Endrun—Secure Digital Communications For Our Modern Dystopia*

Grant Dobbe[†] and Brendan O'Connor[‡]

October 16, 2014

The Internet is no longer trustworthy, having been compromised by bad actors across the globe. Current proposals to work around a compromised Internet rely upon encrypted transport links, mesh networks, or harassing users for being unable to use GPG safely. Each of these strategies fails in different ways that inevitably lead to information leakage or—in the extreme case—death. Endrun, by contrast, takes NASA's Disruption-Tolerant Networking project from a laboratory experiment to a functional system that supports user-friendly encryption in hostile environments. Endrun embraces the nearly-unlimited throughput of a disk-laden station wagon and creates a reliable, eventually-consistent communications system ideal for activists, refugees, and trolls.

1 Introduction

It's easy to speak when everyone loves you. When people in the US or the rest of the western world discuss "free speech," they generally mean some form of the right to communicate with others, and to assemble with others for the purpose of sharing their message. Some may also refer to a right to anonymous speech, but there are those who would attack the right of anonymity; "if you have nothing to hide, you have nothing to fear."[1] In an era when the catchphrase is "Privacy is dead, get over it"[2], it's easy to

*This paper, and the work it describes, represents the work of the authors in their spare time, and represents neither the product, nor the policy, of either author's employer(s). Conclusions of law in this work are not "legal advice," do not establish an attorney-client relationship, and may be neither sane nor coherent.

[†] Lead Security Engineer and Cloud Services Meteorologist, NuCivic Inc.

[‡] Senior Security Consultant, Leviathan Security Group

[1] For an excellent response to this common fallacy, see: Solove, Daniel. "Why Privacy Matters Even If You Have 'Nothing to Hide' " The Chronicle of Higher Education. May 15, 2011. Accessed September 29, 2014. `https://chronicle.com/article/Why-Privacy-Matters-Even-if/127461/`.

[2] Stated by many, including Sun Microsystems' Scott McNealy; `http://archive.wired.com/politics/law/news/1999/01/17538`

say that "Users who try to use anonymity, or cover themselves up on the Internet, are usually doing things that aren't so-to-speak legal."[3]

We dissent. The right of public speech was not created to protect those messages everyone loves, but those messages that are reviled (for whatever reason). We believe that the current age of "registered bloggers"[4] and believing that press members are only real if they have a "$50,000 camera"[5] reflects a misguided and destructive impulse to control thought and expression for the sake of centralization of power. We therefore present a system that is intended to provide a useful counterbalance; an easy-to-use, strongly-encrypted information-sharing network built not upon a reliable Internet source, but upon the distributed, unending movements of people.

You may stop this individual, but you can't stop us all... after all, we're all alike.[6]

2 Design Goals for Endrun

Endrun has five major design goals.

1. Endrun is designed to make an "end run" around the Internet, for the purpose of giving a communications option to people who cannot use the Internet. This could happen for any of several reasons: the Internet could have suffered a major regional or global failure, due to natural disaster, accident, or hostility; the Internet might be considered untrustworthy, due to adversary action; or the areas being communicated across might be so remote that Internet connectivity is simply not a useful option.[7]

2. Endrun is designed for communications between small groups of people who know each other—"cells" or "pods." It is not designed for open-ended communication between people that have no prior relationship.

3. Endrun is "by any means necessary." Endrun is designed to work across any form of communication that is available, whether that's physical data movement and low-power transient unlicensed radio links, as we would recommend, or by

[3] Attributed to Comcast, who has since denied holding this position. *See*: Cook, James. "Comcast Denies It Will Cut Off Customers Who Use Tor, The Web Browser For Criminals." Business Insider. September 15, 2014. Accessed September 29, 2014. `http://www.businessinsider.com/comcast-threatens-to-cut-off-tor-users-2014-9`

[4] See, for example, Russia's new law. `https://www.techdirt.com/articles/20140423/09130227004/russian-net-clampdown-continues-now-its-turn-blogs-vkontakte.shtml`

[5] Mirkinson, Jack. "Ferguson Police Say They May Arrest More Journalists." The Huffington Post. August 19, 2014. Accessed September 29, 2014. `http://www.huffingtonpost.com/2014/08/19/ferguson-police-arresting-journalists_n_5691645.html`

[6] The Mentor. "The Conscience of a Hacker."" Phrack 1, no. 7 (1986): Phile 3 of 10.

[7] While a large portion of the planet is served at least by bidirectional satellite Internet service, there are still many unserved areas. Even when service exists,it is often overwhelmingly expensive for all but the largest corporations.

Earth-Moon-Earth (EME) high-powered signals from fix points, or even amateur (ham) radio.[8]

4. Endrun is disruption- and delay-tolerant. It is designed to work where nodes have no real- or bounded-time connection to each other, for instance, when couriers are moving data stored on physical media either around town or across a continent.

