A Generalized Architecture for Tetherless Computing in Disconnected Networks

Aaditeshwar Seth Patrick Darragh Srinivasan Keshav

School of Computer Science
University of Waterloo
Waterloo, Canada, N2L 3G1.

Abstract: In the emerging paradigm of tetherless computing, client applications running on small, inexpensive, and smart mobile devices maintain *opportunistic* wireless connectivity with back-end services running on centralized computers, enabling novel classes of applications. These applications require an infrastructure that is mobility-aware, disconnection-resilient and integrates support for identity management. We propose an architecture that provides this functionality essentially by adding mobility management to Delay Tolerant Networks [1]. We show that our architecture supports tetherless computing even in highly partitioned networks.[1]

Index Terms: Network Architecture, Mobility, Disconnection Tolerant Networks, Opportunistic Connectivity

I. INTRODUCTION

In the emerging paradigm of *tetherless computing*, client applications running on small, inexpensive, and smart mobile devices maintain *opportunistic* wireless connectivity with back-end services running on centralized computers, enabling novel classes of applications that address problems ranging from rural development, to environmental monitoring, healthcare and education. For instance:

- A bus carrying an 802.11 access point can *wirelessly* pick up email as it drives past rural areas that are far from Internet connectivity [1, 6, 18]. Later, the bus can forward the email to an email server on the Internet when it passes an Internet-connected wireless hot spot.
- A health care provider can record 'vitals' and test results of home-care patients in a mobile device. When driving past an access point, this information can be relayed to a central database for archiving and also sent to the attending physician for follow-up diagnosis.

In these examples, an edge application client opportunistically communicates with a centralized server over one or more wireless links. Such tetherless applications require us to address three key problems. First, to allow scaleable address aggregation, IP addresses implicitly signify location. So, when a host moves, its IP address may change. How can a sender locate a receiver whose IP address periodically changes? Second, TCP connections may restart each time a mobile disconnects and reconnects. How can a client and server maintain communication progress despite disconnections and partitioned networks? Third, a mobile client may attempt to communicate with a server from an access point that is managed by a third party. How can the third party authorize the mobile and how can the mobile authenticate the third party? Based on these considerations, we enumerate some essential goals for a tetherless computing architecture:

1. *Mobility transparency:* It should be possible to locate a mobile even as its (potentially private) IP address changes.
2. *Disconnection transparency:* Communication state should persist across disconnection periods. Moreover, simultaneous presence of both ends of a (transport-layer) end-to-end link should not be necessary because networks might be partitioned based on scheduled or unscheduled disconnections.
3. *Identity management:* A mobile user's identity should be verified during opportunistic connections. Moreover, mobile users should be protected from eavesdropping rogue access points. Although we recognize the importance of achieving this goal, we do not consider issues of identity management further in this paper, deferring that to future work.
4. *Low control overhead:* The architecture should maximize the usage of communication opportunities by minimizing control overheads. This is challenging: statistics in [6] show that 30% of the total connection time of a fast moving mobile host is taken up by connection establishment.

We have proposed an architecture in [13] that addresses these goals in the special case where mobile users are *either* disconnected *or*, when connected, can directly access the Internet (i.e they are present at one end of a transport layer connection that terminates inside the Internet). Here, we present a general solution that provides opportunistic connectivity to mobiles that may never have direct access to the Internet, relying instead on the services of a proxy to carry their data to and from the Internet. This is necessary, for instance, to support PDAs that access the Internet by means of a bus-based data mule. This apparently minor change has major repercussions on the architecture. We compare our solutions in Section II.E

II. RELATED WORK

We present a detailed survey of related work in [13]. In the interests of space, we present only a summary of related work here.

A. Cellular networks

Cellular telephone networks certainly seem to have solved the problem of tetherless communication, providing identity management and nearly seamless voice communication despite mobility and transient disconnections. However, cellular telephony (and data over cellular links) is low-

[1] The research reported in this paper extends the ideas presented in [13]. A detailed comparison can be found in Section II. E.

bandwidth—on the order of 100kbps, even with 3G—and expensive. Moreover, widespread penetration of cellular networks is limited to urban areas; most rural areas, especially in developing countries, cannot afford ubiquitous cellular coverage. Furthermore, 802.11-based wireless networks experience more handoffs and roaming than cellular networks, due to the fact that they have a much smaller coverage area. For these reasons, existing cellular network solutions cannot be trivially retargeted for tetherless computing

The real challenge here is to mimic the considerable capabilities of cellular networks by composing a large number of heterogeneously administered wireless LANs and potentially using a motorized backhaul. This would result in a network that would not support interactive communication, but would be far cheaper and would have up to 500 times the capacity of cellular networks.

