Ad-hoc Storage Overlay System (ASOS): A Delay-Tolerant Approach in MANETs

Guang Yang[1], Ling-Jyh Chen[2], Tony Sun[1], Biao Zhou[1] and Mario Gerla[1]
[1]Computer Science Department, UCLA
{yangg, tonysun, zhb, gerla}@cs.ucla.edu
[2]Institute of Information Science, Academia Sinica, Taiwan
{cclljj}@iis.sinica.edu.tw

***Abstract*— Mobile Ad-hoc NETworks (MANETs) are a type of infrastructure-less networks that are most useful in unprepared emergencies where critical applications must be launched quickly. However, they often operate in an adverse environment where end-to-end connectivity is highly susceptible to various disruptions. Methods of adjusting the motion of existing nodes or deploying additional nodes can improve the connectivity under some circumstances, but there exist scenarios where connectivity cannot be immediately improved, and disruptions must be coped with properly. In this paper we propose an architecture of the Ad-hoc Storage Overlay System (ASOS). ASOS is a self-organized overlay, consisting of storage-abundant nodes that jointly provide distributed and reliable storage to disruption-affected data flows. ASOS is a Delay-Tolerant Networking (DTN) approach that extends the conventional end-to-end data transport model in MANETs and significantly improves their applicability in practice.**

***Index Terms*—Mobile Ad-hoc Networks, Delay Tolerant Networking, Network Overlay, Storage**

I. Introduction

Mobile Ad-hoc NETworks (MANETs) are a type of wireless networks that can be set up rapidly without pre-deployed infrastructure. MANETs are ideal where timely deployment of network infrastructure is impractical, e.g. on a military battlefield or in disaster recovery. Vehicular networks are also amongst the recent advances in MANETs that provide flexible data communications as alternative to infrastructure-based services. MANETs are most useful in unprepared situations where tasks must be fulfilled quickly. On the other hand, MANETs often operate in an adverse environment and are far less reliable than wired or infrastructure-based wireless networks. Nodes in a MANET can crash, lose power, be blocked by obstacles, or move beyond the communication range of its neighbors, etc. As a result, it is very difficult to guarantee continuous end-to-end connectivity in MANETs.

In recent years, research efforts have been made on maintaining end-to-end connectivity in MANETs. For example, popular ad-hoc routing protocols employ route redundancy and local repair to minimize the chance of end-to-end path breakage [12][13][19]. These mechanisms, however, only alleviate the problem by shortening the time of finding an alternative path when the original one breaks. If the network is partitioned into disconnected islands, there is little these routing protocols can do.

Other researchers have looked at the possibility of bridging disconnected partitions with additional nodes. For instance, [16] studied how to reconnect a partitioned network by deploying as few additional nodes as possible. [25][29] proposed using additional mobile nodes to relay messages between network partitions. These methods, when applicable, can improve connectivity fast. However, there still exist situations where additional nodes cannot be deployed fast as needed.

We believe that unavoidable disruptions should be properly *tolerated*. Naive methods would save the data at the source node or a centralized server. These methods obviously suffer various scalability and robustness issues. A more sophisticated scheme should involve replication and distributed storage mechanisms for better reliability and performance. In our proposed solution, named the Ad-hoc Storage Overlay System (ASOS), storage is handled by a *peer-to-peer (P2P) overlay* consisting of memory-abundant nodes in the MANET. These overlay nodes, called *ASOS peers*, jointly provide reliable storage to disruption-affected data flows. When an end-to-end flow is disrupted, ASOS receives data from the source node, stores it across the overlay, and delivers it to the

This material is based upon work supported in part by the National Science Foundation under Grants No. 0335302 and 0520332, and in part by the Office of Naval Research under Grant No. N00014-01-C-0016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Office of Naval Research.

original destination when connectivity improves. ASOS is self-organized; it complements the aforementioned approaches where additional nodes are used to bridge the partitions. ASOS is a Delay Tolerant Networking (DTN) approach [4][5] specifically designed for MANET scenarios. The concept can be integrated as part of the DTN reference framework [4].

The rest of the paper is organized as follows. Section II illustrates application scenarios for ASOS and discusses its design principles. Section III proposes the basic design of the ASOS architecture, focusing on overlay maintenance and service interfacing. Probabilistic data replication is studied in Section IV, along with our specific mobility model. Section V presents simulation results to assess the efficacy and performance of ASOS. Related work is summarized in Section VI. Finally Section VII concludes the paper.

II. Application Scenarios and Design Principles

A. Application Scenarios

ASOS is designed for Delay-Tolerant applications running in a heterogenous MANET environment. Most conventional applications such as file transfer are considered as delay tolerant; some multimedia applications are also (partially) delay tolerant as illustrated shortly. Heterogeneity means although some nodes in the MANET may operate with limited energy or storage, there exist powerful nodes with abundant resources, e.g. motor vehicles.

Assume a MANET is deployed for reconnaissance in a large area. A number of mobile nodes, humans and vehicles, are equipped with video cameras. Distributed across the area, these nodes capture useful data and send back to the control center. In the ideal case, all data is promptly received; complete and detailed images of the area can be quickly reconstructed. In reality, however, part of the data may not be delivered to the control center in time due to connectivity disruptions. The control center can still reconstruct images from partial data, but at a reduced quality with some details missing.

Without ASOS, undeliverable data is either dropped or buffered at the source node. If dropped, data is permanently lost; if buffered at the source, data is highly susceptible to the failure of a single point. Storage capacity of a node can also be limited. With ASOS, on the other hand, undeliverable data is jointly stored by ASOS peers with much more space and higher robustness to failures. The control center can retrieve the data later and add details to the images. In this way ASOS does not improve end-to-end connectivity instantly, but provides a mechanism to store useful data that will later contribute to the overall usefulness of the applications.

