Torrent-Based Dissemination in Infrastructure-Less Wireless Networks*

Kyriakos Manousakis[1], Sharanya Eswaran[1], David Shur[1], Gaurav Naik[2], Pavan Kantharaju[2], William Regli[2] and Brian Adamson[3]

[1]*Applied Communication Sciences*
[2]*Drexel University*
[3]*Naval Research Laboratory*
Corresponding Authors: {*kmanousakis; seswaran; dshur*}*@appcomsci.com;*
{*gn; pk398; regli*}*@drexel.edu; brian.adamson@nrl.navy.mil*

Received 10 February 2015; Accepted 11 March 2015;
Publication 22 May 2015

Abstract

Content dissemination in peer-to-peer mobile ad-hoc networks is subject to disruptions due to erratic link performance and intermittent connectivity. Distributed protocols such as BitTorrent are now ubiquitously used for content dissemination in wired Internet-scale networks, but are not infrastructure-less, which makes them unsuitable for MANETs. Our approach (called SISTO) is a fully distributed and torrent-based solution, with four key features: (i) freedom from any reliance on infrastructure; (ii) network and topology aware selection of information sources; (iii) robust multiple-path routing of content via a proactive peer selection technique; (iv) an integrated distributed content discovery capability, not found in other torrent systems. We have implemented SISTO in software, and evaluated its performance using emulation and realistic mobile network models derived from field measurements. We have measured significant improvements in download latency, resiliency and packet delivery compared to traditional data delivery models and conventional BitTorrent. We have implemented SISTO on both Linux and Android platforms, and integrated it with several android applications for content sharing.

*This work has been funded by the Office of Naval Research (Contract No. N00014-12-C-0377)

Journal of Cyber Security, Vol. 4, 1–22.
doi: 10.13052/jcsm2245-1439.411

Keywords: Mobile ad-hoc networks, peer-to-peer, data dissemination, algorithms, performance, design, reliability, experimentation.

1 Introduction

There exist a wide set of network scenarios, such as first responder and disaster recovery situations, and military and tactical operations, which require applications and protocols to function in a purely ad hoc peer-to-peer fashion, without support infrastructure. (Here support infrastructure is taken to mean the existence of nodes and functionality maintained outside the scope of the set of ad-hoc nodes, yet available to them. The DNS infrastructure is an example of support infrastructure). In many MANET situations, this type of infrastructure support is not feasible. Furthermore, it is critical in such ad-hoc networks, that node or link failures are well tolerated, so that a receiver can obtain data even if the original source is temporarily or permanently disconnected.

While it is possible to address this problem through MANET routing techniques [19], design, development and deployment of routing protocols have long lead times. Consensus in the technical community, on which are the best routing protocols has not emerged despite years of research. Content Distribution Networks (CDNs), such as BitTorrent address the problem above the network layer, thereby avoiding the problem of deploying new routing protocols. In BitTorrent, original content is broken into pieces, and pieces may be individually disseminated. A receiver may obtain pieces concurrently from multiple sources, dispersed across the network. Our approach builds upon the BitTorrent [1] protocol, and addresses the limitations of BitTorrent in a MANET environment as follows:

- Unlike conventional BitTorrent systems, it does not require support from BitTorrent support infrastructure (the servers for content/peer discovery and tracking).
- It includes procedures for identifying and selecting sources based on favorable topological factors and/or network conditions such as congestion in order to form a robust multiple path distribution network for each piece of content. Our multiple path technique is highly tolerant to connectivity disruptions, and offers significantly more robust dissemination than traditional methods, which tend to map all traffic onto a single path between a source and destination.
- It provides for content discovery (not present in BitTorrent and most other CDN technologies, which assume an out-of-band mechanism) using a fully-distributed content discovery mechanism, where in the

users can query for content of interest using key words, and discover the corresponding torrent metadata; by means of the metadata, content is automatically acquired and delivered based on the SISTO torrent algorithm.

The remainder of this paper is organized as follows: Section 2 focuses on related work; Section 3 describes the SISTO architecture and protocols; Section 4 provides a quantitative experimental evaluation of SISTO, and Section 5 provides conclusions.

