Network Coding for Content-Based Intermittently Connected Emergency Networks

Joshua Joy, Yu-Ting Yu, Mario Gerla
UCLA
{jjoy,yutingyu,gerla}@cs.ucla.edu

Samuel Wood, James Mathewson
Suns-Tech
{sam,james}@suns-tech.com

Mark-Oliver Stehr
SRI International
{stehr}@csl.sri.com

ABSTRACT

First responders at the edge of the network rely on situation awareness updates to arrive in a timely matter, even when the fixed infrastructure is unavailable. The technical advancements of the commercial mobile phones make them capable of supporting such requirements under very disruptive network conditions. In this demo, we present a network architecture that exploits partial caches by utilizing network coding to deliver large files (e.g. images) to first responders. The architecture is based on a content centric network platform called ICEMAN(Information CEntric Mobile Ad-hoc Networking) and runs on Android phones. We demonstrate our system in a file dissemination scenario in a CORE/EMANE network emulation. We measure file delivery ratio, latency, and network overhead and report significant improvements that network coding achieves over fragmentation.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Distributed networks

General Terms

Design, Performance, Reliability

Keywords

Network Coding, Ad-Hoc, MANET, ICN

1. INTRODUCTION

Tactical and emergency response scenarios require efficient, robust, and secure network communications to quickly deliver data for situational awareness applications. The dynamic and resource limited constraints in these networks require that nodes opportunistically communicate with limited global knowledge, and make efficient use of the scarce available bandwidth. However, despite the remarkable increase in sensing, storage, processing, and communication capabilities in mobile devices, efficient dissemination and storage of content on the volatile network edge remains a challenge. In this study we fragment a file into segments, called fragments. Then, before transmission we explore two options. In the fragmentation option, we transmit fragments as independent entities. The destination is responsible for reassembly. In the Network Coding option, blocks are linearly combined and coded as later discussed. The destination decodes and reassembles. We show that Network Coding helps to improve the number of files delivered compared to fragmentation in scenarios with intermittent connectivity and severe disruptions.

MobiCom'13, September 30–October 4, 2013, Miami, FL, USA.
ACM 978-1-4503-1999-7/13/09.
http://dx.doi.org/10.1145/2500423.2505291 .

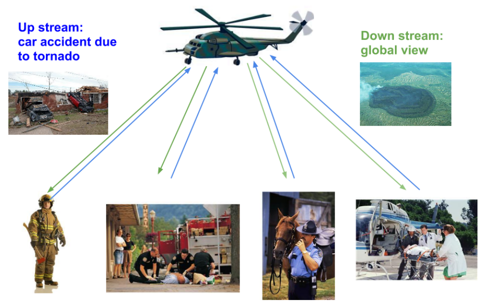

Figure 1: First responders in an emergency situation share location and images with each other. Overhead, a helicopter circles transmitting the global view down to the first responders. Simultaneously, the first responders upload to the helicopter their local view of the situation. Due to the intermittent connectivity, only partial images are often received, cached and shared. The image file completion may require repeated contacts between ground teams with each other (exchanging the partial caches) and with the helicopter [1, 2]

More specifically, due to intermittent connectivity, we face the following challenges during file transfer and dissemination:

- *Last coupon problem*: Network disconnections or partitioning may frequently occur, thus, requiring a file to be transmitted in a piecemeal fashion, possibly with pieces arriving out of order. Missing pieces make it difficult for a receiver to reliably reconstruct a file.
- *End-to-end connectivity not guaranteed*: Nodes must store and cache partial files; then wait for the next contact opportunity to resume transmission.
- *Partial caches*: Various caches may contain different pieces of a file. A receiver does not know which caches contain which pieces of the file in advance.

- *Busy caches*: If a requestor does not receive a response from a cache that contains the missing segments, it may turn out that the cache is busy serving other requestors. Thus, the requestor must wait for this or another cache to respond.

Network coding can help overcome these challenges and improves file delivery in scenarios with severe disruptions. We show that network coding with a carefully designed content-based network can address each of the above challenges. In particular:

- *Eliminate Last Coupon Problem*: By using network coding, the last coupon problem (also known as the "coupon collector's problem", i.e. waiting for the last missing piece) is mitigated since each coded block received is innovative (i.e., helpful) with a high probability and helps filling in the gaps. Thus, even if network partitions occur, more files will be delivered with coding compared to not coding.

- *Overcoming Intermittent Connectivity*: Since end to end connectivity is not guaranteed and transmissions are connectionless, we cache blocks at intermediate nodes. A requestor can then exploit nearby caches for network-coded blocks. The neighbors push coded blocks from their cache.

- *Exploit Partial Caches*: Nodes cache partial files with innovative blocks. Since each coded block is as helpful as any another, nodes can make efficient use of intermittent contacts by transmitting arbitrary innovative blocks at the next contact and transmission opportunity.

