Enhanced Disruption and Fault Tolerant Network Architecture for Bundle Delivery (EDIFY)

Mooi Choo Chuah, Liang Cheng, Brian D. Davison

Department of Computer Science and Engineering
Lehigh University
19 Memorial Drive West, Bethlehem, PA 18015
{chuah, cheng, davison}@cse.lehigh.edu

***Abstract*—Data communication challenges exist in some emerging network scenarios where an instantaneous end-to-end path between a source and destination may not exist, and the links between nodes may be opportunistic, predictably connectable, or periodically-(dis)connected. We propose an enhanced disruption tolerant network architecture to address such challenges. In this paper, we present a generalized naming convention for the enhanced DTN architecture that permits separate representations based on network topology, admini-strative control, physical location, and other factors. In addition, we illustrate possible system operations in this enhanced DTN architecture such as DTN neighbor discovery, gateway selection, mobility management, and route discovery.**

***Keywords*—disruption tolerant networks; flexible naming; mobility management; route discovery**

I. Introduction

The Internet has been a great success at interconnecting communication devices across the globe. Most Internet applications are based on the existing TCP/IP-based protocol suite. Using a packet-switched model of service, the IP protocol is mapped into network-specific link-layer data frames at each router and hence an end-to-end connection can span networks of different technologies, e.g., ATM, frame relay, ISDN, telephone and cellular networks. Current Internet service models rely on a few key assumptions to provide useful services, namely,

(a) An end-to-end path exists between a source and destination pair;

(b) The maximum round-trip time between any node pair is not excessive;

(c) The end-to-end packet drop probability is small; and,

(d) Communication links have relatively symmetric bidirectional data rates.

However, emerging network scenarios are challenging these assumptions. In such scenarios, an instantaneous end-to-end path between a source and destination may not exist, and the links between nodes may be opportunistic, predictably connectable, or periodically-(dis)connected. Some examples are described as follows.

Mobile networks. A commuter bus installed with wireless modem may only have intermittent RF connectivity at various terminals as it travels from place to place, but it can act as a store and forward message switch for bus riders to send email, etc. Other forms of mobile carrier are reported in [1][3].

Battlefield ad hoc networks. These systems operate in hostile environments where jamming, environmental factors and mobility may cause temporary disconnections.

High latency networks. Near-earth satellite communications and very long distance radio or wireless optical links may be subject to high latency with predictable disruptions, e.g., due to planetary dynamics. Such communications may also suffer outage due to environmental conditions such as weather and solar flare activity.

II. Challenges of Emerging Network Scenarios

These challenging network scenarios in general have the following common characteristics: the latency, available bandwidth, or path stability is substantially worse than what is typical in today's Internet. Some of these characteristics are elaborated below so that one can understand the requirements that a new architecture design for disruption tolerant networks (DTN) should address.

A. *End System Characteristics*

In some networks, end nodes are placed in hostile environments, e.g., sensor networks, military networks, and networks used by emergency response teams. In such cases, network nodes may not last long and networks may be disconnected for long periods of time. The conventional end-to-end acknowledgement schemes are not useful for such network scenarios. Instead, it may be more appropriate to delegate to some other party (that is still operational) the responsibility of delivering the message.

In addition, small devices like sensor nodes have limited battery power. Hence, their communication patterns may have to be scheduled *a priori* to ensure a low duty cycle of 1-2% and hence the longevity of the entire network. Small devices additionally have limited memory resources. It is undesirable for such devices to keep a copy of their sampled data until it can be acknowledged by the sink since the end-to-end delay may be prohibitively long.

B. Path and Link Characteristics

In some networks, the link bandwidth may be as low as 10-20 Kbps (e.g., low-power sensors or underwater acoustic links). Data rates may also be asymmetric, e.g., satellite links with a high downlink data rate but low uplink data rate. In extreme cases, there may not be any return channel, e.g., in covert military operations. In addition, we may also have frequent disconnections as a result of motion or battery power exhaustion. Disconnections due to motion may be predictable (e.g., interplanetary dynamics) or unpredictable (due to nodes moving out of communication range). Furthermore, we may have long queuing times, e.g., when next hop routers are not reachable or when networks become temporarily partitioned.