B. Network layer mobility solutions

Although schemes such as mobile IP [7], HIP [19] and I3 [2] provide mobility transparency (and with HIP, identity management), they cannot function effectively in partitioned networks. In particular, they do not address the problem of updating a location register when a mobile is able to access the Internet only through a proxy.

C. Transport layer disconnection tolerance solutions

This includes protocols like TCP Migrate [5] and Rocks-and-Racks [4], which essentially provide OSI session layer functionality to resume a TCP connection on network reconnection. However, these protocols only support TCP, and also only on an end-to-end basis. Therefore they cannot function in a highly partitioned network.

D. Delay Tolerant Networks (DTN)

Unmodified TCP cannot be used over challenged networks where there are frequent disconnections, or when the network regions to be traversed are heterogeneous and connectivity is not always available. DTN [1] instead proposes the notion of bundle transfer, where the data is wrapped into bundles (similar to email messages) and these bundles are transferred on an overlay network. DTN routers have a large persistent bundle store, where bundles await transfer to the next DTN router whenever connections become available. Custody transfer occurs as the bundles are transmitted across the network, and the responsibility of reliable delivery of each bundle is passed from one custodian DTN router to the next.

DTN addresses the problem of disconnections in a novel manner, even though it constrains applications to be non-interactive in nature. End-to-end connectivity is not necessary with DTN. Senders and receivers can inject and retrieve bundles from the DTN overlay according to their individual connectivity schedules. A side benefit of DTN is that high throughput rates can be obtained for opportunistic delivery if bundles are routed to DTN routers physically close to potential receivers [13]. However, the existing DTN architecture has no support for mobility, which is an essential requirement for tetherless computing.

E. DTN/I3 Architecture for Tetherless Computing

In [13], we proposed an architecture that handled transport layer disconnections using DTN, and mobility using I3 [3]. However, this architecture does not handle the case where a mobile never has direct access to the Internet, always relying on a proxy to ferry its data to and from the Internet. In such a situation:

1. I3 triggers in the Internet cannot be easily updated, because there is never a transport level connection between the mobile and the I3 server. Trigger updates *could* be tunneled through to I3 servers, but this complicates the architecture.
2. All data, even that bound for the local region, must travel first to the Internet region's I3 servers, which is inefficient
3. The mobile does not have access to a I3-DNS server to translate from a destination name to its trigger.

In this paper, we present an alternative architecture that avoids these problems.

III. Tetherless Computing Architecture (TCA)

A. Overview and definitions

Based on the goals outlined in Section I, we believe that the following features must be included in any architecture for tetherless computing.

1. *Intermediate persistent storage:* End to end connections will not be always possible for disconnected mobiles. Hence, intermediate infrastructure nodes should have a persistent storage capability where senders can inject data and receivers can pick up data according to their individual connectivity schedules.
2. *Lookup on a globally unique identifier:* Each mobile host should possess a GUID that maps to the current location of the mobile. The GUID should not change as the mobile moves, and its mapping should be updated periodically with the new location.
3. *Forwarding to changing mobile locations:* The data forwarding function must incorporate a lookup step.

In our architecture, we use a distributed hash table (DHT) infrastructure such as OpenHash [11] for looking up the location of a mobile based on its GUID. A DTN overlay network is used for intermediate persistent storage, and data forwarding within the DTN network is handled by DTN routing protocols.

We first present some definitions.

1. *Region:* We follow the DTN definition of a region i.e a collection of mutually reachable DTN routers, determined by administrative policies, communication protocols, naming conventions, or connection types [17]. Regions may also represent physical boundaries.
2. *Gateways:* These are DTN routers with interfaces on more than one region.
3. *Custodians:* These DTN routers act as always-available proxies for intermittently connected mobile hosts. Custodians store data on behalf of disconnected mobile hosts and deliver them whenever the hosts reconnect to the network. Note that

because custodian DTN routers must always be directly available on the Internet we impose the simplifying constraint that custodians must not be mobile.

4. *Near area:* This is defined as the set of wireless access points (APs) that are 'closer' to a particular custodian DTN router than any other DTN router. We do not define closeness precisely; one candidate would be the long-term mean RTT delay between an AP and the custodian DTN router.