B. Design Principles

As we have explained, *when the source and destination nodes are separated in different partitions with no end-to-end connectivity, data should be stored in ASOS, an overlay storage utility, until connectivity improves, after which the stored data is delivered from ASOS to the original destination*. ASOS aims to extend data transport beyond the end-to-end model, with the following design principles:

PRINCIPLE 1: *safe and robust storage*. This is the top design principle. Data must be stored in a distributed manner with redundancy, and survive small-scaled failures.

PRINCIPLE 2: *immediate availability*. ASOS must be available as soon as possible in the events of network disruptions. This implies ASOS solely relies on the collaboration of existing nodes and is self-managed.

PRINCIPLE 3: *efficient storage and easy data delivery*. ASOS storage should be managed efficiently to hold as much data as possible. Also, data should be stored in a way to facilitate future delivery.

PRINCIPLE 4: *friendly interface*. ASOS should provide a friendly service interface to its storage utility. This includes both data submission from a source node, and data retrieval from a destination node.

III. The ASOS Architecture

To deal with lack of infrastructure, node heterogeneity and mobility, we design ASOS as a self-organized P2P overlay of existing nodes. Figure 1 illustrates the generic architecture of ASOS.

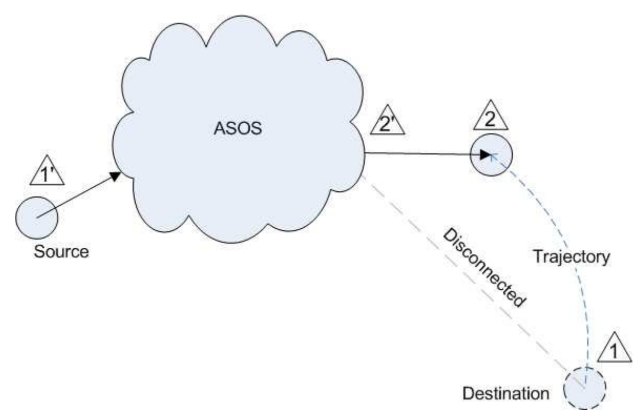

Fig. 1. ASOS architecture. When the destination is disconnected from the source (1), data is submitted to ASOS for storage (1'). Stored data is delivered to the destination (2') after it is reconnected (2).

A. Initialization and Maintenance of ASOS

1) Selecting ASOS peers: ASOS is a P2P overlay; potentially every node in the MANET can be a member, called an ASOS peer. Practically, it is desirable to designate only a set of powerful nodes with abundant storage as ASOS peers. We assume these nodes are pre-configured. Other nodes, called regular ones, must understand the interface of ASOS to access the storage utility.

2) Peer and file IDs: A number of P2P systems generate location-independent hash IDs for both peers and files; a file is stored at peers whose IDs best match the file ID [23][26]. This method automatically spreads files uniformly across geographical locations, a desirable feature for large-scale file sharing. This uniformity, however, is undesirable in ASOS, of which the primary goal is to deliver a file[1] to its original destination. We need an ID-independent algorithm to select storage locations for data, which will be studied in detail later in this paper. For now, we assume that every node has a unique ID. Files can be uniquely identified via hash mapping, e.g. from source/destination IDs and supplementary information such as TCP/UDP port numbers.

3) Initialization and maintenance: After a MANET is deployed, an initialization process is called to set up the ASOS overlay. All ASOS peers form a multicast group. For simplicity we assume that the multicast address is known *a priori* by all peers. Discussion on alternative methods such as dynamic address selection is beyond the scope of this paper. Shortly after deployment, each designated ASOS peer starts multicasting periodic `HELLO` messages to initialize and maintain the overlay. Essential fields of a `HELLO` message is illustrated in Figure 2. A `HELLO` message is sent to all ASOS peers, so that every peer can hear from all other reachable peers and know which files are stored in ASOS and where they are.

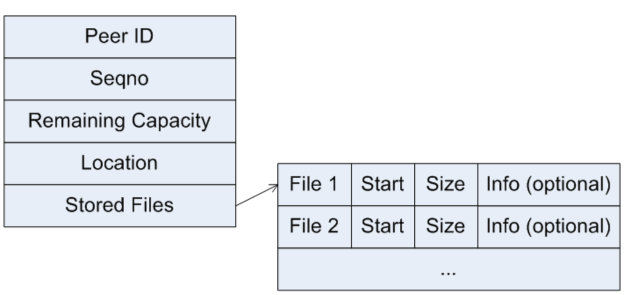

Fig. 2. Format of a `HELLO` message.

Several fields in the `HELLO` message are used for the maintenance of ASOS. First, *sequence number* is a peer-specific integer incremented every time a new `HELLO` message is sent. A new `HELLO` message with a higher sequence number refreshes information contained in previous `HELLO` messages from the same peer. *Remaining capacity* and *peer location* are used for data management in ASOS. *Stored files* are the meta data of the files, e.g. the file ID, start offset, file size and other optional information. We will explain them shortly.

To keep track of active peers and stored files, each ASOS peer maintains a lookup table of its reachable neighbors. Entries in the table contain similar fields as in the `HELLO` messages. Each entry corresponds to one peer and is refreshed when a new `HELLO` message from that peer is heard. Each entry is also associated with an expiration timer that is reset when the entry is refreshed. The initial value of the timer is larger than the refresh interval of `HELLO` messages. Expiration of a timer means that `HELLO` messages have not been heard from the particular ASOS peer for sufficiently long. This indicates that the peer may have become unreachable. The associated entry is then deleted from the lookup table.

B. ASOS Interface

1) Advertising of ASOS peers: Regular nodes in a MANET must know the existence of nearby ASOS peers. Due to the broadcast nature of wireless media, the `HELLO` messages exchanged between ASOS peers could also be used for advertising. By enabling the promiscuous mode, regular nodes could look at the `HELLO` messages to learn about the nearby ASOS peers and the currently stored files. Overhead is a serious drawback of this scheme, in which `HELLO` messages from all ASOS peers must be sent to all regular nodes in the MANET. Aggregation on `HELLO` messages would alleviate the problem, but could also intervene with ASOS maintenance. We decide to decouple the task of advertising from maintenance, and introduce a new type of `ADVERTISE` messages.

