164

Table 2: Latency of delivered photos, showing the number of unique photos delivered within each given delay

too many replicas during those few contact opportunities, thereby impeding delivery of unique photos. Prophet exhibited the largest improvement having 73% more delivered photos with CARE than without it. Overall, these show that as the redundancy increases, CARE better controls the flow of unique information.

To understand the latency improvement obtained by adding CARE, Table 2 shows the number of unique photos delivered within the first x hours. CARE enables all protocols to deliver the photos earlier than they would be operating without CARE. For example, within 6 hours, Prophet can deliver 216 with CARE, but only 159 without it. Coupled with the CARE mechanism, Epidemic and Prophet manage to deliver all 216 messages, while Spray-and-Wait delivered nearly all. Without CARE, none of the protocols manage to deliver all unique photos.

Finally, in order to evaluate the robustness of these improvements to the energy overhead incurred by the image processing algorithms, we performed the same simulations with the CARE overhead set to 50%. We found that even when more energy is consumed by image comparison, the number of photo transmissions is reduced by up-to 60% when CARE is used, which in turn saves energy. The balance between these energy-draining and energy-saving functions was such that we still saw roughly the same improvements (5 − 30%) in CARE as with the 10% overhead. These results are encouraging since they indicate that we can use sophisticated algorithms such as SIFT, rather than the cheaper but less-accurate pHash and GIST.

5. CONCLUSION

We propose a method for maximizing the informational value that users obtain from challenged networks by allowing contextual similarity to be used as a prioritization method in the forwarding path.

Our focus on photos in disaster networks is motivated by the confluence of multiple trends, namely disaster response, image processing algorithms, new networking paradigms and powerful handheld devices. We believe that the use of the Web to provide SA during disasters is a trend that will continue to gain momentum – especially since the Internet has already proven effective in disaster response. This makes such scenarios a great motivating example for our proposed networking paradigm.

Image processing is rapidly moving forward with better algorithms and hardware support, and thus the approach we propose here is likely to become even more viable in the near future. It is important to understand solutions for detecting image redundancy, because the next step will be to extend such ecosystems to support redundancy detection in videos. In our ongoing work, we are flushing out all the tradeoffs occurring in such systems, and we seek to understand the network operating regimes for which CARE brings significant benefits.

6. REFERENCES

[1] Open Source Software for Information Collection, Visualization and Interactive Mapping. http://www.ushahidi.com.
[2] Team Rubicon. http://teamrubiconusa.org.
[3] Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM SIGCOMM Workshop on Delay Tolerant Networking*, 2005.
[4] Google Crisis Response: Haiti Report. http://www.google.org/docs/Haiti.pdf, 2010.
[5] M. Allman. On Building Special-Purpose Social Networks for Emergency Communication. In *ACM CCR*, 2010.
[6] A. Anand, A. Akella, V. Sekar, and S. Seshan. A case for information-bound referencing. In *HotNets*, 2010.
[7] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker. Packet caches on routers: The implications of universal redundant traffic elimination. In *ACM SIGCOMM '08*.
[8] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *IMC '09*.
[9] J. Cowie, A. Popescu, and T. Underwood. Impact of Hurricane Katrina on Internet Infrastructure. Renesys, 2005.
[10] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Information-centric networking: seeing the forest for the trees. In *HotNets*, 2011.
[11] S. Gupta and C. Knoblock. Building Geospatial Mashups to Visualize Information for Crisis Management. In *ISCRAM*, 2010.
[12] A. Hughes and L. Palen. Twitter Adoption and Use in Mass Convergence and Emergency Events. In *ISCRAM*, 2009.
[13] A. Hughes, L. Palen, J. Sutton, S. Liu, and S. Vieweg. Site-seeing in disaster: An examination of on-line social convergence. In *ISCRAM '08*.
[14] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *CoNEXT '09*.
[15] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *ICST SIMUTools*, 2009.
[16] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *SIGCOMM '07*.
[17] A. Lindgren and A. Doria. Probabilistic Routing Protocol for Intermittently Connected Networks. *Internet Draft - http://tools.ietf.org/html/draft-irtf-dtnrg-prophet*, February 2010.
[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Journal of Computer Vision*, 60, 2004.
[19] Matthijs Douze, Hervé Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *CIVR*. ACM, 2009.
[20] M. K. Mihçak and R. Venkatesan. New iterative geometric methods for robust perceptual image hashing. In *Security and Privacy in Digital Rights Management, Springer '02*.
[21] A. Muthitacharoen, B. Chen, and D. Mazières. A low-bandwidth network file system. In *SOSP '01*.
[22] A. Townsend and M. Moss. Telecommunications Infrastructure in Disasters: Preparing Cities for Crisis Communications. 2005.
[23] M. Y. S. Uddin, H. Wang, F. Saremi, G.-J. Qi, T. Abdelzaher, and T. Huang. Photonet: A similarity-aware picture delivery service for situation awareness. In *RTSS '11*, 2011.
[24] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Duke University, 2000.