5. Endrun is built using standard, well-tested, open source components and combined using Python, rather than using all custom code. This lets Endrun benefit from decades of work from thousands of programmers (with all the attendant work on security that implies). It also means that Endrun is customizable by programmers in the field; in an emergency situation, this can give flexibility that fully-custom solutions cannot.

3 Previous Work

There has been a great deal of interest in the area of distributed communication tools, both ones that work over the public Internet, and ones that establish their own wireless links. We will examine several of the largest and most influential projects.

3.1 Tor

Tor[9] is an incredibly-well-known tool for anonymous Internet access. The core security concept of Tor is called "onion routing," where a client picks several nodes to pass the client's message through, and repeatedly encrypts the message with the public key of each node; the message is then routed in turn through each node. This means that if a client picks nodes A, B, and C to transit the message through, in that order, then node A knows where the message originated but not where it is destined (nor its contents), node C knows the message's destination and contents (assuming that no additional encryption is used), but not where it originated, and node B knows neither the message's contents, nor its source, nor its destination.

Tor was originally developed by the U.S. Naval Research Laboratory, subsequently developed by the Defense Advanced Research Projects Agency, and then transferred for ongoing development to The Tor Project, a U.S.-based nonprofit corporation. It is rightly lauded as one of the most important security projects since the invention of networked computing; that said, it has certain deficits which raise concerns for anonymity applications. One of these is that when an adversary is able to monitor both the source and destination networks of a Tor message, a trivial timing analysis will reveal which user sent which message. This was revealed very dramatically through the near-instantaneous

[8] This use of amateur radio would almost certainly be illegal in many countries for several reasons, among them Endrun's use of multiple layers of strong encryption, and the fact that messages may not necessarily be originated by ham radio stations.

[9] `https://www.torproject.org`

reidentification and arrest of Eldo Kim after he used Tor to send a bomb threat to Harvard University, which he attended at the time, apparently to escape a final exam.[10] Another, related, issue is that if an adversary controls a significant percentage of the total Tor network, they will nearly always be able to identify both the source and destination of any message (through a timing analysis of any message that both enters and leaves the Tor network through the adversary's systems). Both of these problems were noted in the original Tor paper.[11] At the time, however, it was believed that increased Tor usage, with an attendant increase in Tor nodes, would render controlling a majority of machines impractical.

Unfortunately, this assumption is now believed to be incorrect. Tor's public metrics[12] indicate the daily average number of nodes is approximately 6,000 (as of September 2014). An adversary could match that number of nodes (and thus control 50% of the total nodes on the network) relatively cheaply; using Amazon Web Services On-Demand Micro Instances, 6,000 nodes would cost just $56,160 per month to operate (plus bandwidth).[13] An adversary with a relatively small datacenter and sufficient bandwidth could spend significantly less on an ongoing basis to compromise the Tor network, and there are allegations that certain entities have already done so. We believe that Tor is now insufficient for truly-suppressed parties to communicate anonymously in the face of a network-level adversary.

3.2 Anonymous AirChat

Airchat[14] was announced to great fanfare on April 23, 2014, promising Internet-free end-to-end encrypted connection suitable for text and low-bandwidth digital voice communication over up to 100 miles using simple radios. Airchat has the dubious distinction of stating an intent to use the system over amateur radio, on which encryption is forbidden (except in certain circumstances), and further stating that "we dont give a fucking shit about prohibitions over the use of encryption. fuck you NSA. [sic]"[15] Setting aside the question of an openly-stated conspiracy to violate U.S. regulations (among many other countries that prohibit encryption on amateur radio frequencies), AirChat seems reasonable. The downside, however, is that it is an online protocol; both sides are communicating continuously, and due to their use of fairly low-throughput proto-

[10] Sandvik, Runa. "Harvard Student Receives F For Tor Failure While Sending 'Anonymous' Bomb Threat." Forbes. December 18, 2013. Accessed September 29, 2014. http://www.forbes.com/sites/runasandvik/2013/12/18/harvard-student-receives-f-for-tor-failure-while-sending-anonymous-bomb-threat/

[11] "Tor does not claim to completely solve end-to-end timing or intersection attacks." Dingledine, Roger, Nick Mathewson, and Paul Syverson. "Tor: The Second-Generation Onion Router." Proceedings of the 13th USENIX Security Symposium, 2004, 4. https://www.usenix.org/legacy/events/sec04/tech/full_papers/dingledine/dingledine.pdf

[12] https://metrics.torproject.org/network.html

[13] Pricing data from http://aws.amazon.com/ec2/pricing/.