Each ASOS peer periodically aggregates its knowledge of reachable peers and stored files, and broadcasts an `ADVERTISE` message to all nodes. Since regular nodes do not participate in ASOS maintenance, fields such as *remaining capacity* and *peer location* are not included in `ADVERTISE` messages. More importantly, `ADVERTISE` messages only contain file IDs rather than the detailed locations of each file. This significantly reduces the size of the message. To suppress excessive broadcasting, one node only forwards the first fresh `ADVERTISE` message it receives. This is achieved by introducing a system-wide sequence number in all `ADVERTISE` messages. This

[1]For simplicity, hereafter in this paper we assume that data is managed in the form of files in ASOS. Alternatively data can be managed as database records, etc. The proposed ASOS architecture can be easily tailored for such cases.

mechanism also guarantees that a regular node receives ADVERTISE messages from its closest ASOS peer.

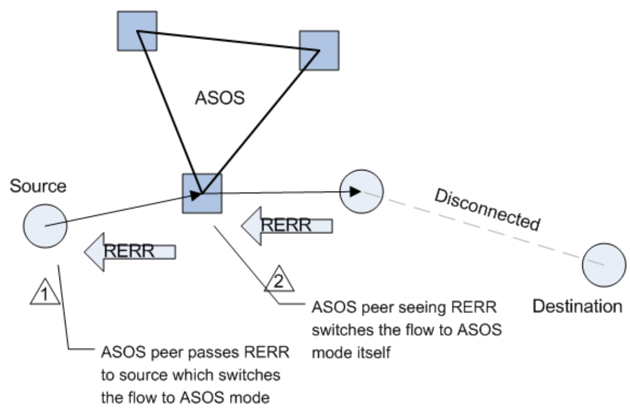

Fig. 3. Two methods to switch an end-to-end flow to the ASOS mode: by the source node or an intermediate ASOS peer.

2) Disruption detection and data submission: We design ASOS as a backup scheme when end-to-end connectivity is disrupted. Disruption is usually detectable at the routing layer. For example, popular ad-hoc routing protocols [12][13][19] use RERR messages to report route errors. Receipt of such an error message is a good indication of disruptions in connectivity. Note that many ad-hoc routing protocols perform local repair to fix a broken path before notifying the application. ASOS accommodates such efforts and will only intervene after local repair fails.

Upon disruption detection, there are two methods to switch an end-to-end flow to the ASOS mode. For the first, when the path between a pair of nodes is broken, the source node contacts the ASOS agent (known from ADVERTISE messages) once it receives a routing error message. Alternatively, if there is an ASOS peer on the path along which hears a routing error message, this ASOS peer can switch the flow to the ASOS mode immediately. The second method is faster, but the ASOS peer must notify the source node of the switching. See Figure 3 for an illustration.

We treat the entire data of an end-to-end flow as a file. Due to disruptions, transmission of the file may be divided into several chunks. An old chunk ends and a new one starts when 1) the end-to-end connectivity is disrupted and the flow is switched to the ASOS mode; 2) the current ASOS agent becomes unreachable and the source node finds a new one; and 3) the end-to-end connectivity has improved and the flow is switched back to the end-to-end mode. Therefore, the *start* offset in the file and the *size* of the chunk are critical information in the local lookup table. See Figure 4 for an illustration.

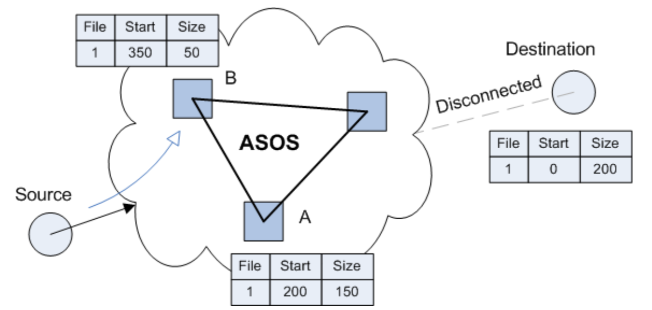

Fig. 4. Distributed storage of a file in ASOS. The first 200 data units of File 1 are delivered to the destination. The next 150 units are stored at an earlier ASOS agent A. The current ASOS agent is B with 50 units stored.

3) Data retrieval from ASOS: In terms of which party initializes the retrieval process, data can be either *pushed* to the destination node by ASOS, or *pulled* by the destination node itself. The *pull* model is applicable when a destination node, by receiving ADVERTISE messages, learns that ASOS has stored files for it. The *push* mode is applicable when an original destination node is seen by an ASOS peer, e.g. appearing in the peer's routing table.

C. Data Management

1) Probabilistic selection of storage locations: After a source node submits data to its ASOS agent, the agent becomes the first ASOS peer to hold a copy of the data. To increase storage reliability, data is also replicated to other ASOS peers. Intuitively, it is desirable to store data near the destination. For simplicity we assume that nodes in the MANET know their (relative) locations; this can be done with GPS or other localization mechanisms. If not available, geo-information may be substituted by other metrics such as the hop count.

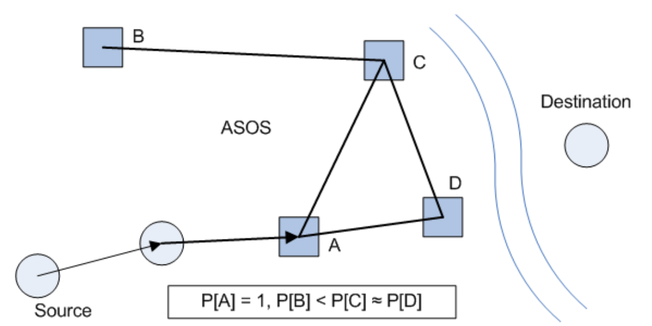

Fig. 5. Probabilistic replication of data. Node B has a lower but non-zero probability of holding a copy. Nodes C and D have comparable probabilities, though neither deterministically holds a copy.