2 Related Work

Significant related work on peer-to-peer networking exists. Some examples are based on BitTorrent (e.g., SPAWN [5], and CodeTorrent [6], which also uses network coding to help deal with mobile network issues), while others such as 7DS [7], XL-Gnutella [8], are not. In the above related work, the infrastructure question is not addressed, and mechanisms for the discovery of content are not provided – it is assumed that content metadata (e.g., *.torrent* files) are provided through an out-of-band mechanism such as a well-known web server. ORION [9] proposes content query and searching features, but the content technique is does not appear to be separable from the highly specific dissemination technique in that work. In our work, we incorporate a fully distributed mechanism for content discovery based on ProtoSD [13], which is loosely coupled with the rest of the system. In [10], a cross-layer approach for using network information is described, but it does not use peer-to-peer mechanisms, and therefore lacks the required robustness and dissemination efficiency. In [11] a novel torrent-based system is described based on Bluetooth communications, but the mechanisms are coupled with the blue-tooth protocol and do not generalize to other communication techniques. A topology-aware BitTorrent client is developed in [12] for Internet-scale networks in which peers are selected based on hop count and transmission rates. However, it does not take the link quality into account, and uses passive monitoring of connections between peers to estimate the rates. Other work such as [13] and [14] also select peers based on estimating available bandwidth using the technique of packet-pair dispersion. However, as shown in [15], packet dispersion methods do not provide accurate bandwidth estimation in wireless networks. The original SISTO concept was proposed in [16], and in this follow-up paper, we report on the design enhancements of the original concepts and also on results from testing of our fully functioning implementation.

3 SISTO Architecture

When applied to MANETs, the main drawbacks of conventional torrent BitTorrent are (a) that it was developed for large scale, stable networks, and (b) assume infrastructure support, either via peer tracker servers or a core set of servers forming a decentralized peer discovery overlay network. Such assumptions are not suitable for small-scale wireless mobile ad hoc networks (MANETs), especially those that display dynamic behaviour, where connections to any specific servers or infrastructure may not be available, or if available may be subject to frequent disruption.

3.1 Distributed Infrastructure-Free Peer Discovery

In SISTO, like BitTorrent, the peers are discovered using the Distributed Hash Table (DHT) technique in a distributed manner. However the conventional BitTorrent DHT bootstrap process, while distributed, depends on the infrastructure support of the global DHT overlay network. This global DHT network assumed by BitTorrent is “infrastructure” for torrent systems in the same way as the DNS server overlay is infrastructure for IP networks. Furthermore, because of the scale of the global DHT overlay network support infrastructure, in conventional BitTorrent, a new peer needs to connect to at least 50 other peers in order to participate in the DHT network. In a typical MANET, there may not even be 50 nodes in the entire network. In SISTO we have modified the bootstrapping by: a) removing the dependencies on the global DHT network, and b) we have set the number of DHT peers to be a configurable parameter depending on the characteristics of the corresponding network (e.g., size). In SISTO, a peer bootstraps from a local ad-hoc DHT overlay network, using a set of known locally stored addresses that have been configured manually or learned from previous peer connections. This enhancement allows SISTO use the members of the torrent swarm to form a peer discovery DHT network among themselves, allowing it to operate without any connectivity to the torrent DHT support infrastructure in the Internet. This seemingly minor alternation in the architecture completely changes the applicability of this approach, making it possible to be used in MANETs. **Figure 1** depicts the enhanced, fully distributed infrastructure-less DHT bootstrap process in SISTO.

Since the DHT discovery network in SISTO will now be typically a much smaller (MANET) sized entity, we take advantage of this fact to speed-up peer discovery, by introducing a new peer discovery control parameter, which can be set to allow an earlier start of downloads.

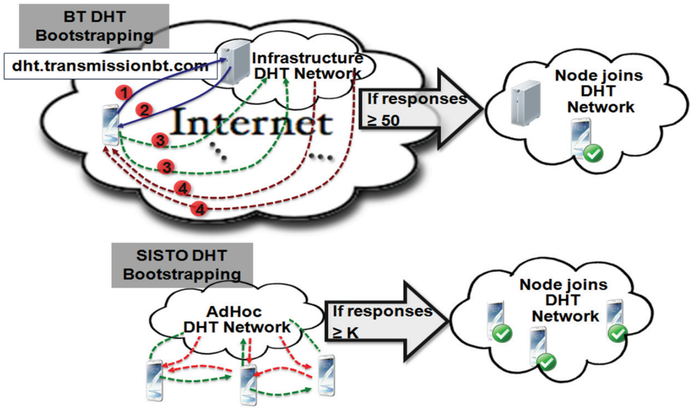


Figure 1 DHT bootstrapping in SISTO vs. BitTorrent.

3.2 Network-Aware Peer Selection

Peer selection is the process by which each peer decides which subset of its peers to upload data to. In SISTO, we have made another change relative to conventional BitTorrent where peers are selected in a random fashion. In SISTO, peers are selected using cross-layer information in a network- and topology-aware manner. Furthermore, the number of peers (called the Upload Number in this paper, which is normally fixed in BitTorrent) is adjusted dynamically based on network conditions. The network-aware peer selection is designed with the following objectives: (i) reducing long distance transmissions and localizing the transmissions, so that the channel contention and interference is reduced, (ii) utilizing stable, high performing links so that the efficiency of data dissemination is higher, and (iii) since it is common for such networks to have links with frequently varying bandwidth resulting from factors such as mobility and terrain effects, avoiding the problem of underutilization of low bandwidth or mildly lossy links (which are treated unfairly with the conventional BitTorrent scheme).