[14] https://github.com/lulzlabs/AirChat/

[15] https://github.com/lulzlabs/AirChat/commit/1b289a8a46c7c25c419a8d2b9b761db13142e53d#diff-c47c7c7383225ab55ff591cb59c41e6bR174

cols,[16] stations are communicating for a long time. The use of ham radios means that stations are also communicating using relatively high power,[17] which, when combined with the need to transmit for long periods of time in once place, means that stations can easily be located using simple techniques.[18]

If stations can be located easily, they can be destroyed very easily. It has been reported that Syria may have used missiles that track satellite phones to kill journalists.[19] One representative satellite phone, the Iridium Extreme 9575 Satellite Phone, transmits at approximately 250mW,[20] or 1/20th the transmit power of a typical cheap ham radio. This means a node running AirChat could face a missile even more easily than a satellite phone. Unfortunately, we believe that high-powered wireless links cannot be the basis of a system designed to protect anonymity in hostile areas, given this capability.

3.3 Commotion Wireless / Project Byzantium

Commotion Wireless[21] and Project Byzantium[22] represent two different ideas in mesh WiFi. Commotion is designed to be installed to consumer WiFi routers, providing mesh connectivity that can give wireless Internet access to neighborhoods; Byzantium describes itself as "Ad-hoc wireless mesh networking for the zombie apocalypse," and provides software to turn consumer laptops into mesh nodes that provide communication services without Internet access. While both projects are of course admirable, they necessarily fall victim to the same problem discussed in Section 3.2: long-lived, fixed wireless transmitters are dangerous for environments of censorship. (The hypothetical zombie apocalypse might be an acceptable time for use of these systems, however, since zombies are unlikely to employ automatic direction finding technology, or missiles, according to standard reference sources.[23]

3.4 Delay-Tolerant Networking

The concept of delay- or disruption-tolerant networking has been the subject of a great deal of research, both by NASA, in the context of Vint Cerf's Interplanetary Networking (IPN) proposals,[24] and by DARPA-funded researchers looking for land-

[16] The default protocol for AirChat is PSK500R, providing 506 baud of throughput; since PSK500R uses binary keying, this means it is transmitting just 506 bits per second.

[17] An average ham radio "Handie-Talkie," or HT, transmits at 5W out of the box; amateur radio operators in many countries may use up to 1500W of transmitting power.

[18] `http://en.wikipedia.org/wiki/Direction_finding`

[19] Smyth, Frank. "Caveat utilitor: Satellite phones can always be tracked," Committee to Protect Journalists. February 24, 2012. Accessed September 29, 2014. `http://cpj.org/blog/2012/02/caveat-utilitor-satellite-phones-can-always-be-tra.php`

[20] `https://apps.fcc.gov/eas/GetApplicationAttachment.html?id=1508135`

[21] `https://commotionwireless.net`

[22] `http://project-byzantium.org`

[23] *See, e.g.,* Zombieland (2009), Day of the Dead (1985), or Plan 9 from Outer Space (1959).

[24] `http://ipnsig.org`

based solutions.[25] The concept is the same, with minor differences in application; both groups want to enable networked communication between nodes that may not be able to support bidirectional end-to-end communication, either due to vagaries of remote terrain (for DARPA), or light-speed communications delays across the solar system (for NASA). This is achieved through store-and-forward routing, with opportunistic data transmission when nodes indicate that they will be able to talk to a destination node (or a node in that direction) sometime in the future.

The DTN concept is perfect in many ways for decentralized communication between marginalized groups; the current implementations, however, leave much to be desired. There is no common implementation of DTN; rather, there are several different codebases, such as DTN2[26], ION[27], and IBR-DTN[28], none of which are considered by their authors to be either complete or stable. The DTNBone project[29] catalogues the efforts being made to create a continuously operating DTN network in academia. NASA, too, has failed to create a DTN network; while several DTN-capable payloads have been launched, other projects necessary to establish a true interplanetary link were scrapped due to "sequestration" and other budgetary cuts.

The DTN protocol as currently designed suffers from other problems. Message security is an optional add-on layer, rather than being a core part of the protocol. The binary protocol is highly complex and requires a parser to conditionally process different parts of the bundle depending on earlier flags,[30] meaning that parsing protocols must be at least context-sensitive.[31] While the DTN concept is highly desirable, then, the implementation is not usable in the field at this time.

4 Method

As noted in Section 3.4, we believe that the DTN concept of store-and-forward routing with opportunistic sending is great for disconnected parties wishing to communicate without Internet access. However, the specific protocol is difficult to work with and has security as an afterthought—and even if it were perfect for our needs, no stable implementations exist. We have therefore implemented our own version of the DTN protocol. The following sections will describe how each major Endrun component works.

[25] https://www.fbo.gov/index?s=opportunity&mode=form&id=9e7abdcb0d96df48de122b0ec28e96d1&tab=core&_cview=0

[26] http://sourceforge.net/project/showfiles.php?group_id=101657

[27] https://ion.ocp.ohiou.edu/

[28] http://www.ibr.cs.tu-bs.de/trac/ibr-dtn

[29] https://sites.google.com/site/dtnresgroup/home/dtn-bone

[30] See IETF RFC 5050, http://tools.ietf.org/html/rfc5050

[31] This is undesirable for security functions; for more information, see the work of the Language-theoretic Security Working Group (Patterson, Hirsch, Bratus, Goodspeed, Sassaman, Shubina, et al) at http://langsec.org.