C. Enhancements to Existing Protocols are Insufficient

To adapt Internet services to emerging ad hoc environments, one approach is to make the problematic links look more like the types of links for which TCP/IP was designed. Some examples of so-called "link-camouflage" approaches are described in [4], e.g., using reliable link-layer protocol, using split TCP connections, and end-to-end explicit loss notification. Disadvantages of the "link-camouflage" approach include (a) the enhancements may work well in one environment (e.g., high packet loss rate or LAN environment) but not in another (e.g., highly variable link bandwidth availability or WAN environment); and (b) the technique still requires an end-to-end path to exist which may be an invalid assumption in network environments where network elements may be partitioned for long periods. Another approach is via performance enhancing proxies [5] and application-layer proxies [6]. However, such proxies may be specific to a particular application, may not work with IPSEC, and do not include an inter-proxy routing capability.

Electronic mail [14] provides an abstraction that comes close to addressing many of the problems posed by the challenging network scenarios [7]. Its flexible naming, asynchronous message-based operation, and in-band error reporting are useful features that enable it to run over a rich set of network technologies. However, email falls short due to its lack of dynamic routing, and weakly-defined delivery semantics. Email delivery seems to be "mostly reliable delivery" with "occasional failure" notification. Upon failure, the original message and accumulated errors are generally returned to the sender but the sender has little direct ability to correct the problem.

D. Noticeable Holes in Existing DTN Proposals

From the above discussion, it is clear that a new architecture is needed that can combine some overlay routing capability with the delay-tolerant and disconnection-tolerant properties of electronic mail. A new overlay architecture called *Delay Tolerant Networking* has been proposed in [7] to provide virtual message switching capabilities with limited expectations of end-to-end connectivity and node resources.

In the existing delay-tolerant networking proposal [7], the network is divided into different regions and the regions are connected by gateways. A gateway that spans two regions consists logically of two halves, each half in one of the adjacent regions above their corresponding transport protocols. Gateways are responsible for storing messages in nonvolatile storage when reliable delivery is required, and mapping between differing transports by resolving globally-significant name tuples to locally-resolvable names for traffic destined to an adjacent region.

However, we believe that such a naming convention—while useful for stationary delay tolerant network scenarios—may not be able to deal with ad-hoc mobile environments that battlefield networks often face. In battlefield networks, military personnel often form an ad hoc network and move together as a group. Often, the group may be forced by environments, e.g., hills or enemy attacks to be split into disconnected groups. Nodes in other groups/regions which wish to communicate with such a partitioned group require a better naming convention than what is currently proposed in [7]. Even when communication links are only temporarily disconnected, networking services within a region may be disrupted.

In our research, we design an enhanced disruption tolerant network architecture to address the above-mentioned challenges and unsolved issues in DTN network design. We refer to our enhanced architecture as the Enhanced Disruption and Fault Tolerant Bundle Delivery (EDIFY) system. In this paper, we present a generalized naming convention for the enhanced DTN architecture that permits separate representations of network topology, administrative control, physical location, and other factors. This allows for bundle routing preferences or requirements to be expressed as functions of a (possibly incomplete) name. It also permits extensions to incorporate service operations within the naming construct. Networks that are partitioned can get new names dynamically while retaining their old identities so that information can still be delivered if needed. Details are provided in Section III. In addition, Section IV illustrates system operations in this enhanced DTN architecture. Section V concludes this position paper.

III. FLEXIBLE NAMING CONVENTION

Our DTN naming convention allows for role-based addressing and multiple namespaces. It provides layered resolution of address and routing information. We illustrate the hierarchical naming convention in our enhanced DTN architecture in Figure 1. We show four groups: three of which belong to US-DOD and one is a NATO squad team made up of army personnel from US, UK and France. Two of the three US-DOD teams are from US-DOD.Army while the third one is from US-DOD.Navy. There is a platoon member (UserHost-1093) that is currently with the US-DOD.Navy.Battalion5. This platoon member can be given a visiting identifier like US-DoD.Navy.Battalion5.Visitor5. Information is kept at US-DoD.Navy.Battalion5.GW1 that there is a visitor from US-DoD.Army.Platoon44. Similarly, information is kept at US-DoD.Army.Platoon44.GW3 that one of their members is at US-DoD.Navy.Battalion5. Whenever there is any broadcast message for Platoon44, US-DoD.Army.Platoon44.GW3 will send a copy to US-DoD.Navy.Battalion5.GW1 to be delivered to UserHost-1093. In Fig. 1, we also show an example of a squad that consists of army personnel from some NATO countries. The squad members each have their own original identity as well as a temporary identity from the squad.