The cross-layer information needed by SISTO can be acquired in multiple ways. Firstly, SISTO includes a simple Network Monitoring tool, enabling each node to periodically gather latency, hop count and loss information with respective to other known peers in the swarm. SISTO is designed to read this information from a pre-specified *location:port* and in a self-descriptive, standardized JSON format. This architectural decision allows any third-party process that provides network information, such as a routing protocol agent, Dynamic Link Exchange Protocol (DLEP) agent or a dedicated network awareness service [23], to be easily plugged into the SISTO interface in place of the SISTO-provided tool.

SISTO implements the following three algorithms for network-aware peer selection (PS):

Hop-only PS: The peers are ordered based on the hop count from the uploading node, preferring the closer peers. When there is a tie, the download rates (which are used in conventional BitTorrent – referred to as TitForTat or TFT) are used as secondary criteria.

Latency-Hop PS: The peers are ranked in the increasing order of latency (round trip time) between the peer and the uploading node. When there is a tie, the hop count and download rates are used as secondary and tertiary criteria.

Loss-Hop PS: The peers are ranked in the increasing order of packet loss between the peer and the uploading node. When there is a tie, the hop count and download rates are used as secondary and tertiary criteria.

Once the peers are ranked using one of these policies, the data is uploaded to the top N peers, where N is the upload number, and the other peers are choked. This peer selection process (called the re-choke cycle) is repeated every 10 seconds. It is assumed that the Network Monitoring tool described above is providing hop count, latency and loss measurements.

Another important factor that impacts performance is the value of the upload number itself, since if it is too low, the throughput achieved is low, and if it is too high, network congestion may result. Unlike BitTorrent, SISTO adjusts the upload number *dynamically*, based on the network conditions. The re-evaluation of upload number occurs every alternate (configurable) re-choke cycle, i.e., every 20 seconds (by default) in our implementation. The parameters that are used to determine the upload number value are (i) total upload rate (TUR) across all peers, i.e., the total number of bytes per second that the node uploads to all its peers averaged over the sampling time window and (ii) average latency (AL) between the node and its peers. When it is time to evaluate the upload number at a node, the current TUR and AL are compared with the TUR and AL from the previous cycle. If there is an increase in the AL or a drop in TUR, it indicates the build up of congestion in the network. If there is a decrease in AL or increase in TUR, it indicates the availability of network capacity, especially when new peers join the network. Based on these observations, a heuristic adjustment policy is employed, as shown in **Table 1** the upload number is linearly incremented when AL decreases or TUR increases, linearly decremented when AL increases or TUR decreases, and unchanged in other cases.

Table 1 Dynamic adjustment policy

AL	TUR	Upload Number
↑	↓	–1
↓	↑	+1
↑	↑	+0
↓	↓	+0

3.3 Adaptive Re-Routing via Proactive Peer Creation (PPC)

In conventional BitTorrent, peers that serve as seeds are selectively reactively, based solely in whether or not an application attached to that peer wants to receive the information in question. Proactively selecting new seeds enables a highly robust dynamic route selection at the application layer. For example, suppose there is a single source-receiver pair (*n5* and *n8*) in the network, as shown in **Figure 2**. Suppose that the shortest path between the two nodes, as chosen by conventional routing, is of poor quality, possibly due to congestion. Suppose that there exists an alternative, but longer path between *n5* and *n8*, which is not congested. If we trigger one of the nodes in the alternative path, say *n12*, to become an additional seed for the torrent, then n5 sends data to *n12*, and *n8* subsequently receives it from *n12*. The torrent algorithm will then naturally begin to favor the high-performing path. In other words, the torrent dissemination process steers the traffic through the alternate, better route. SISTO exploits this potential and defines mechanisms to identify such peers, which can improve performance significantly. Furthermore, enabling more than one such peer can cause multiple new paths between the source

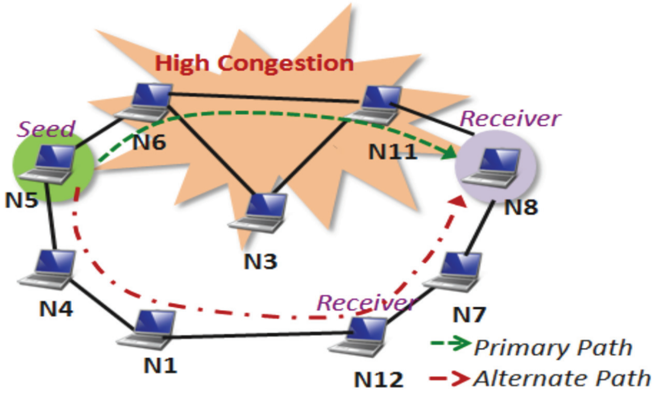


Figure 2 Example scenario for proactive peer selection.

and the destination to emerge. Thus traffic can be disseminated to receivers on multiple paths, which offers not only performance improvements, but makes the entire dissemination process more robust, since multiple paths may be active in parallel, and information dissemination adaptively favors the better performing paths. Note that that can also occur in conventional BitTorrent, but without PPC, if it occurs it is dependent on other peers in the right place being interested in receiving the content, while PPC ensures that it happens by design. We believe that there is a high potential in proactive peer creation, especially in congested networks, when transient links become available, and in the vicinity of "weak spots" in the network, (which are nodes that if they fail, will cause partitioning).