4.1 Routing

Delay-Tolerant Networking, like TCP/IP, assigns an address to each node in a network, and passes traffic between nodes via whatever route is available. Unlike TCP/IP, however, DTN nodes do not engage in any sort of handshake to exchange data in an end-to-end fashion; instead, each sequential pair of nodes in a communication pathway exchange data, and the recipient signs a chain of custody showing that it has successfully received and has taken responsibility for the data.

Data may take any available path between a pair of nodes in a chain. It might use physical data transport (sneakernet), a radio link, or anything else that is available (including TCP/IP, where that is possible). The crucial difference is that there is no expectation that the source and destination node will be communicating simultaneously at any point; instead, only pairs in the chain will communicate, as shown in Figure 1.

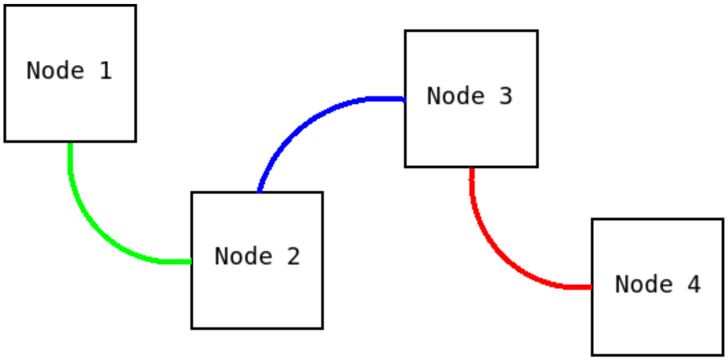

Figure 1: An example of how nodes can transmit data. In this example, there is no direct path between Node 1 and Node 4; Node 2 and Node 3 are used to relay the information.

Because links between nodes are transient, our implementation of DTN does not use anything resembling a traditional routing table. Instead, it uses the "binary spray and wait" transmission strategy.[32] Binary spray and wait allows for the transmission of data with minimal delay and highest efficiency with regards to network traffic, while still allowing for transience in connectivity.

Binary spray-and-wait sets a ceiling on the number of payload copies L that can exist within the network. When a payload is being sent, the originating node starts in "spray" mode, where it transmits L copies of the payload to all available nodes; each subsequent node then transmits $floor(X/2)$ of their total payload copies to every other node without a copy of the payload, where X is the number of payload copies available to the node. When a node has 1 copy of the message remaining, it switches to "wait" mode, where it holds a copy of the payload until either it makes direct contact with the payload's intended recipient or the payload's TTL is reached. Figure 2 shows an example of how the binary spray-and-wait algorithm works.

[32] Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi Raghavendra. "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks." Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking, 2005.

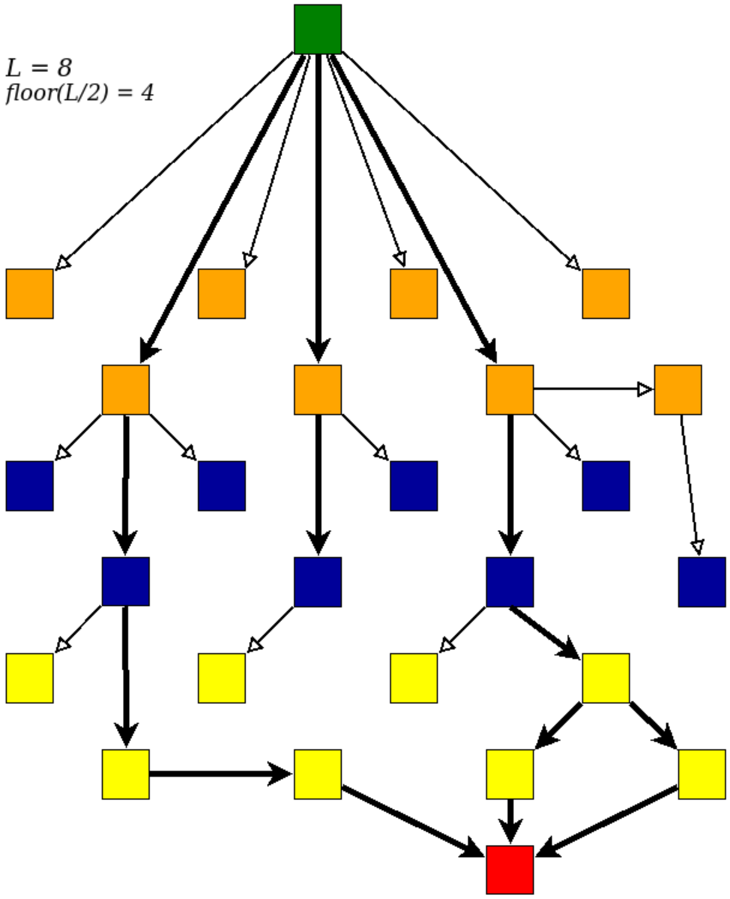

Figure 2: An example of how nodes distribute copies of a payload using the binary spray-and-wait strategy. The green node begins by distributing 8 copies of the payload to every other node it has contact with; the second batch of nodes send 4 copies, the third batch 2 copies, and the final batch holds 1 copy until direct contact with the payload's intended recipient can be made.