A naive way to replicate data is to greedily push the data towards its destination. This method has a drawback: replicas will be geographically close to each other and vulnerable to clustered failures around the locale. More sophisticatedly, one can replicate the data probabilistically across the overlay. The ASOS agent selects among its reachable peer neighbors $K-1$ locations to replicate the data to, K being a configurable parameter. With the assumption that pairwise distances between nodes can be measured, storage locations are selected based on the following guidelines (also see Figure 5):

i) A peer closer to the destination node should have a higher probability to be selected,
ii) A peer further away from other ASOS peers that have been selected as storage locations should have a higher probability to be selected, and
iii) A peer less heavily loaded should have a higher probability to be selected.

The issue of probabilistic data replication will be studied in further detail in the next section.

2) Other data transfer between ASOS peers: In addition to the probabilistic data replication, an ASOS peer may also dynamically transfer stored data to another peer. This can happen when a peer is running short of power or storage space. A peer may replicate data when it detects that a former peer holding a replica has recently failed. A third situation is when the design parameter K, the number of copies of the data, needs to be increased.

3) Data deletion and replacement: ASOS supports both implicit and explicit data deletion and replacement, enabling it to deal efficiently in circumstances where data storage falls short of demand. In the explicit scheme, the original source or destination deletes the data from ASOS by messaging its ASOS agent. Data deletion in this case corresponds to situations where data is successfully delivered, or has lost its usefulness. The source node ID, destination node ID, and file ID are required to identify the file to be removed. The agent disseminates this message to the ASOS peers that currently hold a copy of the data, which will then delete the data from their local storage.

In the implicit scheme, ASOS can accommodate storage scarcity with prioritized storage management, such as the Least Recently Used (LRU) and First-in-First-out (FIFO) algorithms. ASOS removes data deemed less important to make storage space for more valuable data.

IV. Probabilistic Data Replication

We formulate the probabilistic data replication in ASOS as an optimization problem. Due to space limit we do not discuss the generic problem in this paper; for details please refer to [27]. Instead we turn to a specific mobility model for a concrete location selection algorithm.

A. Virtual Track Mobility Model

The Virtual Track (VT) mobility model [30] is used to mimic the mobility patterns of various MANET nodes. This model targets the scenario where mobility of the grouped nodes is constrained. It defines a set of "switch stations" and "tracks". Grouped nodes can only move on the tracks. Nodes belonging to the same group have the same group velocity; each node then has its own internal velocity with respect to the group.

Starting from its initial position, a group chooses to move to one end of the track, i.e. a switch station. After arrival, a new track is selected, along with a new group velocity. This process is repeated every time the group arrives at a new switch station. A group may split into multiple groups at a switch station and move forward in different directions. Multiple existing groups arriving at the same switch station within a certain time frame may also merge and move along together.

Other than the grouped nodes, randomly deployed static and individual nodes are also considered in the VT model. Static nodes are fixed with no mobility. Individual nodes move independently, not subject to any group velocities or track constraints. Figure 6 illustrates the concepts of the VT model.

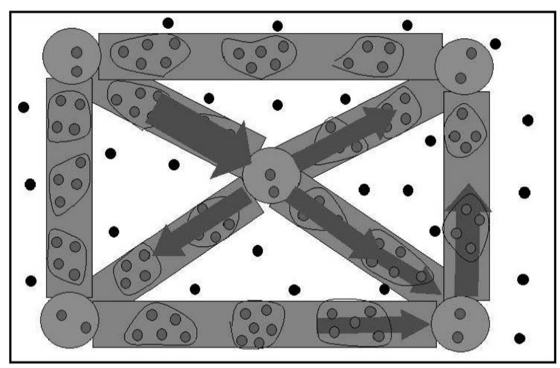

Fig. 6. The Virtual Track mobility model.

B. ASOS Peer Deployment and Probabilistic Location Selection under the VT Mobility Model

All three types of nodes, i.e. static, individual and grouped, can be considered as ASOS peers. Initially, each group contains a certain number of ASOS peers moving along with the regular nodes. These in-group peers respond fast when storage is needed. Due to splitting, a group may temporarily have zero ASOS peers. In this case regular nodes turn to ASOS peers in nearby groups, or to static/individual peers within reach.

Under the VT model, it is difficult to predict future connectivity or the distance between two nodes: the number of possible paths grows exponentially as nodes traverse across switch stations. Our approach is to use the *current* positions and velocities of ASOS peers to determine the replication locations. These values are disseminated in the latest HELLO messages. We assume that the current position and velocity of the destination node are also known. From such information, the next switch station where an ASOS peer or the destination node will arrive can be determined. We use the distance between two switch stations where two nodes will arrive, respectively, as the distance between these two nodes.

a: ASOS agent
z: original destination node
$\mathbb{P}$: set of reachable peers from a
$\mathbb{S}$: selected peers as storage locations, $\mathbb{S} \subseteq \mathbb{P}$
K: number of data copies (assumed as pre-configured)

ST_p: next switch station where p will arrive
$d_{p,q}$: distance between ST_p and ST_q
c_p: remaining capacity of peer p
$random(r_1, r_2)$: uniform random number generator on $[r_1, r_2)$

begin
  $\mathbb{S} \leftarrow \Phi$
  if $(|\mathbb{P}| < K)$
    $\mathbb{S} \leftarrow \mathbb{P}$
  else
    repeat
      for each $p \in \mathbb{P}$ **do**
        $d_{min} \leftarrow min\{d_{p,q} | q \in \mathbb{S}\}$
        $w_p \leftarrow (c_p \cdot d_{min})/d_{p,z}$
        $r_p \leftarrow random(0, w_p)$
      enddo
      $\mathbb{S} \leftarrow \mathbb{S} \cup \{p | p \in \mathbb{P}, r_p = max\{r_q | q \in \mathbb{P}\}\}$
      $\mathbb{P} \leftarrow \mathbb{P} \setminus \{p | p \in \mathbb{P}, r_p = max\{r_q | q \in \mathbb{P}\}\}$
    until $(|\mathbb{S}| = K - 1)$
  endif
end

TABLE I
ALGORITHM OF PROBABILISTIC SELECTION OF REPLICATION LOCATIONS AMONG ASOS PEERS.