- *Take Advantage of Peer Caches*: When a requestor does not receive a response from a nearby cache because it is too busy to answer requests, it can exploit other nearby caches to collect blocks, since each network-coded block is as good as any other.

2. ICEMAN ARCHITECTURE

ICEMAN (Information CEntric Mobile Ad-hoc Networking) [11, 12] enables flooding of interests and epidemic dissemination of interests and content. ICEMAN is implemented in the Haggle framework [5, 6, 9, 10]. Haggle is an open-source content-centric networking architecture originally aiming at pocket-switched networks, a particular form of delay-tolerant networks (DTN) that use a store and forward paradigm. Several extensions and significant changes where made to the Haggle framework to better support MANETs and DTNs with richer connectivity as well as message ferrying. An extensive overview of ICEMAN with implementation details and performance reports can be found in [11, 12].

3. NETWORK CODING REVIEW

We first begin with a quick overview of network coding [7, 4]. We start with a *source node* that wishes to disseminate a file F. The source node first subdivides F into a set of fragments. The fragments are then coded into a set of m vectors $\mathbf{v_1}, ..., \mathbf{v_m}$ in an n-dimensional vector space over a finite field $\mathbb{GF}_p$ where p is a prime number. These vectors are then linearly combined by drawing from the finite field $\mathbb{GF}_p$ an encoding coefficient $\mathbf{e_i}$ to create m coded blocks $\mathbf{b_1}, ..., \mathbf{b_m}$. The set of coefficients used to create the vector is called the encoding vector $\mathbf{e}$ which is $[\mathbf{e_1}, ...\mathbf{e_n}]$.

To reconstruct the file, a node must recover enough linearly independent coded blocks to be able to perform matrix inversion. Formally, we take the transpose of the received vectors such that $\mathbf{E^T} = [\mathbf{e^T{}_1}, ..., \mathbf{e^T{}_n}]$, $\mathbf{B^T} = [\mathbf{b^T{}_1}, ..., \mathbf{b^T{}_n}]$ and $\mathbf{V^T} = [\mathbf{v^T{}_1}, ..., \mathbf{v^T{}_n}]$. Then we compute $\mathbf{E}^{-1}\mathbf{B}$ which is the reconstruction of all the original blocks in the file. In practice, Gaussian Elimination is used to speed up the decoding process [7, 4].

4. NETWORK CODING IMPLEMENTATION

To improve the delivery of content, ICEMAN supports fragmentation and network coding. We segment a large file into fragments (in case of fragmentation) or blocks (in the case of network coding). This overcomes the problem arising when file transmissions are atomic and intermittent connectivity causes repeated unsuccessful attempts to retransmit the entire file. Inspired by CodeTorrent [8], we exploit that peer nodes can become seeds. This is the case after the peer has accumulated enough innovative network-coded blocks and has reconstructed the original file. These seeds then serve as a source of new innovative blocks. Network coding and fragmentation differ in a key aspect: With network coding the source and seed nodes generate innovative blocks with high probability, while fragmentation can select only from a finite set of fragments (the same set for sources and caches). This capability allows network coding to efficiently solve the "coupon collector's problem", while the fragmentation approach has to wait precisely for the missing fragments.

In order to study the effects of partial caches, we apply a simple LRU caching policy for collecting blocks (or fragments) and for aging and timing out these blocks (or fragments). Additionally, after a seed successfully reconstructs a file using blocks or fragments, the blocks used for reconstruction are aged after inactivity. Blocks which are generated by the source or a seed are kept in the LRU cache for a defined period of time and are then expired.

5. DEMO DESCRIPTION

With reference to the scenario in Figure 1, we will demonstrate that with intermittent connectivity, fragmentation typically requires multiple helicopter passes while network coding can reconstruct the published original file within a single pass. The helicopter is an example of a highly mobile node, which could also be a UAV or a ground vehicle. The system used in the demo, ICEMAN, is a fully functional peer-to-peer information-centric architecture running on Android phones and utilizing UDP broadcast for the dissemination of interest and content. Figure 1 describes the demo scenario. Because of intermittent connectivity only partial images (i.e. fragments or blocks) can be directly shared in both directions between helicopter and first responders arriving at the aftermath of the emergency (say, a tornado).

The demo will consist of an emulated scenario utilizing the CORE emulator [3] in which network coding and fragmentation run side by side to download the files. Attendees will be able to visually appreciate that network coding is the first to complete the downloads all the files first, while fragmentation experiences significant download delay due to the ”coupon collector’s problem”. One phone in each group acts as the helicopter, while the remaining three phones act as the first responders with actual ICEMAN code running in emulation mode. The network connectivity is emulated by CORE.

The demo will be using ad-hoc WiFi mode on the Android phones, so no additional wireless access is required. Space is required for a laptop and eight Android phones.