4.2 User Services

The services provided to users within Endrun are fairly simple. In addition to transport-related services, discussed in Section 4.4, users have access to Gollum,[33], a Git-based wiki. The wiki provides standard text-based services, and also supports file uploads.

On top of Gollum, then, users can implement any of the following that are desired:

- Encrypted user-to-user messaging, using GPG or another encryption system.
- Forum/bulletin board functionality, using the wiki directly.
- Encrypted file transfer, using a simple encryption tool such as Minilock.[34]

[33] `https://github.com/gollum/gollum`
[34] `https://minilock.io`

These services allow the basic use case—allowing communications between groups of people unable to use the general Internet—to take place without adding excessive complexity.

4.3 Message Format

The Endrun payload message format is designed to be very simple to parse. It has the following fields:

1. **Source** - The identifier for the source node; by default, the node's public key fingerprint.
2. **Destination** - The identifier for the destination node; by default, the node's public key fingerprint.
3. **TTL** - The message time to live, an integer number representing the epoch time at which the bundle was created. Nodes destroy messages after a configurable amount of time has passed since their creation (by default, 24 hours).
4. **Contents** - The signed, then encrypted, then signed message bundle.
5. **Chain of Custody** - A JSON array detailing the transmission path of the payload. See Section 4.4.1 for more details.

Payloads are encrypted with the libsodium[35] implementation of Daniel J. Bernstein's NaCl (pronounced "salt") library.[36] To prevent a variety of attacks,[37] we use the Sign-Encrypt-Sign methodology, whereby the bundle is signed, the signed bundle is encrypted, and the encrypted payload is then signed again; this lets a recipient verify that the bundle has not been tampered with before decrypting it, then check for message validity afterward.

Our implementation uses Git as the primary medium of information exchange; data is stored in Git at each node, and to transmit data in an Endrun payload, we generate a Git bundle[38], Sign, Encrypt, and Sign it, and then transmit it over the DTN. At the receiving end, after the Verify, Decrypt, and Verify steps, we run a Git pull from the bundle, allowing the data to be absorbed into the local git tree.

To generate a shared Git tree base, each tree must set up a Git remote for each other tree, which is an operation with $O(n^2)$ complexity. As each operation in this sequence is relatively processing-intensive, we ordinarily generate nodes in groups of just eight, allowing node generation to proceed in a reasonable amount of time. We have tested node generation in groups of up to 200, but also note that the sheer number of nodes in such groups would make trees take an exceptionally long amount of time to converge through opportunistic data transmission.

[35] http://doc.libsodium.org

[36] http://nacl.cr.yp.to

[37] http://world.std.com/~dtd/sign_encrypt/sign_encrypt7.html

[38] http://git-scm.com/docs/git-bundle

4.4 Transport

4.4.1 Chain of Custody

There are multiple concerns that arise from implementing disruption- and delay-tolerant networking without maintaining a full routing table. Because links between nodes are transient:

- Significant amounts of time can pass between a node becoming compromised[39] and the users of a network realizing that it is compromised;
- It can be non-trivial to identify whether a node has been compromised or whether it is simply unreachable by most of the nodes in a cell; and
- Bandwidth and/or human effort can be wasted in attempting to contact a node, which can jeopardize operational security.

Instead of using a formal handshake, like SYN/SYN-ACK/ACK in TCP, most DTN protocols use a system called a **chain of custody**. Chain of custody allows us to create an audit trail without maintaining global knowledge, which is crucial in solving the aforementioned problems.

Chain of custody in Endrun consists of three parts: a JSON array on the payload, a "receipt" table on the node's SQLite database, and a SQLite "map" table on the SQLite database. The JSON array consists of objects containing a receiving node's identifier, a fingerprint of that node's public signing key, a signed SHA2 hash of the payload when it was handed off from one node to another, and a timestamp in epoch time for when it was handed off to the next node. Every time a node changes custody, a new object is appended to the array.

The "receipt" table records data similar to the objects in the chain of custody array, but only for the recipient node. The table contains columns for entry ID, payload fingerprint, timestamp of when the payload was sent, and the identifier of the node to which the payload was sent. Entries expire after a period of $1\frac{1}{2}$ times the TTL value in the node's configuration file. The table is also stored in temporary storage and recreated upon each reboot of the node.

The "map" table is used to record a list of nodes "around" the current node; in other words, nodes we have been able to contact recently, regardless of cell affiliation. This is important for intermediate routing, which we will cover in Section 4.4.5. This table has three columns: one for entry ID, one for node identifier, and one for timestamp. Whenever a new node is encountered[40], an entry is added to the table; the timestamp for an entry is updated with every subsequent encounter. Entries expire after a period of $1\frac{1}{2}$ times the TTL value in the node's configuration file; this ensures that we do not attempt to contact once-familiar nodes ad infinitum.