The probabilistic location selection algorithm is shown as pseudo-code in Table I. The weight w_p favors an ASOS peer that is expected to be closer to the destination node, since due to spatial locality it is more likely to have a connection between them in the future. The algorithm also favors an ASOS peer further away from other replication locations. This improves the immunity against clustered failures. Finally, the algorithm favors an ASOS peer with more storage space, for the purpose of load balancing.

V. EVALUATION

A. Simulation Setup

We have implemented ASOS as an application module in the simulator of QualNet [20]. We select the UCLA campus map as the basis of our simulation scenario. Seven campus landmarks are identified as switch stations of the VT mobility model. A total of 30 nodes are deployed in the $1600\ m \times 1600\ m$ square area: 5 static, 5 individual and 20 grouped. Mobility of individual nodes follows the Random Way-Point (RWP) model; grouped nodes are divided into 4 groups of 5 nodes each. The screen snapshot from the QualNet Graphic User Interface (GUI) is shown in Figure 7. Nodes 1 to 5 are static, 6 to 10 are individual, 11 to 30 are grouped.

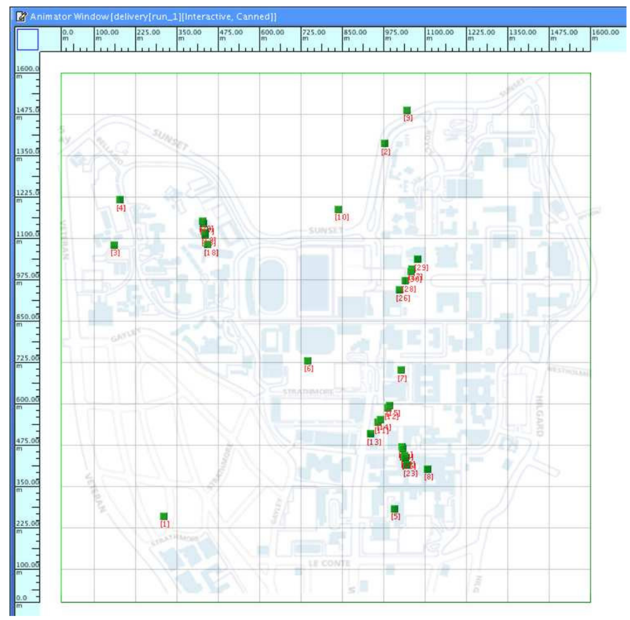

Fig. 7. Screen snapshot of the simulation topology in QualNet.

Node 30 is designated as the destination node where all data traffic is directed. Five flows are set up from different source nodes, including one static, one individual and three grouped nodes from each of the groups which node 30 does not belong to. Each data source generates a periodic constant-bit-rate (CBR) flow, at the rate of 80 $Kbps$, for 10 seconds every minute. Wireless communications between nodes use the IEEE 802.11b standard at 2 $Mbps$ with an effective transmission range of approximately 280 m. Each simulation runs 20 minutes. Data traffic stops at the 10^{th} minute; ASOS, if enabled, has an additional 10 minutes to exploit node mobility and deliver data to the destination. Simulation parameters are summarized in Table II.

B. Delivery Ratio

We first compare the instantaneous throughput in non-ASOS and ASOS scenarios. Hereafter in this paper, the non-ASOS scenario means data is always delivered in the conventional end-to-end fashion, while the ASOS sce-

topology	$1600m \times 1600m$
simulation time	$20\ min$
total number of nodes	30
number of static nodes	5
number of individual nodes	5
number of groups	4
number of nodes in each group	5
number of static nodes as ASOS peers	2
number of ind. nodes as ASOS peers	2
number of ASOS peers in each group (excluding the group with node 30)	2
number of replicas	3
interval between `HELLO` messages	$10\ sec$
interval between `ADVERTISE` messages	$10\ sec$
expiration time of entries in lookup tables	$30\ sec$
RWP model min speed	$0\ m/sec$
RWP model max speed	$10\ m/sec$
VT model group min speed	$0\ m/sec$
VT model group max speed	$10\ m/sec$
VT model internal min speed	$0\ m/sec$
VT model internal max speed	$1\ m/sec$
VT model max track length	$750\ m$
VT model group split prob.	0.25
nominal 802.11b data rate	$2\ Mbps$
approx. transmission range	$280\ m$
number of flows	5
average per-flow data rate	$13.33\ Kbps$
start time of flows	$0^{th}\ min$
stop time of flows	$10^{th}\ min$

TABLE II

SUMMARY OF SIMULATION PARAMETERS IN QUALNET.

nario means data is normally delivered end-to-end but will switch to the ASOS mode when connectivity is disrupted.

Figure 8 shows the instantaneous throughput measured every minute at the destination node. Ideally, this throughput should be 67 $Kbps$, the aggregate sending rate from all sources, for the first 10 minutes. Due to the disruptions however, throughput in the non-ASOS scenario is consistently below the ideal value. ASOS does not instantly increase end-to-end connectivity in the MANET; instead undeliverable data is temporarily stored, and delivered when connectivity improves. Clearly reflected in the figure, at the 8^{th} minute the instantaneous throughput is well above 67 $Kbps$. This includes both fresh data produced during the minute, as well as previously stored data. After the 10^{th} minute when all flows have stopped, some amount of data can still be delivered to the destination, e.g. at the 11^{th} and 17^{th} minute. Figure 9 shows the cumulative amount of data delivered to the destination as time proceeds. At the end of the simulation, ASOS is able to deliver about twice as much data as delivered in the compared non-ASOS case. We have repeated the simulation with different random seeds; all runs have shown the similar trend.