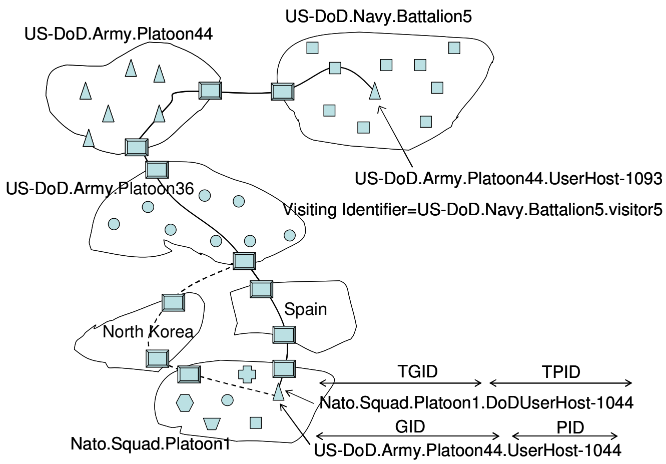


Figure 1. A flexible hierarchical naming convention for DTN

This naming convention supports policy-based routing. DTN nodes are configured with individual and domain-wide routing policies that determine the conditions for determining the routing approach at any particular time. The routing policies can mandate the preferred domains for the bundles to go through and those domains that should be avoided due to security or cost reasons. An example is shown in Fig. 1 where bundles from US.DoD.Army prefer to be routed via the Spain domain than via the North-Korea domain even though both routes can deliver the bundles to the squad.

Real-world DTNs will need to incorporate mobile and *ad hoc* groups of all sizes. Instead of regions, we provide the concept of groups, e.g., an army platoon can be a group. As in [7], each group has a group identifier (GID) and each entity within a group has its own personal identifier (PID). Any device within the group can be identified with the appropriate tuple (GID, PID).

Like [7], we choose to use a hierarchical naming technique for groups. This allows both for scaling (since there will be large numbers of groups) and to better map real-life complexity, such as geographical location or an administrative hierarchy. For example, instead of naming a node as (RegionA, UserHost-1093), we use a structured group such as (US-DoD.Army.Platoon44, UserHost-1093). Such naming may provide additional routing hints (such as preferring a gateway to the longest-matching prefix).

Unlike [7], we go further, and generalize naming to permit multiple, different naming hierarchies. This allows us to incorporate information from multiple naming systems, including those based on network topology, network administration, physical location, and more. For example, in addition to being (US-DoD.Army.Platoon44, UserHost-1093), this node might also have a geographic name of (US.NJ.Monmouth, P44-UserHost-1093) while stationed at Fort Monmouth, but would change when deployed abroad.

Members of different groups can form an *ad hoc* group which adopts a different group identifier denoted as TGID (Temporary Group ID). Members of such an ad hoc group will assume two identifiers, namely the original (GID, PID) as well as (TGID, TPID). When they intend to communicate with the ad-hoc group, they will use the identifier (TGID, TPID) but when they intend to communicate with the original group member, they will use the identifier (GID, PID).

In addition, every user (or device) can have a canonical, universally unique name. Such an entity name would be fixed. The other names (e.g., US-DoD.Army…) are only 'temporary' assignments of location, or administrative position, etc., but they do correspond to hierarchical groups, allowing for scalable routing (which might not be possible with the canonical name). This canonical name, if it refers, for example, to the person using or reachable with this device, can move from system to system as the person moves from home desktop to mobile phone to work desktop, etc. Likewise, even when a node moves from one group to another, the canonical name can stay fixed. Such naming suggests the creation of supporting services, such as to resolve a canonical name to its last known set of non-canonical naming tuples so that a message can be properly addressed (that is, with one or more names that are routable). Services like name resolution may be deployed via supporting infrastructure or as additional responsibilities of participating DTN nodes. It also suggests that an individual node might represent more than one entity (person), each with a canonical ID and routable naming tuples.

IV. System Operations in the Enhanced DTN Architecture

In our work, we assume that not all nodes participate in our enhanced DTN architecture. Thus, the nodes that participate in this architecture look like an overlay network over existing legacy networks. We further assume that each group runs its own preferred routing protocols internally but those nodes that participate in the DTN perform DTN neighbor discovery, DTN gateway selection, DTN mobility management, and DTN route discovery described in subsequent subsections.