In the current version of SISTO, we design and implement PPC to address the aforementioned problem of congestion. Accordingly, when the observed latency between a peer and its actively downloading peer (i.e., peer that it uploads data to) exceeds a threshold value, PPC is enabled for this source and receiver pair (*S, R*) and torrent *T*. The source peer (i.e., the uploading peer where PPC is enabled) obtains a list of known peers in the network by reading the DHT overlay node list (we assume that all peers want to participate in PPC). These peers may or may not already be a member of torrent *T*'s swarm (since the DHT network is established independent of the torrents being exchanged). Let this list of peers be $L_1 = \{p_1, p_2, p_3, ...\}$. *S* then obtains the latencies between *S* and each node in L_1 from the local Network Monitor. Let this list of latencies be $L_2 = \{lat(S, p_1), lat(S, p_2), lat(S, p_3), ...\}$, where *lat(a, b)* is the latency between peers a and b. The nodes in L_1 are sorted in the increasing order of latencies from L_2. A fraction of nodes in the sorted list (50% in our implementation), are queried for their observed latencies to *R*. Let this list be $L_3 = \{lat(p_1, R), lat(p_2, R), lat(p_3, R), ...\}$. The number of nodes queried can be changed according to the desired trade off in communication overhead. Based on L_2 and L_3, the node p_i which yields the least total latency $lat(S, p_i) + lat(p_i, R)$ is selected to be enabled as a peer for *T*. A request is sent to p_i to add the torrent; if the new node rejects the request, then the node with the next lowest total latency is selected. This algorithm is highly adaptive to changing network conditions.

3.4 Content Discovery

In most conventional CDNS (including BitTorrent) content discovery is assumed to take place out of band (e.g., via email, or social media). In SISTO, an application can request content either using a set of keywords, by selecting content from keyword search results, or by directly referencing specific content metadata. The Content Discovery component is responsible for creating metadata and distributing these across the network, as well as publishing advertisements from peers that want to seed and share content. Nodes that are interested in the published content use the relevant metadata obtained in the content discovery process, and then discover peers by means of DHT techniques. Once a torrent swarm of peers is established for the requested content, data dissemination begins.

For content metadata, SISTO uses magnet links instead of the traditional "*.torrent*" metafiles, because of their small size. The small size of the metadata associated with the torrent content, allows it to be disseminated using very simple techniques, i.e., the magnet links are pushed onto the network and discovered by peers in a distributed manner, using ProtoSD [13], which is a discovery system, which helps publish, query and disseminate content references across the network. ProtoSD uses two service discovery protocols: (1) multicast-DNS (MDNS) [4] and (2) Independent Discovery Interface (INDI) [12]. Using INDI, ProtoSD is able to discover and disseminate services effectively on dynamic, low-connectivity networks without any infrastructure, similar to the modified DHT technique described above.

SISTO allows content-providing peers (seeds) to tag content with keywords and other metadata, and push content advertisements to the network periodically, which are picked up by other nodes and stored in their local knowledge base, as shown in **Figure 3**. A sample advertisement of media content is shown in **Figure 4**. Along with advertising, the nodes also seed their content, i.e., makes the content available to other peers as a torrent for

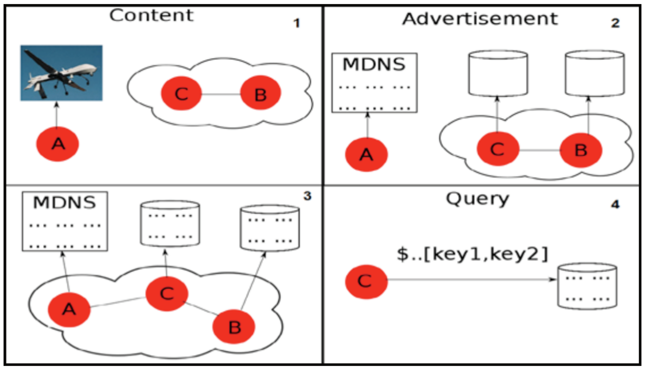
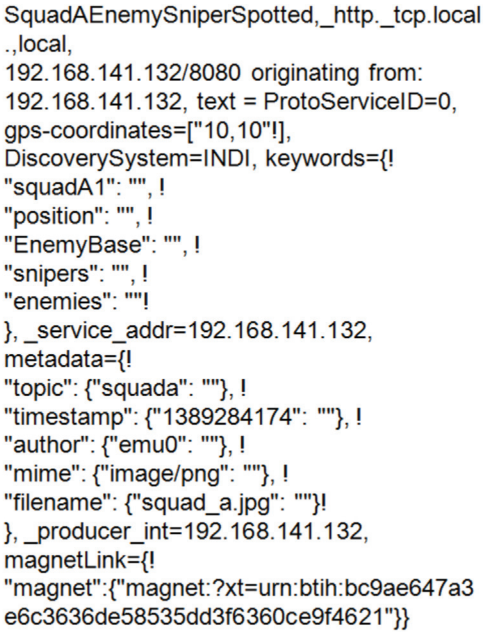