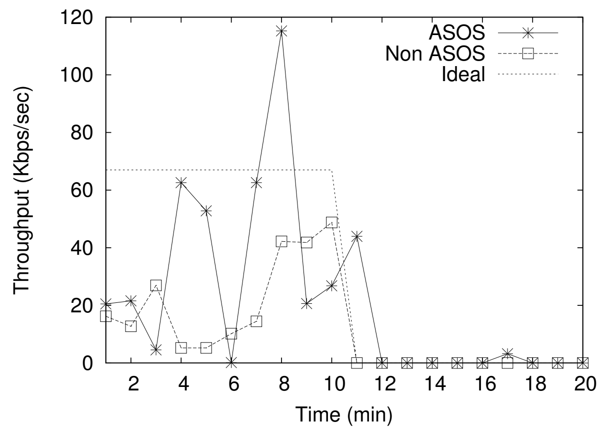

Fig. 8. Instantaneous throughput measured at the destination node.

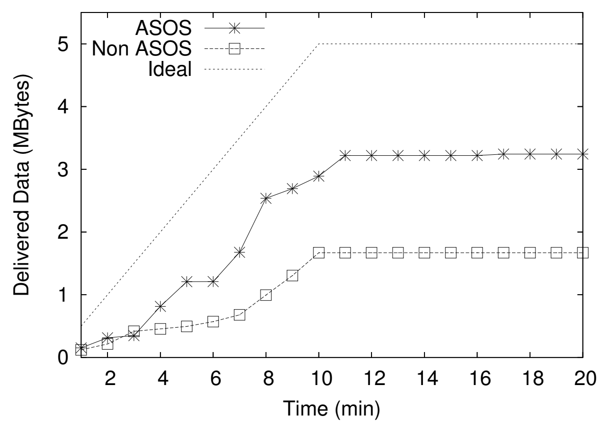

Fig. 9. Cumulative amount of data delivered to the destination as time proceeds.

We are also interested in how individual flows benefit from ASOS. For this purpose we compare in Figure 10 the delivery ratios of three flows in ASOS and non-ASOS cases, respectively: one from a static node (left), one from an individual node (center), and one from a grouped node (right). From results depicted in Figure 10, the delivery ratio for the grouped node has improved the most with ASOS, while the static node has improved the least. This coincides with our expectation. Static nodes, scattered across the area and stay still, lack the ability of moving to see other nodes and participate in/benefit from ASOS. Individual nodes can move, but their territory is not constrained by the tracks and can be much larger than the grouped nodes. Grouped nodes only move on the tracks; this effectively reduces the area of their territory and increases the chance of seeing each other. Therefore, we have observed the best delivery ratio improvement on grouped nodes.

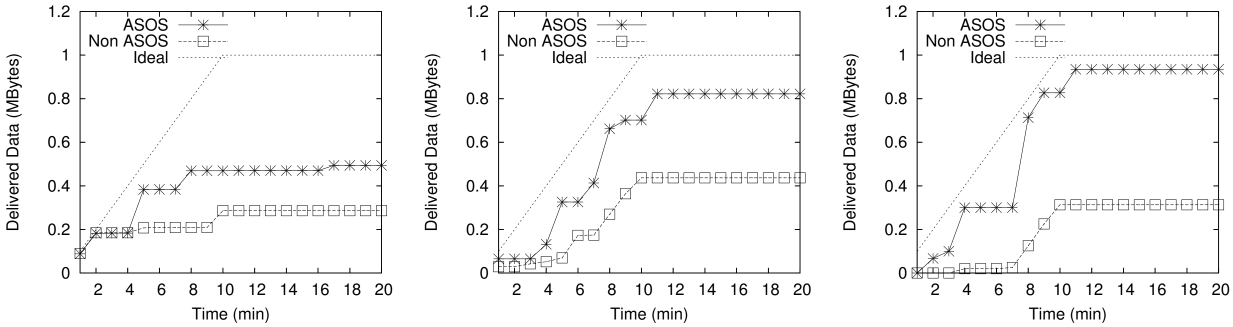

Fig. 10. Delivery ratios of data generated in each minute. Three flows are shown, one from a static node (left), one from an individual node (center), one from a grouped node (right).

C. *Messaging Overhead*

Both HELLO and ADVERTISE messages in ASOS incur overhead, with HELLO messages being the heavier source since they are generally larger with more detailed information. In Figure 11 we plot the cumulative distribution function (CDF) of the size of HELLO messages. Over 70% of the HELLO messages are less than 200 bytes long, with a maximum of 460 bytes only. In our simulation where 10 ASOS peers send out HELLO messages every 10 seconds, the control traffic injected is merely $1.6\ Kbps$. This is negligible compared to the data traffic. It is worth the expense to significantly increase the data delivery ratio with ASOS.

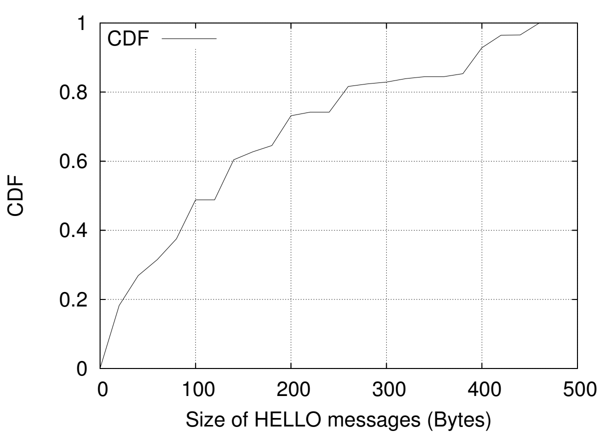

Fig. 11. Cumulative distribution function (CDF) of the size of HELLO messages.

D. *Impact of ASOS Parameters*

Two key parameters in ASOS are the number of ASOS peers deployed in the MANET, and the number of data replicas, i.e. K. So far they have been fixed at 10 and 3, respectively. We now vary these numbers and study the impact on ASOS.

First we vary the number of ASOS peers from 5 to 20 in the 30-node scenario in Figure 7; the results are shown in Figure 12. At the beginning, the delivery ratio increases with the number of ASOS peers. This is obvious since more ASOS peers provide better availability to all regular nodes. However, the increasing delivery ratio quickly reaches the peak and then starts to decline as more ASOS peers are added. This is mainly due to the fact that excessive ASOS peers incur significantly more messaging overhead, which negates the marginal gain brought by these additional ASOS peers.