A. *Neighbor Discovery*

Both infrastructure-based and infrastructureless networks are considered in our DTN design. Individual networks may be in wired or wireless domains. An infrastructureless DTN node needs to determine its location and neighbors upon initialization. Thus, in a wired domain where only some nodes support DTN functionality, the DTN nodes can discover one another using an approach similar to peer-to-peer network. They can send a neighbor discovery message with a TTL of 1 to the designated multicast address to which every DTN node will listen. We refer to this multicast address as the "Neighbor Discovery Multicast Address". Any node that hears such a message should respond with a Node Announcement message. If the new node does not hear any response, it sends another neighbor discovery message with increasing TTLs until a sufficient number of responses are heard. Each node announcement message may contain (*i*) name tuple(s), (*ii*) node-type (whether the node is regular node or gateway node or message ferry), and (*iii*) a list of reachable groups (only if the node is a gateway). To prevent too many simultaneous replies, each node should employ a random delay before replying. If no responses are found via multicast, a DTN node may attempt a broadcast in its own local-subnets to see if they can discover any DTN nodes. In addition, a DTN node can attempt to contact any previously encountered DTN participants whose information is cached.

After the discovery phase, each regular node unicasts heartbeat messages periodically with its neighbors. The heartbeat message contains information such as the node's identifier, the number of its own group members it can hear, the node's buffer availability, link duration/schedule (i.e., duration during which the node will be reachable), link characteristic (the number of hello messages received from neighbors), possibly the node's encounter histories (e.g., I have reached D before), and the number of external groups that it hears. Thus, link availability and capacity patterns can be learned and modeled via such neighbor discovery procedures.

In an infrastructure-based network, e.g., a message ferrying system, special nodes that offer services to regular nodes exist. Such special nodes will announce their presence so that regular nodes can register with them to obtain services. Consider the example shown in Fig. 2 where there is a message ferry. The message ferry periodically broadcasts a ferry announcement message. Any nodes that wish to use the ferry's service should register with the ferry. The message ferry includes the currently registered group in its ferry announcement messages so that nodes from one group can determine if they can reach nodes from another group via the message ferry. Note that a group may not be physically connected to another group (e.g., Network 1 and Network 4 in Figure 2) but the message ferry allows the two groups to communicate with one another via the store-and-forward mechanism.

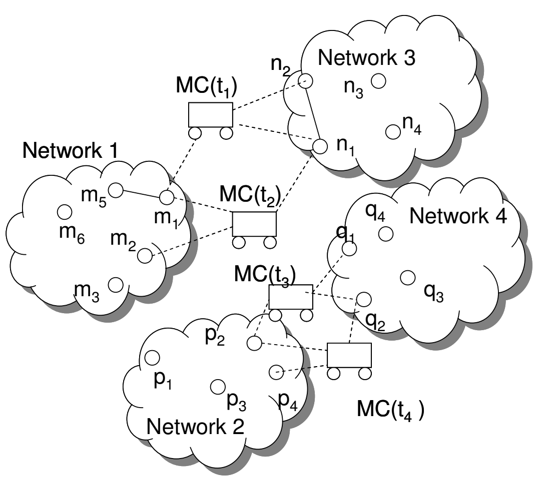


Figure 2. Neighbor discovery and ferry announcement procedures

B. Gateway Selection

In DTN scenarios, one group of nodes (say Group 1) may not be able to hear another group of nodes (say Group 3) directly but they may hear members of a third group (say Group 2) that can communicate with Group 3 as shown in Figure 3. Different groups may use different algorithms to route packets within their own groups. In our design, we assume that different groups are willing to support a few common intergroup routing messages to facilitate the ability for nodes from one group to route packets destined to another group. These include (i) a heartbeat message which contains the Group-ID, the External Groups it can reach, (ii) the Intergroup Route Request which contains the external group name and some route policies (if any), and, (iii) the Intergroup Route Reply which contains the success/failure code, and the next-hop gateway information. We refer to the support of such messages as "turning-on" the intergroup routing feature. To minimize the need for all nodes to turn on such a feature, we provide for a gateway selection protocol whereby only nodes which have been selected as gateways need to turn on the intergroup routing feature.