Figure 3 Content publishing and discovery: Content created at node A (1) is advertised to other nodes (2, 3). If a query fails in the local knowledge base, it is retrieved from a different node (4).



```
SquadAEnemySniperSpotted,_http._tcp.local
.,local,
192.168.141.132/8080 originating from:
192.168.141.132, text = ProtoServiceID=0,
gps-coordinates=["10,10"!],
DiscoverySystem=INDI, keywords={!
"squadA1": "", !
"position": "", !
"EnemyBase": "", !
"snipers": "", !
"enemies": ""!
}, _service_addr=192.168.141.132,
metadata={!
"topic": {"squada": ""}, !
"timestamp": {"1389284174": ""}, !
"author": {"emu0": ""}, !
"mime": {"image/png": ""}, !
"filename": {"squad_a.jpg": ""}!
}, _producer_int=192.168.141.132,
magnetLink={!
"magnet":{"magnet:?xt=urn:btih:bc9ae647a3
e6c3636de58535dd3f6360ce9f4621"}}
```

Figure 4 Sample content advertisement.

download. A client that wants to download content can query for keywords pertaining to the relevant magnet link. If the node that pushed the link advertisement fails, the magnet link may be retrieved from the knowledge base of other nodes. The node subsequently uses this magnet link to download the data via the torrent algorithm.

4 Evaluation

The SISTO system has been implemented in C++ building from the libtorrent library [20]. The software implementation has been evaluated using a realistic 30 node mobile network obtained from field measurements emulated on a Common Open Research Emulator (CORE)/Extendable Mobile Ad-hoc Network Emulator (EMANE) testbed [21, 22]. Both static and mobile versions of this network are used in the experimentation. For the experiments in Section 4.1 below, we used the basic range model (130 m) with the bandwidth on all the links set to 200 Kbps, an average packet loss of 5% and delay of 20ms. For the experiments in Sections 4.2, 4.3 and 4.4, the CORE testbed, which emulates the network layer and upper layers, was integrated with

Extendable Mobile Ad hoc Network Emulator (EMANE) for emulating lower layers (e.g., the 802.11abg MAC model was applied), and the link rates were set to 2 Mbps.

4.1 Peer Discovery

We compare the performance of SISTO's enhanced DHT with BitTorrent's DHT by downloading 5 video files across the network (in both static and mobile scenarios); each file size was 4.5 MB, and was seeded by 1 peer and requested by 10 peers. The experiments were repeated with different sets of seeds and receivers that were selected randomly. **Figure 5** shows the average peer discovery latency in SISTO and BitTorrent. We observed that SISTO reduces the peer discovery latency by 7.7% on average. To see the impact of these differences on the actual data download, we measured the download latency, i.e., the average time taken to for a receiver to download the entire file. **Figure 6** shows the download latency for BitTorrent and

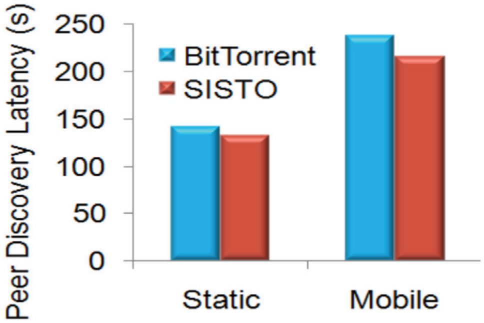


Figure 5 Average peer discovery latency of SISTO vs. BitTorrent.

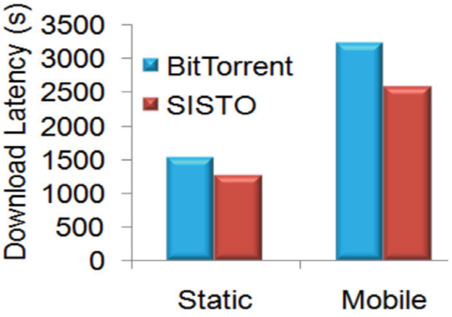


Figure 6 Download latency of BitTorrent vs. SISTO.

SISTO (without network/topology awareness). We see that SISTO reduces the download latency by 19% on average. The overhead of peer discovery in SISTO, i.e., the overhead of SISTO's enhanced DHT messaging, as a fraction of the total network traffic is only a small fraction of the total network traffic (<1.5%) and scales well with network churn.