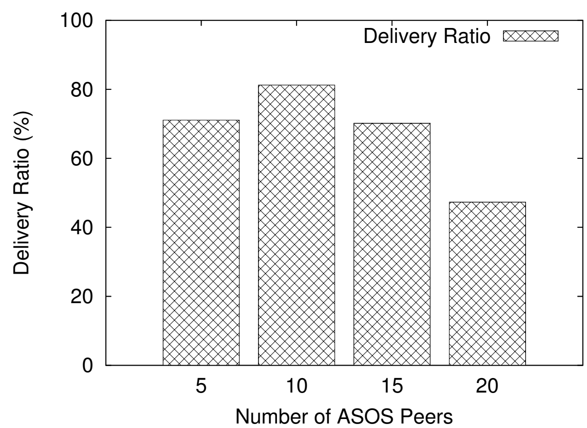

Fig. 12. Delivery ratio vs. number of ASOS peers. Deliver ratios are the overall values of all flows directed to the destination.

We then fix the number of ASOS peers at 10 and vary the number of replicas from 1 to 5. The results are presented in Figure 13. At the beginning, the delivery ratio grows with the number of copies, but quickly hits the plateau; further increasing the number has little impact on the delivery ratio. This is different from what we have observed in Figure 12. The main reason is that increasing the number of ASOS peers leads to *more* HELLO and ADVERTISE messages, while increasing the number of data copies leads to *larger* HELLO and ADVERTISE messages. Since these messages are relatively small in our simulation scenario, an increase in the number of messages has larger impact on the traffic load than an increase in the message size.

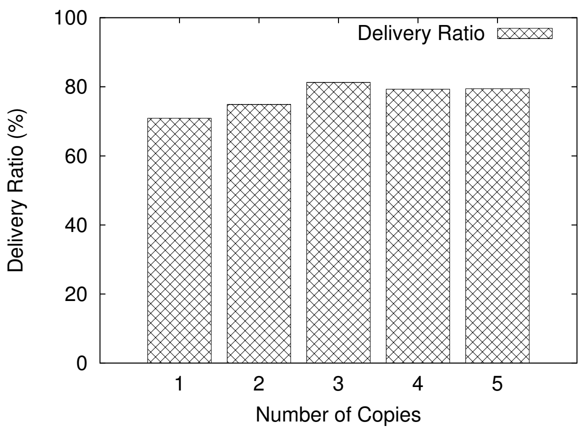

Fig. 13. Delivery ratio vs. number of replicated data copies. Deliver ratios are the overall values of all flows directed to the destination.

Figures 12 and 13 indicate that choosing the appropriate number of ASOS peers and number of data replicas has big impact on the ASOS performance. In general this depends on a number of factors such as the topology, mobility and traffic patterns. We do not intend to explore an optimal solution for the problem in this paper; a comprehensive investigation is left for future study.

VI. Related Work

We now summarize the status of current research efforts related to our work. The DTN Research Group (DTNRG) [4] is dedicated on providing inter-operable communications in challenged networks [6][10][11], which has enlightened us on developing a P2P storage overlay architecture to cope with frequent disruptions in MANET scenarios. Among other DTN efforts for MANETs, [9] proposes using controlled flooding in sparse mobile networks for packet delivery when end-to-end paths do not exist. [15] studies several different opportunistic forwarding strategies using mobile routers (similar to the mobile ferries [29]) in vehicular ad-hoc networks. Adaptive routing for intermittently connected MANETs is investigated in [18]. Our work differs from these efforts and can be integrated as an added feature in MANETs where the above schemes are already implemented.

P2P overlay storage is widely studied on the Internet. Cooperative File System (CFS) has been proposed in [3]; PAST [22] is another P2P storage utility. These systems are good paradigms on P2P storage; however, our target problem is different. CFS and PAST aim to provide fast file access to all potential users on the Internet. Files are likely to be replicated geographically to many distant locations, a desirable property for robust file sharing. On the contrary, ASOS aims to extend the conventional end-to-end delivery model where stored data is to be delivered to the intended original destination only.

In sensor networks, Data Centric Storage (DCS) [21] is proposed to save all data of the same type at a designated storage server, such that data queries are directed only to the server. The basic DCS is extended with *local refreshes* and *structured replication* for better data robustness and system scalability. Resilient DCS (R-DCS) [7] further improves the robustness and scalability by replicating data at strategic locations. DCS/R-DCS share many commonalities with the idea of ASOS; the main differences are: 1) DCS/R-DCS use geographic information mostly for routing while ASOS uses it for location selection, and 2) data replication is deterministic in DCS/R-DCS but probabilistic in ASOS.

In the context of MANETs, [2] has studied scalable P2P computing. A cross-layer design is proposed to provide better interaction between P2P addressing and ad-hoc routing. Mobile Information Retrieval (IR) is studied in [8]. It splits, indexes and replicates a given database probabilistically across all mobile nodes, and a node only contacts its 1-hop neighbors to answer a query. This scheme is claimed to be sufficiently accurate and robust against network partitioning. PAN [17] is another probabilistic storage system in MANETs. In PAN, small data objects are replicated to multiple servers and can be dynamically updated. To control overhead, PAN "lazily" updates data in multiple rounds such that inconsistency may temporarily exist. It is difficult to apply Mobile IR and PAN directly to ASOS for two reasons. First, the database in mobile IR is static, while in ASOS data is produced in real time. Second, both mobile IR and PAN replicate data to improve chance of access from *all* nodes; this is unnecessary and undesirable in ASOS.

Cooperative caching has also been studied for MANETs. [14] proposes adding an application manager component between the network layer and applications. The application manager will switch to a better data source if the QoS provided by the current one is degrading. This requires that at least one data source is available at any time, which cannot be guaranteed in our target scenarios. [24] and [28] have investigated mechanisms of caching popular data among ad-hoc nodes, such that when the original source is not available, data can be obtained from nearby nodes. These studies have revealed valuable insight into distributed data storage in MANETs; however, they do not target the problem of extending end-to-end flows to survive disruptions.