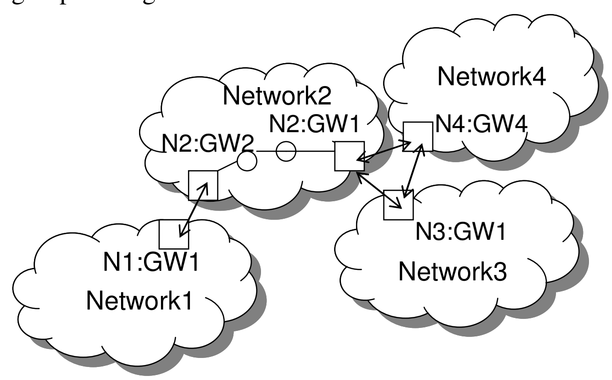


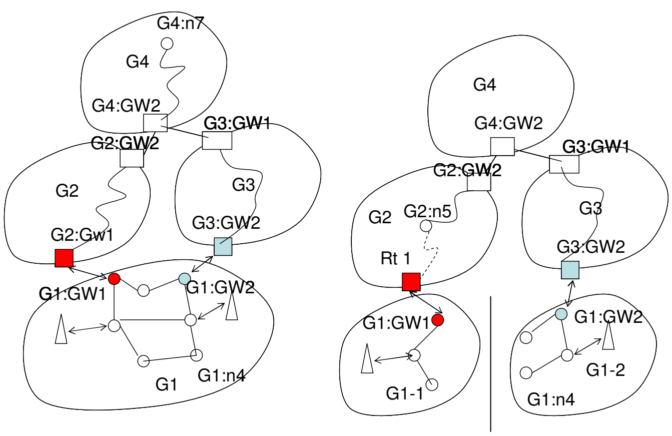
Figure 3. Intergroup routing issue

C. Mobility Management

In some scenarios, a whole network, e.g., the network hosted inside an airplane [13], may move around at a fast speed. A mobility management scheme needs to be designed to handle such mobile network scenarios as well as scenarios where a network can be partitioned into multiple networks due to geographical obstructions. For the mobile network scenario, there are three approaches that one can consider. The first approach is to assign each individual node on the plane a temporary identifier and have this node register this information with its home gateway (similar to the Mobile-IP approach designed for mobile hosts in the Internet [15]).

The second and third approaches are based on the concept of assigning a group identifier to the whole mobile network. These two approaches are more scalable than the first. In the second approach, each plane gets a special group identifier e.g. SIA.Plane101. This mobile network will "register" with a nearby gateway and that gateway's routing agent will help to inject route information so that packets destined to this plane can be delivered to that nearby gateway and hence to the mobile network. When the mobile network moves, the gateway will stop announcing such routes. Individuals currently on the plane only need to inform their home networks that they will be on SIA.Plane101. Such individuals will also register with the DTN gateway on the plane. The DTN gateway on the plane communicates with other DTN gateways so anyone interested in communicating with an individual on the plane will discover that that individual is currently on SIA.Plane101. In the third approach, each gateway on the ground can advertise some group identifiers that can be leased to a mobile network when that network is registering with the gateway, e.g., Plane300@JFK. When the plane moves to another gateway, the plane will get another temporary group identifier, e.g., Plane101@Heathrow. The downside of this approach is that the group identifier of the mobile network changes when the airplane is served by different ground stations connected to different gateways. So, in our example, we need an extra database access at the home gateway of SIA to discover that SIA.Plane101 is now Plane300@JFK.

Next, we illustrate via an example shown in Fig. 4 how our enhanced architecture deals with node mobility and network partitioning. In Fig. 4(a), we assume that gateway G4:GW2 knows that both G2:GW2 and G3:GW1 have a route to any group members in Group 1. Assume that node G4:n7 wishes to communicate with G1:n4. G4:n7 will use the routing protocol in G4 to discover that G4:GW2 knows a route to G1 and forwards its bundles to G4:GW2. G4:GW2 may decide to use multiple paths to send bundles to G1:n4 or merely use one path and use the other path only when the existing utilized path is not available. Assume G4:GW2 decides to route the bundle to G2:GW2. G2:GW2 will use group 2's routing protocol to deliver the bundle to G2:GW1. G2:GW1 then forwards the bundle to G1's gateway (G1:GW1) which then uses Group 1's routing protocol to forward the bundle to G1:n4. We assume that G2:GW1 and G2:GW2 cache the information that they have routes to G1.