[39] In this case, a compromised node is one that is no longer trustworthy due to an extremely long period of time without contact, physical capture by a hostile actor, etc.

[40] This usually occurs when a payload is received from an Endrun node.

4.4.2 Sneakernet

> *"Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway."*[41]

As we discussed throughout Section 3, there are significant security issues related to both wired and wireless network connections. One additional issue is that in many situations, traditional network connections may not be available *at all*. Over the last 5 years, we've seen examples in Egypt[42], Turkey[43], and the United States[44] where governments have attempted to curtail unrest by cutting off people's access to telecom services. Even in situations where such outages are *not* intentional[45] [46], we still see that people's ability to communicate digitally can be incredibly fragile.

These issues are largely responsible for our decision to make removable USB storage—fondly known as Sneakernet—Endrun's primary transport method. The benefits are numerous: high-capacity USB storage is inexpensive, easily obtainable, reusable, and (in emergencies) disposable. Most USB flash drives are under 3 centimeters in size and weigh less than 30 grams, making them easily portable and highly concealable. USB also has immeasurable architectural advantages; almost every desktop or laptop computer manufactured in the last decade has at minimum 1 USB port, meaning that Endrun can be used with almost any computing hardware that will still boot.

Endrun has both automatic and manual support for Sneakernet. The automatic method is simple to configure: a user places a hidden[47] text file in the root directory of the USB drive with a list of node identifiers. Once configured, whenever the drive is plugged in to a node, the Endrun daemon automatically mounts it, copies any payloads already on the drive, and attempts to import them. It then looks for the hidden text file and generates payloads for every node listed in the text file for which it has a public encryption key. Once it has finished generating payloads, it unmounts the drive.

The manual method, by comparison, requires a user to log into the node, manually mount a USB drive, and run one or two shell commands (depending on whether payloads are being imported, exported, or both.) While requiring a modest amount

[41] Tanenbaum, Andrew S., and D. Wetherall. "Physical Layer." In Computer networks, 96. 5th ed. Boston: Pearson Prentice Hall, 2011.

[42] Williams, Christopher. "How Egypt Shut down the Internet." The Telegraph. November 28, 2011. Accessed September 29, 2014. `http://www.telegraph.co.uk/news/worldnews/africaandindianocean/egypt/8288163/How-Egypt-shut-down-the-internet.html`

[43] Welch, Chris. "Turkey Reportedly Cuts off Twitter Backdoor by Blocking Google DNS." The Verge. March 22, 2014. Accessed September 29, 2014. `http://www.theverge.com/2014/3/22/5536230/turkey-reportedly-cuts-off-twitter-backdoor-blocks-google-dns`

[44] Bell, Melissa. "BART San Francisco Cut Cell Services to Avert Protest." Washington Post. August 12, 2011. Accessed September 29, 2014. `http://www.washingtonpost.com/blogs/worldviews/post/bart-san-francisco-cut-cell-services-to-avert-protest/2011/08/12/gIQAfLCgBJ_blog.html`

[45] Farrell, Michael. "Cellphone Networks Overwhelmed after Blasts in Boston." BostonGlobe.com. April 17, 2013. Accessed September 29, 2014.`http://www.bostonglobe.com/business/2013/04/16/cellphone-networks-overwhelmed-blast-aftermath/wq7AX6AvnEemM35XTH152K/story.html`

[46] "YouTube Hijacking: A RIPE NCC RIS Case Study." RIPE Network Coordination Centre. March 17, 2008. Accessed September 29, 2014.`http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study`

[47] For purposes of preventing accidental deletion, rather than preserving utmost secrecy.

of technical knowledge, this method is convenient in situations where a user might need to specify the exact order of import, such as with multiple payloads from a single source. It also allows a user to prevent the automatic mounting of USB storage hardware.

4.4.3 XBee Radio

We also support using Digi XBee modules[48] for inter-node communication. XBee modules present a raw serial interface to a client, and handle everything else—mesh formation, link encryption, CTS/RTS, etc.—on their own, which makes it extremely easy to integrate. In addition, XBee modules present a unified form factor and interface, regardless of what frequency band (868MHz, 900MHz, 2.4GHz), protocol (ZigBee, 802.15.4, proprietary meshing, multipoint), or power level (10mW, 16mW, 100mW, 200mW, 250mW) is chosen. For our implementation, we used 900MHz automeshing 250mW modules (XBP9B-DMWT-002 and XBP9B-DMST-002);[49] which module is correct for a given application will depend on distance to be travelled, throughput requirements, etc. XBee does provide AES-based link encryption; while we enabled it as a matter of course, we believe that our NaCl-based encryption provides sufficient integrity guarantees with or without link encryption. (That said, we found no reason to disable it.)