VII. Conclusion

In this paper we have proposed the Ad-hoc Storage Overlay System (ASOS) that extends end-to-end data transport in MANETs when connectivity is disrupted. ASOS is a DTN approach to tolerate inevitable disruptions in MANETs, by storing undeliverable data reliably in an overlay of storage-abundant nodes and delivering it later when connectivity improves. We have implemented the ASOS in QualNet and shown that it can significantly increase the overall data delivery ratio in disrupted MANET scenarios.

In the future, we plan to investigate effective integration of ASOS with the DTN reference implementation [4]. We also plan to address a few open issues. First, we have only considered pre-configured ASOS peers in this paper. Alternatively ASOS peers can be dynamically elected, e.g. when no ASOS peers are within the reach of regular nodes. Second, in order to better support multicast applications, the current data management and interface in ASOS need to be revised. Erasure codes can also be incorporated into ASOS to improve data reliability. Also, data encryption, user authentication and intrusion detection are needed when ASOS operates in a hostile environment against malicious attackers. Finally there exist "soft disruptions" where end-to-end connectivity is not totally lost but can be maintained at a reduced effective capacity. We are interested in upgrading ASOS for such scenarios.

References

[1] S. Chessa and P. Maestrini, "Dependable and Secure Data Storage and Retrieval in Mobile, Wireless Networks", DSN'03, San Francisco, CA, June 2003.

[2] M. Conti, E. Gregori and G. Turi, "Towards Scalable P2P Computing for Mobile Ad Hoc Networks", IEEE PerCom Workshops 2004, Orlando, FL, March 2004.

[3] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris and I. Stoica, "Wide-area cooperative storage with CFS", ACM SOSP 2001, Banff, Canada, October 2001.

[4] Delay Tolerant Networking Research Group, `http://www.dtnrg.org`.

[5] Disruption Tolerant Networking, `http://www.darpa.mil/ato/solicit/DTN/`.

[6] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.

[7] A. Ghose, J. Grossklags and J. Chuang, "Resilient data-centric storage in wireless ad-hoc sensor networks", The 4th International Conference on Mobile Data Management (MDM), January 2003.

[8] K. M. Hanna, B. N. Levine and R. Manmatha, "Mobile Distributed Information Retrieval For Highly-Partitioned Networks", IEEE ICNP 2003, Atlanta, GA, November 2003.

[9] K. Harras, K. Almeroth and E. Belding-Royer, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks", IFIP Netwoking 2005, Waterloo, Canada, May 2005.

[10] S. Jain, K. Fall and R. Patra, "Routing in a Delay Tolerant Networking", ACM SIGCOMM 2004, Portland, OR, August/September 2004.

[11] S. Jain, M. Demmer, R. Patra and K. Fall "Using Redundancy to Cope with Failures in a Delay Tolerant Network", to appear in ACM SIGCOMM 2005, Philadelphia, PA, August 2005.

[12] D. B. Johnson, D. A. Maltz and Y.-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", IETF Internet draft, draft-ietf-manet-dsr-09.txt, April 2003.

[13] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", ACM MobiCom 2000, Boston, MA, August 2000.

[14] W. Lau, M. Kumar and S. Venkatesh, "A Cooperative Cache Architecture in Support of Caching Multimedia Objects in MANETs", ACM WoWMoM'02, Atlanta, GA, September 2002.

[15] J. LeBrun, C. N. Chuah and D. Ghosal. "Knowledge Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks", IEEE VTC Spring 2005, Stockholm, Sweden, May 2005.

[16] N. Li, and J. C. Hou, "Improving Connectivity of Wireless Ad-Hoc Networks", UIUC DCS Technical Report UIUCDCS-R-2004-2485, October 2004.

[17] J. Luo, J-P Hubaux and P. T. Eugster, "PAN: Providing Reliable Storage in Mobile Ad Hoc Networks with Probabilistic Quorum Systems", ACM MobiHoc 2003, Annapolis, MD, June 2003.

[18] M. Musolesi, S. Hailes and C. Mascolo, "Adaptive Routing for Intermittently Connected Mobile Ad Hoc Networks", IEEE WoWMoM 2005, Taormina, Italy, June 2005.

[19] C. E. Perkins, E. M. Belding-Royer and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing", IETF Internet draft, draft-perkins-manet-aodvbis-00.txt, Oct 2003.

[20] QualNet, `http://www.scalable-networks.com/`.

[21] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, L. Yin and F. Yu, "Data-Centric Storage in Sensornets", `http://lecs.cs.ucla.edu/Publications/papers/dht.pdf`.

[22] A. Rowstron and P. Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility", ACM SOSP 2001, Banff, Canada, October 2001.

[23] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems", IFIP/ACM Middleware 2001, Heidelberg, Germany, November 2001.

[24] F. Sailhan and V. Issarny, "Cooperative Caching in Ad Hoc Networks", The 4th International Conference on Mobile Data Management (MDM), January 2003.

[25] R. Shah, S. Roy, S. Jain and W. Brunette, "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks", Intel Research Technical Report, IRS-TR-03-001, January 2003.

[26] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", ACM SIGCOMM 2001, San Diego, CA, August 2001.

[27] G. Yang, L-J. Chen, T. Sun, B. Zhou and M. Gerla, "Ad-hoc Storage Overlay System (ASOS): A Delay-Tolerant Approach in MANETs", UCLA CS Technical Report 050030, July 2005.

[28] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks", IEEE Infocom 2004, Hong Kong, China, March 2004.

[29] W. Zhao, M. Ammar and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks", ACM MobiHoc 2004, Roippongi, Japan, May 2004.

[30] B. Zhou, K. Xu and M. Gerla, "Group and Swarm Mobility Models For Ad Hoc Network Scenarios Using Virtual Tracks", MILCOM 2004, Monterey, CA, October/November 2004.