5. Mobility within a near area is *near mobility.* Pre-authentication can used during near mobility to help reduce reconnection delays [8, 9].

6. Mobility between near areas is *far mobility.* We assume that near mobility is far more common than far mobility, and we optimize our architecture for this case.

7. *Local DTN router:* This is the DTN router that communicates directly with a mobile. A local DTN router may or may not also be a custodian. A local DTN router that is not a custodian may itself be mobile. For instance this models a bus that travels between villages with a DTN router on board.

B. *Location management*

We identify all mobile hosts using an opaque globally unique identifier (GUID). The Internet region, which has special status in our architecture, maintains a DHT that maps from a mobile's GUID (I) to its current region (R). Following cellular telephony terminology [15], we call this lookup table the *Home Location Register (HLR)*. Unlike cellular networks (and unlike Mobile IP), our HLR uses a DHT to gain scalability and fault-resilience. If a mobile is simultaneously present in multiple regions—for instance, when the mobile host is reachable on two different bus routes—the HLR will resolve the GUID to multiple regions.

Each region maintains at least one Visitor Location Register (VLR) that is either stored at, or accessible to, *all* of its gateways. The VLRs store a mapping from the GUIDs (I) of all mobile hosts currently in the region to the custodian DTN router (C) of the mobile. Multiple mappings for different custodians can also be stored if the mobile can pick up bundles from more than one custodian.

Finally, each custodian maintains a Local Location Register (LLR) that maps from the GUID to the best last-hop fixed or mobile local DTN router (M) for each mobile. If the mobile picks up data directly from the custodian, without an intermediate local DTN router, this is reflected in the LLR.

This three-stage lookup hierarchy is shown in Fig. 1. When a mobile device moves, its location information is updated, if necessary, in zero or more location registers. If a mobile moves such that its local DTN router doesn't change, then none of the location registers are updated after the move. If a mobile moves such that its custodian doesn't change, but its local DTN router changes (i.e. a near move), then only the LLR at the custodian is updated to reflect the new best local DTN router for the mobile. If a mobile changes custodians within a region, the LLR at the new custodian has to be set up, and *all* VLRs in the region have to be updated to point to the new custodian. Finally, on a far move, the HLR, VLR, and LLR must all be updated both in the new and old regions. It is important the HLR only be modified *after* all the VLRs in the new region are aware of the mobile, as described in Section III.E.

There are many ways of maintaining the location registers. If the region is large (like the Internet region) then the location register can be maintained in a DHT like OpenHash, or in a central database. If the region is small then it can be maintained in a lookup table at each gateway and custodian.

C. *Discovering local and custodian DTN routers*

When a mobile moves, its location must be updated in the location registers. Therefore, the mobile device needs to determine its new region, custodian, and local DTN router. A local DTN router can always be queried for a list of 'nearby' custodians and regions accessible through it. So, the problem reduces to that of a mobile finding a local DTN router after it has associated with a new wireless access point.

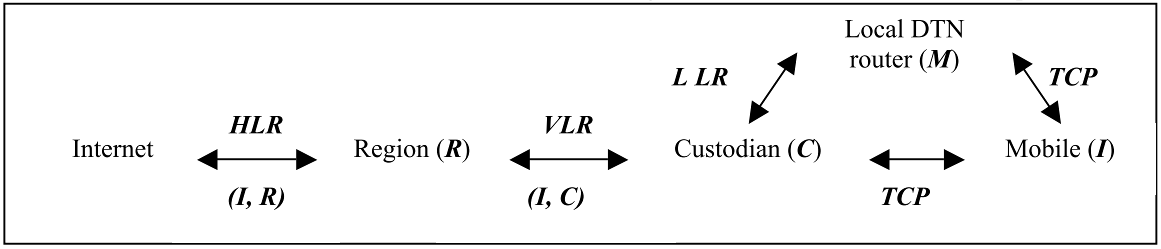
Note that when a mobile associates with a wireless access point, one of two cases must hold.

1. If the wireless access point is not itself connected to the Internet, then, in order for data to be ferried to the Internet, it must be co-resident with a mobile DTN router (as on a bus). This must therefore also be the local DTN router.
2. If the wireless access point is connected to the Internet, then the mobile can access a centralized location service with location information, such as the SSID of the access point, the current zip code or its GPS location, to find a 'close' local DTN router. Alternatively, when the access point is initially set up, this information can be hand-configured in the same way that it is configured with the address of a DNS server.