(a) Before Group Partition (b) After Group Partition

Figure 4. Mobility management in enhanced DTN

Now assume that Group 1 members encounter some hurdles as they move and the group is partitioned into two groups as shown in Fig. 4(b). We assume that group 1's gateways (G1:GW1 and G1:GW2) will pick a new temporary group ID and updates G2:GW1 and G3:GW2 respectively with this information during its regular heartbeat exchange with them. Assume now that the node G2:n5 wishes to talk to G1:n4. It can send a request to G2:GW1 to see if it has a route to G1:n4. G2:GW1 will find out from G1:GW1 that G1:n4 is not reachable. G1:GW1 can discover that it no longer can talk to G1:n4 using the routing protocol of Group 1 and some timeout mechanisms. G2:n5 will have to re-issue a route request to G2:GW2. G2:GW2 will broadcast such a request to nearby gateways and eventually find the route G2:GW2-G4:GW2-G3:GW1-G3:GW2-G1:GW2. In the reply, G3:GW2 can inform the rest of the gateways of the temporary group identifier of group 1 (TGID1) so that the next time other nodes wish to communicate with group 1's members, they can check gateways that can reach TGID1. Note that in our approach, the nodes within a group that discover that they have lost their communications with certain nodes can exchange messages among themselves to decide whether or not they want to use a temporary group identifier. One way to achieve this is to have the group decide on creating a temporary group identifier when the new subgroup contains some minimum fraction of the original group size. If only one or two nodes are partitioned from the rest of the group, the orphan nodes may decide to just join a nearby group and obtain a temporary identifier.

Let us consider an example where a single node moves to a place in which the neighboring nodes are all from one particular group. This single node can broadcast an inter-domain gateway discovery message when it realizes that most of its one-hop neighbors are from a new domain. Any gateway that receives such a discovery message should respond with a unicast reply (Gateway Announcement Message). The single node can then register itself with the nearby group and be assigned a visiting identifier.

Since bundles may be destined for its old location, a mobile node may wish to ask the previously associated group to take the responsibility of forwarding messages to it via its new address. When the DTN node returns (or the forwarding request expires), then the responsibility of message forwarding is released.

D. *Route Discovery*

Next, we describe how the nodes in a DTN environment can discover routes to other nodes via a simple DTN example where a message ferry exists. In Fig.5, there are four nodes that have access to cellular links (which are wide area wireless links, denoted by the triangular nodes). We refer to them as the gateway nodes. There are seven other nodes (referred to as regular nodes) that merely have wireless LAN links. However, the nodes are sufficiently far apart from one another that they are not all connected. In addition, there is a mobile carrier that travels from point x1 to point y1 and then pauses for some time at point y1 before returning to point x1. At point x1, the mobile carrier will pause for another period of time before it repeats its route. When the mobile carrier is within the coverage area of the wireless LAN transmission, then the regular nodes can communicate with the mobile carrier. We assume that ad hoc routing protocols such as [11] are supported by regular and gateway nodes. We further assume that node 5 registers with node 9 to be its gateway during its cellular service discovery [11][12].

Consider the case in which node3 needs to communicate with node 5. It will broadcast a route request which n2 and n4 receive. Node 2 will relay this request to n1, which uses the routing algorithm in its existing domain to determine that a route exists between itself and node 9 which can reach node 5.

Eventually, a route reply will arrive at n3 indicating that the route to take is n3-n2-n1-n9-n5.