From the user's perspective, XBee works much like Sneakernet, but without physical motion; once a bundle is generated, Endrun drops it into a queue for transmission. Since XBee handles waiting for a clear channel and waiting until a message has been fully passed before handing it off to the receiving node, Endrun nodes can simply treat the XBee module in the same way as a USB drive.

4.4.4 Sneakernet+

As an alternative to Sneakernet that still involves physical couriers and very-short-range wireless links, we offer mobile-device-based storage, which we refer to as Sneakernet+. In this strategy, a bundle is generated, and loaded into HTML5 Web Storage on a mobile device. The user then disconnects from the node, travels to another node, and connects to the web service on the remote node; that node accesses the user's Web Storage, retrieves the bundle, and then proceeds as though it had come from a USB drive, as above.

4.4.5 Intermediate Routing

One limitation of the original Endrun design was that the nodes of a cell could only talk to one another. Intermediate routing—the ability to move data between Endrun

[48] `http://www.digi.com/xbee/`

[49] Note that 900MHz is in a license-free ISM (Industrial, Scientific, and Medical) band available in the United States; this frequency band is not available in Europe, but Digi provides modules on appropriate bands for all markets.

nodes using any other Endrun node, regardless of its cell affiliation—has significant consequences for the overall utility of the project, and thus became the focus of much of our development.

When a payload arrives at an Endrun node, the Endrun daemon first attempts to verify and import the payload. If the payload does not match any existing public key that the node has available to it, the daemon then compares the payload's destination node identifier to the ones stored in the 'map' table of its SQLite database.[50] If the node has access to the destination node, it queues the payload for direct transmission at the next available opportunity. Otherwise, an entry is added to the "map" table for the payload's source node, and the payload is queued for transmission to all available nodes. This continues until the payload either reaches its destination or its TTL expires.

In theory, given sufficient internal storage on nodes and a long enough TTL, intermediate routing should permit Endrun cells to function in similar fashion to existing packet-switched networks. However, due to Endrun's relative nascency, we have not yet had the opportunity to sufficiently test this.

5 Communication Example[51]

Figure 3 illustrates how Endrun can be used in practice. Tom initiates a data transfer by generating an Endrun payload, and it gathers additional information at each node. Tom transferred the data to Marie, who transferred it to Harry, who provided it to Louise.

Meanwhile, in France, Francois and Jacques compile some interesting information and then generate an Endrun payload for Pierre. Pierre grabs it via Sneakernet+, and then generates a payload for Sheila, who is visiting him in Paris.

Louise wants to transfer data to Agnes. Louise transfers the data via Jim. Louise has additional data from Joan, who couriered a USB stick from Sheila (who is now in Ireland), and passed the data in turn to Daphne and Giles.

Edith's father obtains some sensitive information and gives it to Edith for collation and cleanup. Edith then issues two Endrun payloads, saves them to some spare SD cards she has lying around, and gives one copy each to Daniel and Max. Once Daniel returns home, he imports the payload into his personal Endrun node, generates a new payload for his dentist (who happens to be an intelligence operative), and saves it to a MicroSD card. Daniel then clandestinely places the card in the collar of his prize-winning spaniel, whom he takes to his dentist's appointment the next day.

Agnes, who happens to be primary data coordinator in her circle of friends, then transmits the data to Tom, who also receives data from Sue, Millie, Billie, Gillie, and Willie. All of this data happens to be duplicate transmissions of the same payload.

[50] See Section 4.4.1 for more information.

[51] With apologies to Tom Lehrer, Emeritus Lecturer in Mathematics, Cowell College, University of California at Santa Cruz.