A local DTN router, when queried, may return more than one choice of custodian. Indeed, since the local DTN router may be one overlay hop away from all custodians on the Internet, the choice of custodian may be non trivial. We believe that a scheme similar to I3 *trigger sampling* [3] can be used to choose a 'close' custodian. We are looking into algorithms for optimal choice of custodian in ongoing research. In any case, our scheme will work correctly with *any* custodian – choosing a 'close' custodian is simply a performance optimization.

D. *Late binding of regions*

The current DTN architecture [17] supports the notion of late binding, where the administrative ID portion of the DTN



address is bound to an actual next hop only at the destination region. We extend this notion to allow even a node's region to be late bound. More precisely, we assume that every DTN router has a *default route* that is its next hop to get to the Internet region. We also assume the existence of a special region name called 'unknown'. When a DTN router gets a bundle addressed to the 'unknown' region, it either forwards the bundle on its default route to the Internet, or, if it has an interface on the Internet, looks up the destination's GUID in the HLR to rewrite the 'unknown' region with the mobile's current region. Subsequently, the bundle is forwarded using normal DTN routing. This optimization allows a disconnected node to send a bundle to a destination knowing only its GUID—it does not have to query the HLR for the mobile's current region.

E. Avoiding race conditions during location updates

Mobility intrinsically introduces race conditions. Bundles may be sent to a mobile's old region, or they may arrive to a gateway or custodian in the new region before it has heard of the mobile. To avoid race conditions, the location registers must be updated with care. The basic principle is to always have new location information reliably set up, using a group communication protocol [20], before old information is deleted ('make then break'). We now outline an algorithm for location update management in case of an inter-region far move (the case of a near move is a straightforward subset):

1. The mobile associates with a wireless AP and is given an IP address, for instance using DHCP. It discovers its local DTN router using one of the techniques in III.C.
2. The mobile tells its local DTN router that it has moved to its new region from its old region.
3. The local DTN informs the mobile of its choice of 'nearby' custodians.
4. The mobile chooses one or more custodians and informs the local DTN router.
5. The local DTN router participates in a group communication protocol to update all the custodians' LLRs to make itself as their next hop to get to the mobile.
6. When Step 5 terminates, one of the chosen custodians participates in a group communication protocol with all the gateways in the region to update their VLRs so that the GUID of the mobile maps to its set of chosen custodians.
7. When Step 6 terminates, one of the gateways updates the HLR to point the mobile's GUID to its region.
8. Next this gateway participates in a group communication protocol with the set of gateways in the mobile's old region to update their VLRs so that these VLRs 'unmap' the GUID by mapping the GUID to 'undefined'.
9. When Step 8 terminates, the gateway reliably multicasts an update to all the custodians in the old region. The custodians send any stored bundles to the mobile in the new region by rewriting the bundles' destination region as 'unknown' and forwarding them on its default route. Since this is the last step, group communication is not necessary.
10. Any bundles addressed to the old region whose GUID is unmapped are rewritten with an 'unknown' destination region and sent back into the Internet region.

This 'make-then-break' approach has the interlocks necessary to prevent race conditions and provides 'eventually always consistent' semantics. For instance, if a VLR points to an old custodian, bundles reaching the old custodian will either be stored and eventually forwarded (when the location update reaches the custodian), or the custodian will find the GUID to be unmapped, in which case the bundles will automatically be forwarded to the new custodian. A detailed algorithm can be found in [14].

F. Multiple home regions

Our architecture treats the Internet region as a special region that maintains the HLR. In fact, multiple regions can be designated as home-regions by embedding the Id of the home-region within the GUID of the mobile hosts. Thus, each mobile host can belong to its own home-region that maintains an HLR for it. All lookup or data delivery requests for a mobile host automatically get routed to its home-region. This allows administrative regions to manage their own HLR and GUID assignments.

G. Example

Fig. 2 illustrates the scheme in operation. The shaded region represents the Internet region. Region *R1* is directly connected to the Internet, while Region *R2* is connected through Region *R1* to the Internet. (Note that *R2* may have a scheduled link to *R1* and so may never be able to directly access the Internet region.) The network contains custodians *DTN-1* through *DTN-3*, and *M-DTN* represents a mobile DTN router, such as a bus.