6. EVALUATION

In this section we report typical measurements collected in previous experiments. To isolate the effects of coding from fragmentation, we evaluate each strategy individually. We implement network coding above the transport layer; e.g. network coding occurs at the application layer for both TCP and UDP broadcast. Naturally, TCP handles congestion control while UDP does not. Moreover, when utilizing UDP broadcast, RTS and CTS is disabled at the MAC layer.

In Figure 3, we evaluate a thirty node scenario using CORE emulation with EMANE 802.11 network simulation. There are three groups of ten member first responder teams each roaming in an open field. Connectivity is sparse. The rescue teams exchange various types of files with each other (stream, data and commands) taking advantage of the times when they are in contact. During a disaster, these images will direct resources and first responders to where they are needed the most. Time can be a critical factor during these rescue situations. The node movement follows the Nomadic Community Mobility model which will represent search and rescue movement. In this model, members of a team perform a random walk around their reference point which in turn follows a Random Waypoint Model. Occasionally teams pass one another and benefit from brief periods of high connectivity.

Figure 3 reports the number of files downloaded for each strategy, within a given delay tolerance. Asymptotic performance is reached as delay grows to infinity. We can observe that network coding outperforms fragmentation (for a given latency) regardless of the underlying transport technology, UDP broadcast or TCP. This is mainly tied to network coding’s ability to overcome the ”coupon collector’s problem”. Comparing different transport techniques, we must recall that this is a file distribution scenario where a file must be broadcast to all neighbors. While TCP provides reliable block transfer on a single link, it cannot compete with the broadcast advantage of UDP. UDP with Network Coding fares particularly well as Network Coding provides the ability to recover from losses, thus matching the reliability of TCP.

7. ACKNOWLEDGMENTS

This work was supported 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. We thank Victor Perez, and Shao-Cheng Lu for their input.

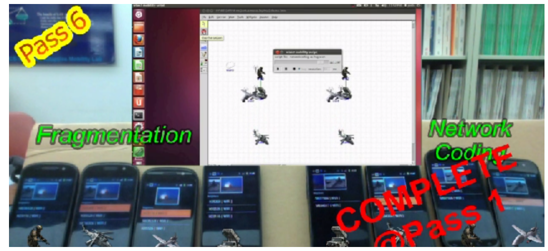

Figure 2: Screenshot capture of demo. Fragmentation (on the left) always requires multiple helicopter passes to reconstruct the file. Network coding (on the right) reassmbles the file in only a single pass.

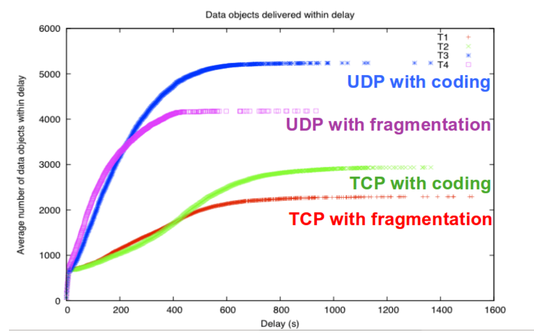

Figure 3: The size of the files range from 250 KB to 1MB. All nodes serve as publishers and subscribers of content.

8. REFERENCES

[1] http://www.nesec.org/images/haz_tornado_car.jpg.

[2] http://www.fire.uni-freiburg.de/other_rep/research/rus/rus_re_1bor_16.jpg.

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

[4] C. Fragouli, J.-Y. Le Boudec, and J. Widmer. Network coding: an instant primer. *ACM SIGCOMM Computer Communication Review*, 36(1):63–68, 2006.

[5] Haggle - A content-centric network architecture for opportunistic communication. https://code.google.com/p/haggle/.

[6] Haggle: An innovative Paradigm for Autonomic Opportunistic Communication Research project funded by the European Commission under the Information Society Technologies Programme of the 6th Framework between January 2006 and June 2010. http://www.haggleproject.org/.

[7] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413–4430, 2006.

[8] 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*, pages 1–5. ACM, 2006.

[9] E. Nordstrom, P. Gunningberg, and C. Rohner. A search-based network architecture for mobile devices. Technical report, Uppsala University, 2009.

[10] E. Nördstrom, P. Gunningberg, and C. Rohner. A search-based network architecture for mobile devices. *TR, Department of Information Technology, Uppsala University*, 2009.

[11] S. Wood, J. Mathewson, J. Joy, M.-O. Stehr, M. Kim, A. Gehani, M. Gerla, H. Sadjadpour, and J. Garcia-Luna-Aceves. ICEMAN: A practical architecture for situational awareness at the network edge. http://www.csl.sri.com/users/stehr/CBMEN/iceman-arch-2013.pdf, 2013.

[12] 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. http://www.csl.sri.com/users/stehr/CBMEN/iceman-eval-2013.pdf, 2013.