4.2 Content Dissemination Efficiency

Similar sets of experiments described in Section 4.1 were used to evaluate the content dissemination efficiency of SISTO. **Figure 7** shows the download latency for experiments in both static and mobile network scenarios. The performance of SISTO is compared with conventional HTTP, which relies on the underlying routing protocol (OLSR in this case) for failure recovery.

It may be noted that the network-aware optimizations were not enabled in SISTO for these experiments. We see in **Figure 7** that even in a static network only 55% of the file downloads actually completed – this is due to the conventional approach's limited ability to recover from mobility driven degradation. Furthermore, in the mobile network, even less, namely 32% of downloads completes. In contrast, SISTO delivered 100% of the requested content both static and mobile cases, indicating that it is much more robust to both congestion and mobility effects.

Figure 8 shows the measured the traffic overhead incurred by SISTO control messages for data dissemination, such as peer handshake, piece request, etc. We observe that this is a small fraction of the total traffic (<1%), and also that the overhead is lower in 5-torrent than 1-torrent, indicating scalability of SISTO.

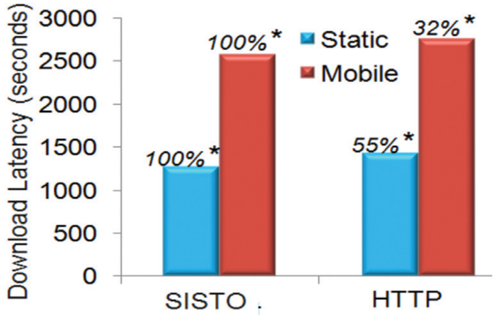


Figure 7 Download completion (HTTP vs. SISTO).

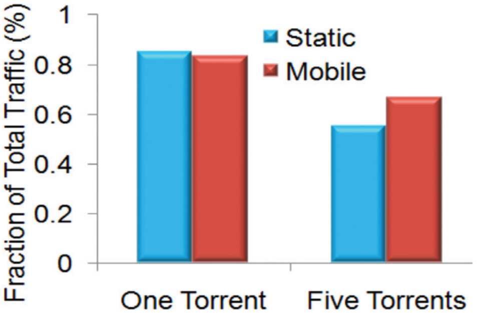


Figure 8 Content dissemination overhead of SISTO.

4.3 Network-Aware Peer Selection

In this series of experiments, we employed the same five-torrent scenario described above and evaluated the SISTO network-aware peer selection algorithms against each other, and against tit-for-tat (TFT) peer selection of BitTorrent. A static network without any node mobility was used, and the upload number of peers was fixed to 3.

Figure 9 shows the average download latencies. We see that the network-aware peer selection yields better performance than TFT. For this scenario, the use of the Hop-only scheme reduced the download latency by 19% and Latency-Hop by 28% compared to TFT. Note that although Latency-Hop performed the best in this experiment, there may be other scenarios or situations where the other network-aware peer selection algorithms are useful. E.g, when some links are lower bandwidth than others, Hop-only prevents starvation of these links; or when only loss or hop measurements are available, Loss-Hop or Hop-only algorithms may be selected.

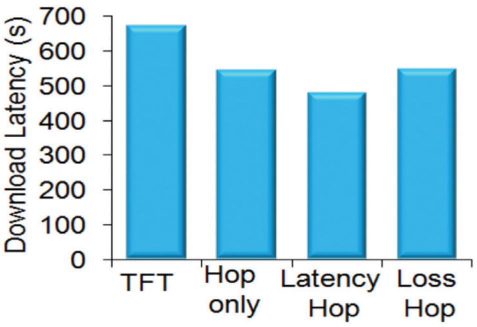


Figure 9 Download latency with different peer selection algorithms.

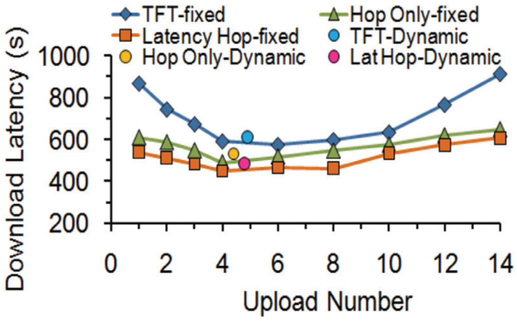
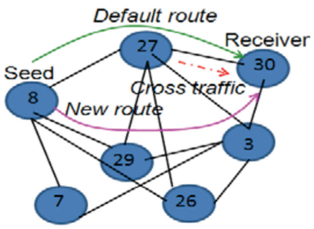


Figure 10 Download latency with and without dynamic adjustment.