Data transfer is illustrated from a fixed sender located in the Internet, through a custodian in the Internet, to a mobile receiver that changes its location from *L-1* to *L-5*. In Fig. 2a, the mobile host is present in Region *R2*, where it does a near move within the vicinity of *DTN-1*. This is followed by intra-region move to the *DTN-2* area in Fig. 2b. An inter-region far move occurs in Fig. 2c to the *DTN-3* area in the Internet region, where the mobile host moves to a new location *L-4*. Finally, the mobile host in location L-5 gains connectivity through a mobile local DTN node, also in the Internet region.

IV. DISCUSSION

A. Globally unique Ids and routing

Address aggregation and mobility are mutually antagonistic concepts. Address aggregation is possible only when nodes with similar addresses are topologically close, so that an address range can be assigned a common next hop; mobility means that this is precisely not the case—even if nodes with similar addresses were close by to begin with, over time they would move apart. Therefore, any scheme for mobility must support location-independent GUIDs that are mapped either to location-specific addresses, or are directly incorporated into routing tables. By the former, we mean that a node has a location-specific (aggregable) address that changes as it moves, and a lookup table maps from the GUID to the current location. By the latter, we mean that each router, for each

GUID, keeps track of the next hop. The latter solution is not scaleable: it would mean that the routing table size at *every* router would be size of at least the number of mobiles in the system. Consequently, every mobility solution must use some form of translation from a GUID to a location-specific address, with aggregate (address-range) based routing and forwarding tables.

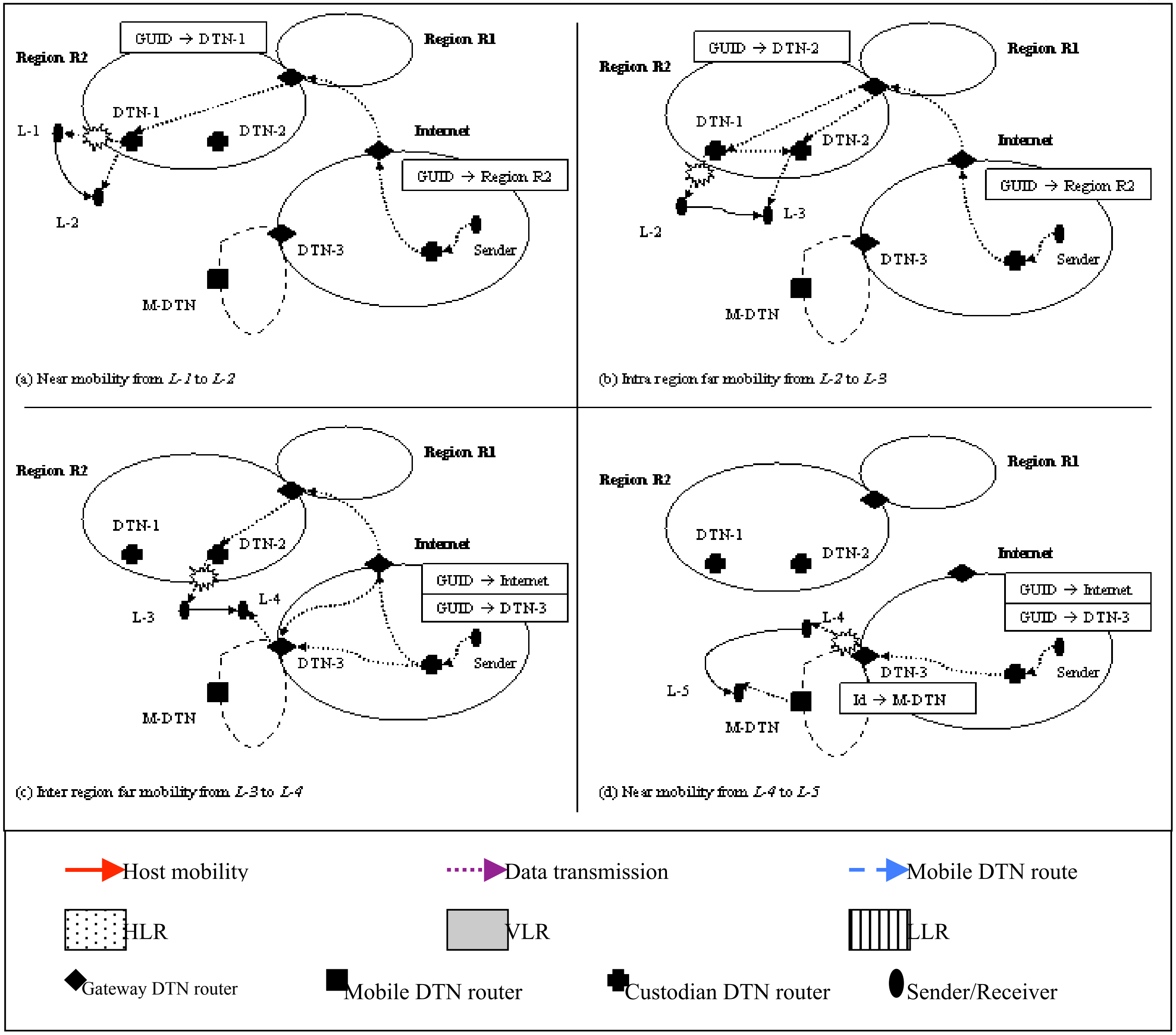
The current DTN architecture uses globally unique node identifiers of the form (region, admin id), where the admin id is opaque outside the region. This allows one level of aggregation, but suffers from the problem that the identifier changes as a mobile changes regions. Consequently, we need to have different a GUID to identify mobile nodes, and we have introduced this into the DTN architecture.