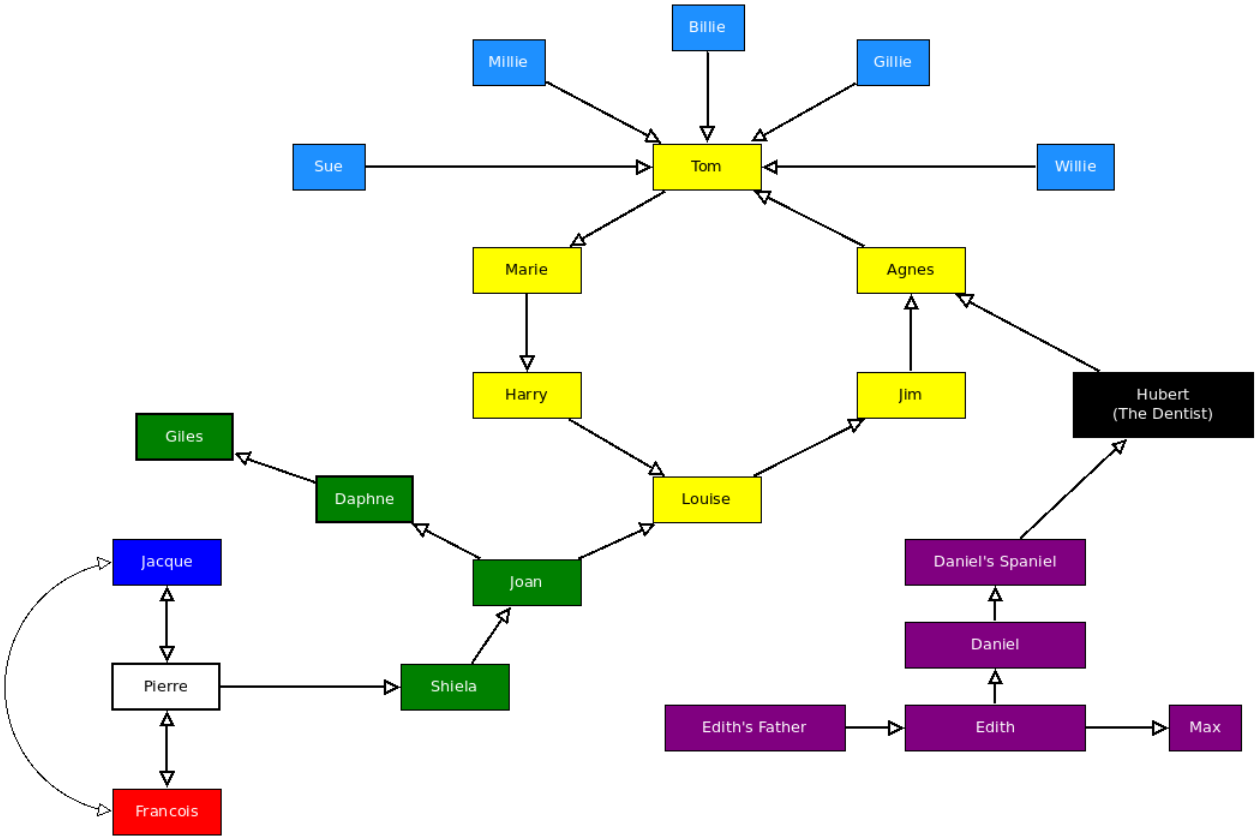

Figure 3: An example of how Endrun can be used by a small number of people.

6 Future Work

There are several features we would like to add to Endrun to fill out its capabilities. In addition, as a distributed, sequenced data storage and communications system, Endrun could be used to implement a wide variety of systems. We will detail two that we think are particularly compelling.

6.1 Additions to Endrun

- **Auditing** - At the moment, the chain of custody can be used manually to trace the path a message took, and to debug problems in a transmission path. However, the chain is stored locally on each node and on each bundle, and so when bundles fail, it is currently labor-intensive to contact each node to find which processed a particular bundle. To rectify this, nodes could be configured periodically to send their lists of known bundles to a centralizer, which could perform traffic analysis after the fact. We have not yet implemented this, as there are significant security concerns about centralizing that level of audit logs, but there are use cases in which it would be helpful.

- **Bluetooth Beaconing** - Both iOS and Android provide for Bluetooth beacons (for Apple, they are called iBeacons) to identify locations and data sources; these beacons do not require pairing to communicate their (very short) messages, and correctly-programmed applications can use the beacons to trigger additional actions. We could add beacon functionality to native apps implementing the Sneakernet+ functionality we currently use, so that iOS and Android phones could receive a beacon, automatically connect to an Endrun node, download/upload bundles, and disconnect with little or no user interaction. This would allow Endrun nodes to be serviced by mobile platforms that may not have the time or capability for constant human interactions; for instance, mobile nodes could be attached to cars, bicycle couriers, or city buses.

6.2 Best-Effort Surveillance

Current large-scale surveillance infrastructure is based around centralizing as much data as possible. Some proposals for increasing the scale of surveillance in major metropolitan areas include not only unprecedented government-installed camera density, but also integration of private cameras (e.g., business surveillance cameras). Attempting to get real-time video from an increasing amount of cameras into a single location requires an extreme amount of data throughput—but this is unnecessary. Most alleged goals of widespread surveillance could be met with eventual data delivery, rather than immediate, because there are few situations in which data is actually analyzed in real time. (This is useful, as it is not cognitively possible for a human to analyze tens of thousands of simultaneous real-time video feeds.) Therefore, we could extend Endrun to be used with cameras that will store data on nodes. Periodic collections—for instance, by mounting mobile Endrun nodes on city buses—could "pick up" the data and deliver it to centralized surveillance centers after some time, thus capturing the same data with more reasonable throughput needs.

6.3 Long-Distance Debt Instruments

In *Neptune's Brood,* Charles Stross details a three-tiered system for interstellar currency, with "fast money," "medium money," and "slow money" describing progressively longer-term debt instruments. We believe that this concept, in conjunction with Endrun, could be used to create a system of debts and favors that could be traded between hackerspaces around the world, providing not just a way to identify key members of foreign hackerspaces (the so-called "hackerspace passport" concept), but a way for hackerspaces to help traveling members and have some ability to be repaid in turn. We hope to explore this idea in a future project.