Next, the dynamic adjustment of upload number was evaluated. **Figure 10** shows the average download latencies for TFT, Hop-only and Latency-Hop, for various fixed values of upload number (N). Note that N significantly impacts performance. Next, the dynamic adjustment algorithm is enabled, and the experiments are repeated. The average value of N over the course of the experiment was computed. We see that the dynamic adjust algorithm tends to converge to the optimal region of the upload number.

4.4 Proactive Peer Creation (PPC)

PPC was evaluated in CORE/EMANE using 5 seeds, each with one receiver, and with a file of size 4.5 MB. The seeds were chosen randomly and the receivers are chosen such that they were at least two hops away from the corresponding seeds. Cross-traffic at the rate of 750 Kbps was introduced in the routes between each seed-receiver pair, using IPerf UDP [3]. Five runs of the experiment were conducted, each with a different set of seeds and receivers. The latency threshold was set to 4000ms (exceeding that level triggers PPC).



	Avg. Download Latency (s)
No PPC	385.76
With PPC	299.68

Figure 11 PPC experimental setup and latency.

In the scenario of **Figure 11**, when PPC was activated, node *n29* was found to yield the least latency path between *n8* and *n30*, and hence was enabled as a peer. Subsequently, *n30* received most of its data via *n29*. Similarly, the other seed-receiver pairs enabled new peers as needed. **Figure 11** shows the average download latency per receiver with and without PPC. We see that PPC reduces the download latency by about 22%. Furthermore, the percentage of retransmissions in the network was 15.2% without PPC, but was reduced to 9.6% with PPC. The additional overhead due to the enabling of new peers was measured in terms of the number of control bytes per data byte (i.e., ratio of torrent control bytes to file size), which was 0.003 without PPC, and 0.007 with PPC.

5 Conclusions

We have developed a distributed torrent-based file distribution suite that operates efficiently without any infrastructure support. It is agnostic to underlying network routing protocols, and includes a fully decentralized content discovery component. Our evaluation shows that SISTO is significantly more robust both than conventional routing and infrastructure-based CDN approaches, and the addition of two key elements, i.e., adding network-awareness to peer selection, and proactive instantiation of peers to enrich routing diversity further improves robustness and throughput. We have also prototyped SISTO on an Android platform, and tested it in smart-phone networks envisaged for disaster-recovery, tactical or first responder networks.

References

[1] B. Cohen, “Incentives Build Robustness in BitTorrent”, P2PECON03.

[2] P. Maymounkov et al: “A Peer-to-peer Information System Based on the XOR Metric”, IPTPS, 2002.

[3] https://code.google.com/p/iperf/

[4] S. Cheshire et al., *Multicast DNS.* IETF RFC 6762.

[5] S. Das, et al, “SPAWN: A Swarming Protocol For Vehicular Ad-Hoc Wireless Networks”, VANET 2005.

[6] U. Lee et al, “CodeTorrent: Content Distribution using Network Coding in VANET”, MobiShare 2006.

[7] S. Srinivasan et al, “7DS - Node Cooperation and Information Exchange in Mostly Disconnected Networks, IEEE ICC 2007.

[8] M. Conti et al, “A cross-layer optimization of gnutella for mobile ad hoc networks”, MobiHoc 2005.

[9] A. Klemm et al, “A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks, IEEE VTC 2003.

[10] M. Schurgot, et al, “Providing local content discovery and sharing in mobile tactical networks”, IEEE MILCOM, 2013.

[11] Sewook Jung, et al. “BlueTorrent: Cooperative Content Sharing for BluetoothUsers”, IEEE Percom, White Plains, NY, USA, March 2007

[12] S. Ren, E. Tan, et al, “TopBT: A topology-aware, infrastructure-independent BitTorrent client”, IEEE INFOCOM 2010.

[13] M.K. Sbai, et al, “P2P content sharing in spontaneous multi-hop wireless networks”, COMSNETS, 2010.

[14] M. Kawarasaki, “Network-aware peer selection method for P2P file downloading using packet-pair measurement”, IEEE ICUMT, 2011.

[15] J. Sucec, et al. “A resource friendly approach for estimating available bandwidth in secure mobile wireless networks”, MILCOM, 2005.

[16] D. Shur et al, “SISTO: A proposal for serverless information services for tactical operations”, IEEE MILCOM 2012.

[17] J. Macker, I. Taylor, “INDI: Adapting the multicast DNS service discovery infrastructure in mobile wireless networks”, IEEE MILCOM 2011.

[18] http://www.nrl.navy.mil/itd/ncs/products/protosd

[19] A. Shrestha and F. Tekiner, “On MANET Routing Protocols for Mobility and Scalability”, International Conf on Parallel and Distributed Computing, 2009.

[20] *Libtorrent,* www.libtorrent.org

[21] Ivanic, N et al. “Mobile ad hoc network emulation environment.” *MILCOM 2009.* IEEE, 2009.

[22] Ahrenholz, J. et al. “Integration of the core and emane network emulators.” *MILCOM 2011.* IEEE, 2011.