We could have mapped from a GUID directly to the location-specific address of a mobile in the Internet DHT. However, this would have required a deluge of DHT updates even for moves within a region. Our approach--to use the DHT to map from a GUID to a region--allows us to avoid DHT updates for the common case of near mobility. On the other hand, it forces us to maintain at least one other lookup table to actually determine a node's location-specific identifier. In fact, we further partition lookup in two: the VLR maps to the custodian's address, and at a custodian the LLR maps to the best next hop. This is because the only way to reach the mobile is through a custodian. So, it makes sense for the VLR to point to the custodian rather than directly to the mobile. Note that unlike the HLR, the VLR only needs to track mobiles actually in the region, so it can be much smaller.

Vanilla DTN routing ought to be enough for a custodian to choose the best next hop to a mobile's location-specific address. Since this routing has yet to be defined, at the time of this writing, we use a simple GUID-based lookup table to map from a custodian to the next-hop local DTN router. Because of default routes, this table can be small.

Given that we need GUIDs anyway, the question is where to put them. One option is to extend a mobile node's DTN address with a GUID. Our solution, a minor optimization, is to have all regions use the GUID as the admin ID.

Note also that non-mobile nodes can be assigned aggregable (location-specific) addresses. We are looking into assigning



(a) Near mobility from *L-1* to *L-2*

(b) Intra region far mobility from *L-2* to *L-3*

(c) Inter region far mobility from *L-3* to *L-4*

(d) Near mobility from *L-4* to *L-5*

fixed nodes an aggregable address, and allocating GUIDs only for mobile nodes.

B. Cellular telephony architectures

TCA is modeled on cellular telephony architecture [15]. In cellular networks, multiple Base Station Systems (BSS) are grouped under a single MSC (Mobile Switching Center). A mobile's GUID is mapped in a VLR (Visitor Location Register) to an MSC. The VLR and MSC's lookup tables locate a mobile's BSS. A global HLR (Home Location Register) tracks the current location of the mobile host. TCA works similarly, with the HLR pointing to the current region, and each gateway hosting a VLR for that region.

Differences in the two architectures arise because of the partitioned network structure in TCA. Cellular telephony assumes always-available connectivity of the mobile host with the central management system. For example, although SMS (Short Message Service) can be viewed as a form of delay tolerant data transfer in cellular telephony, SMS data is always stored on a central SMS Center (SMSC). In contrast, the partitioned network in TCA requires a distributed network of custodians that are used to relay the data from one mobile host to another. Other macroscopic differences between the architectures are reviewed in Section II.A.

V. CONCLUSIONS AND FUTURE WORK

We believe that the emerging paradigm of tetherless computing requires an infrastructure that deals well with mobility, disconnection, and identity management. The architecture proposed in this paper (TCA) achieves the goals enumerated in Section I. Unlike the solution in [13], our architecture seamlessly supports mobility and disconnection even in networks where end points may never have a direct connection to the Internet, relying instead on a proxy to ferry data to and from it. We showed how to avoid race conditions in such networks and presented a novel technique of late-binding region names to avoid expensive lookups from disconnected nodes. We also discussed the role of GUIDs in mobility, and their impact on DTN architecture. We now consider some avenues for future work.