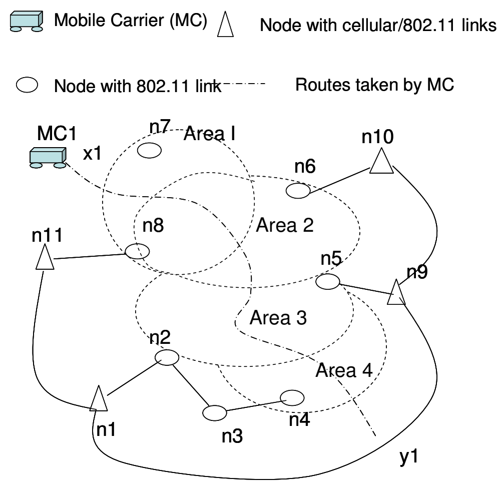


Figure 5. An example of DTN

Assume that n4 caches the route request as a "potential contact request" because n3 indicates its desire to use DTN service. Upon getting a service announcement message from the mobile carrier, MC1, n4 registers to use its service. After registration, n4 sends the mobile carrier a list of "contact requests". The mobile carrier consolidates all contact requests periodically into a list and broadcasts this list using a batched contact discovery message as it visits different areas. Node n5 hears the relayed contact request and sends a unicast reply to MC1. MC1 caches all such replies and periodically broadcasts a batch "contact response" message. Alternatively, the contact request may have an option to allow MC1 to relay a reply immediately after hearing a response.

In some cases, MC1 needs to cache those "incomplete" contact requests (i.e., those which have not discovered enough contacts or those which do not receive any response) and rebroadcasts them when it visits the next area. For example, if node 3 wishes to communicate with node 8 (or with any nodes behind node 8), a mobile carrier can only respond after node 8 registers itself with the mobile carrier.

Assume that link n1-n9 is a high latency low bandwidth link. Only when the mobile carrier is in Area 3, node 2 will be aware of the route n2-MC1-n5-n9-n10. When the mobile carrier is in Area 2, node 2 will be made aware of another additional route n2-MC1-n6-n9-n10. Note that for this route, n2 does not communicate directly to n6. The bundles sent to MC1 will be dropped off only when MC1 can hear n6. This is the main difference between a DTN route and an end-to-end route in a conventional ad-hoc network. To allow node n2 to make the decision of which route to take, MC1's response to contact request should include the estimated delivery time to n6 after visiting node 2.

V. CONCLUSIONS

New network scenarios are challenging the fundamental assumptions of Internet service models. In such scenarios, an instantaneous end-to-end path between a source and destination may not exist, and the links between nodes may be opportunistic, predictably connectable, or periodically-(dis)connected. We have proposed an enhanced disruption tolerant network architecture (called EDIFY) to address such challenges.

In this paper, we present a generalized naming convention for the enhanced DTN architecture that permits separate representations of network topology, administrative control, physical location, and other factors. In addition, we illustrate system operations in this enhanced DTN architecture such as DTN neighbor discovery, gateway selection, mobility management, and route discovery.

REFERENCES

[1] Y. Gitman, "Magic Bike: Wireless Internet", http://www.magicbike.net/

[2] L. Arent, G4techTV, Tech-Live, "Wi-Fi on Two Wheels", http://www.g4techtv.com/techtvvault/features/47657/WiFi_on_Two_Wheels.html, May 2004

[3] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations", *IEEE Computer*, 37(1):78-83, January 2004.

[4] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links", *IEEE/ACM Transactions on Networking*, 5(6):756-769, December 1997.

[5] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations", *IETF RFC* 3135, June 2001.

[6] B.D. Davison, K. Komaravolu, and B. Wu, "A split stack approach to mobility-providing performance-enhancing proxies", Technical Report LU-CSE-02-012, Computer Science & Engineering, Lehigh Univ., November 2002.

[7] K. Fall, "A delay-tolerant network architecture for challenged internets", *Proceedings of ACM SIGCOMM*, pp. 27-34, August 2003.

[8] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures", *Ad Hoc Networks*, 1(2-3):293-315, 2003.

[9] S. Weber and L. Cheng, "A survey of anycast in IPv6 networks", *IEEE Communications Magazine*, 42(1):127-132, January 2004.

[10] W. Ma and M.C. Chuah, "Comparison of interdomain routing schemes for heterogeneous networks", *Proceedings of IEEE WoWMoM*, June 2005.

[11] Y. Sun, E. Royer and C. Perkins, "Internet connectivity for ad-hoc mobile networks", *International Journal of Wireless Information Networks Special Issue on Mobile Ad Hoc Networks*, 9(2):75-88, April 2002.

[12] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu, "UCAN: A unified cellular and ad-hoc network architecture", *Proceedings of ACM Mobicom*, pp. 353-367, September 2003.

[13] Connexion By Boeing, http://www.connexionbyboeing.com.

[14] J. Klensin, Editor, "Simple mail transfer protocol", *IETF RFC* 2821, April 2001.

[15] C. Perkins, Editor, "IP mobility support for IPv4", *IETF RFC* 3220, January 2002.