[23] Chen, Ta, et al. “Enhancing application performance with network awareness.” *MILCOM 2011.* IEEE, 2011.

Biographies

K. Manousakis is a senior scientist with Applied Communication Sciences (ACS) since 2005. He has earned his MSc and PhD degrees from the Electrical and Computer Engineering Department of University of Maryland, College Park in 2002 and 2005, respectively. His expertise is in the areas of network protocols and network optimization. His recent interests involve content distribution for mobile networks, cognitive radios networking and interference mitigation. He has been a primary investigator (PI) for many US government funded programs like the ARL Collaborative Technology Alliance (Advanced Structures for Tactical Networks), the CERDEC NetMining (Dynamic Network Management for MANET), CERDEC Network Design (Offline and real time network planning tool) and the DARPA Fixed Wireless At a Distance (Content Distribution in Challenging Environments). Dr. Manousakis has patents and numerous publications in high profile conferences, journals and magazines in the areas of optimization techniques and algorithms for resource constrained mobile wireless networks. He has organized and chaired multiple IEEE sponsored conferences. Dr. Manousakis is the recipient of Award of Excellence in Telecommunications by Ericsson and in 2009 he has been awarded the IEEE PCJS Leadership Award.

S. Eswaran is a Research Scientist at Xerox Research Center India. Previously, she was a Senior Research Scientist at Applied Communication Sciences at Basking Ridge, NJ, where she worked on several research topics in the field of wireless and mobile networking. She received her Ph.D. degree in Computer Science and Engineering from Penn State in 2010, and M.S in Computer Engineering from Univeristy of Virginia in 2006. Her research focuses on optimal and robust data delivery across resource-constrained and variable wireless networks.

D. H. Shur received his Ph.D and MS degrees in Electrical Engineering from Stanford University in 1987 and 1981, respectively. From 1986–1996, he was with AT&T Bell Labs, Holmdel, NJ, where he was a Distinguished Member of Technical Staff. From 1996–2003, he was with AT&T Labs-Research, Middletown, NJ, where he was a Technology Consultant. From 2003–2008, he was with Bloomberg LP, New York, NY, where he was an Infrastructure Systems Architect. Since 2008 he has been with Telcordia Technologies/Applied Communication Sciences, where he is currently a Chief Scientist in the Mobile Networking Department. Dr. Shur has been the Principal Investigator for a number of research programs funded by DARPA, the US Office of Naval Research, and the US Army Communications-Electronics Research, Development and Engineering Center (CERDEC). His research has spanned the areas of IP-TV convergence (and was standardized in DOCSIS 3.0), real-time multimedia, IP over ATM, Packet Switching technologies, Electronic Watermarking, Web Caching, Multimedia Communication, Wireless Systems, Distributed Systems, Data Distribution, and Circuit Network/Packet Network convergence. He received the best paper award at IEEE WOWMOM in 2010.

G. Naik is a Senior Research Scientist with the Applied Informatics Group (AIG) at the Drexel University College of Computing & Informatics. His research interests in applied R&D include software defined networks, cyber security, and mobile computing. He has contributed to a number of open source projects and to community standardization working groups. Gaurav holds a M.S. degree in Computer Engineering.

P. Kantharaju is a graduate student studying Computer Science at Drexel's College of Computing and Informatics (CCI). He has a B.S. in Computer Science from Drexel University and his research interests include distributed computing and peer-to-peer networking.

W. Regli is a Professor (on leave) of Computer and Information Science in the Drexel University College of Computing & Informatics. He holds departmental appointments in Mechanical Engineering and Mechanics (College of Engineering); Electrical and Computer Engineering (College of Engineering); and in the College of Biomedical Engineering, Science and Health Systems. From 2007–2010 Regli was Senior Science Adviser to the National Institute of Justice's Communications Technologies Center of Excellence. He currently (2010–) serves as Senior Scientific Adviser to the Defense Programs Office of the National Nuclear Security Administration of the U.S. Department of Energy in areas of information technology for design, manufacturing, production in support of the stewardship and surety of the United States' nuclear weapons stockpile.

B. Adamson has been involved in research in radio communications and data networking at the Naval Research Laboratory (NRL) since 1984. His background includes digital signal processing and spread spectrum communications. Mr. Adamson's research interests include data and multimedia network transport, group communications, dynamic routing for wireless networks, and peer-to-peer networking. He has been focused on data networking for wireless and other networks for the past twelve years. This has included active participation in the Internet Engineering Task Force (IETF) in the areas of IPv6, reliable multicast transport, and Mobile Ad-hoc Networking (MANET). He also serves as co-chair of the Internet Research Task Force (IRTF) Network Coding Research Group (NWCRG). Mr. Adamson is also the principal author of several different network protocol implementations and network test, analysis, and visualization tools that have been applied in DoD, government, academic and commercial research and development, demonstration, and operational use.