Should all addresses be late bound? In some cases it might be more efficient to first do a lookup and then dispatch the data addressed directly to the current region of the receiver. Furthermore, in the case of pre-scheduled transmissions, the receiver itself can update the sender with its current location before the data transmission commences. The optimal decision on when to bind depends on factors such as the application scenarios being considered, degree of movement of the receivers, class of service, volume of data, lookup latency, cost, and resource constraints. This leads to a number of open questions: Which nodes should be looked up instead of delayed-bound? What should be the degree of caching or aggressiveness in the lookups? We have attempted to answer these questions to some extent in [14] by using service discovery protocols like Jini and SDP.

Detecting movement is not always simple. Consider a mobile host reachable on two bus routes that lead to different regions. How does the mobile host determine whether it is present in the same physical location but on two different bus routes, or has moved physically to a new location on another bus route? The answer might be obtained using GPS or by the buses carrying information about the immediate postal code of the area that they are driving through, or possibly by other means. We are looking into some solutions to this problem.

As mentioned in Section IV, construction of a hierarchical address based routing scheme for fixed DTN nodes is an open area. We are exploring this further by looking at IPv6 to allocate addresses on hierarchical topologies for a fixed DTN network. Augmenting routing with optimization on delivery-time based metrics is another enhancement [16].

Finally, identity management is a substantial open issue in tetherless computing. We are currently addressing this issue using well-known techniques in identity-based cryptography.

VI. REFERENCES

[1] K. Fall. "A Delay-Tolerant Network for Challenged Internets," *Proc. SIGCOMM 2003.*

[2] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. "Internet Indirection Infrastructure," *Proc. SIGCOMM 2002.*

[3] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker. "Host Mobility Using an Internet Indirection Infrastructure," *Proc. MOBISYS 2003.*

[4] V.Zandy and B. Miller. "Reliable Network Connections," *Proc. MOBICOM 2002.*

[5] A. Snoeren. "A Session-Based Approach to Internet Mobility," *PhD Thesis, MIT, http://www.cse.ucsd.edu/~snoeren*, Dec 2002.

[6] J. Ott and D. Kuyscher. "Drive-Thru Internet: IEEE 802.11b for Automobile Users," *Proc. INFOCOM 2004.*

[7] C. Perkins. "IP Mobility Suport for Ipv4," *http://www.ietf.org/rfc/rfc3344.txt*, Aug 2002.

[8] B. Aboba. "IEEE 802.1x Pre-Authentication," *http://www.drizzle.com/~aboba/IEEE*, June 2002.

[9] A. Mishra, M. Shin, and W. Arbaugh. "Pro-active Key Distribution using Neighbor Graphs," *IEEE Wireless Communications*, Feb 2004.

[10] M. Luby. "LT Codes," *Proc. FOCS 2002.*

[11] B. Karp, S. Ratnasamy, S. Rhea, and S. Shenker. "Spurring Adoption of DHTs with OpenHash, a Public DHT Service," *Proc. IPTPS 2004.*

[12] K. Fall. "Erasure Coding for Error Control in DTN," *Personal Communication.*

[13] A. Seth, Y. Lin, S. Liang, P. Darragh, and S. Keshav. "An Architecture for Tetherless Computing," Submitted to *INFOCOM,* 2005. Available from *http://mindstream.watsmore.net*

[14] P. Darragh. "The TCA Protocol," *http://mindstream.watsmore.net*

[15] A. Samjani. "General Packet Radio Service (GPRS)," *IEEE Potentials*, 2002.

[16] S. Jain, K. Fall, and R. Patra. "Routing in a Delay Tolerant Network," *Proc SIGCOMM 2004.*

[17] V.Cerf, S.Burleigh, A.Hooke, L.Torgerson, R.Durst, K.Scott, K.Fall, H.Weiss. "Delay Tolerant Network Architecture," *Internet Draft http://www.dtnrg.org/specs/draft-irtf-dtnrg-arch-02.txt*, July 2004.

[18] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking Connectivity in Developing Nations," *IEEE Computer,* 37(1):78-83, 2004.

[19] R. Moskowitz, P. Nikander. P.Jokela,T. Henderson, "Host Identity Protocol," *http://www.potaroo.net/ietf/ids/draft-ietf-hip-base-00.txt*, 2004.

[20] K. Birman, "Building Secure and Reliable Network Applications," Manning Publications and Prentice Hall, 1996.