Numi: Collaborative Mobile Data Management in Infostation Networks

Olga Ratsimor, Sethuram Balaji Kodeswaran, Anupam Joshi, Tim Finin, Yelena Yesha
Department of Computer Science and Electrical Engineering,
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250

Abstract

A network of infostations offering high bandwidth islands of data connectivity has often been suggested as a viable alternative to cellular WAN for providing network connectivity for mobile devices. Current data management models proposed for such networks treat these infostations mainly as passive entities that offer devices in range, access to needed information. We propose a novel approach wherein infostations actively participate in delivering content even to devices that are not in range. Other participating mobile devices that have excess spare capacity are used as carrier nodes to transport data towards remote devices. Short range wireless technologies enabling creation of ad-hoc peer-to-peer networking capabilities between devices are used to form transient communities to exchange information. The infostations are responsible for selecting appropriate carrier devices to use so that these carriers meet the device in need at the appropriate time to participate in this collaboration. Through this, dependency on the more expensive WAN networks is minimized. We present a design of our proposed framework and its components along with a prototype implementation.

1 Introduction

Recent years have seen a tremendous proliferation of mobile computing devices. Devices such as Personal Digital Assistants (PDAs) have undergone constant improvements and are today full fledged computers with improved processing power, multimedia capabilities etc. Fueled by such advances, new and improved services and applications are beginning to be offered on them. Wireless networking has also witnessed remarkable growth in recent years. Also, there has been a lot of advances in the field of short-range wireless communication technologies like Bluetooth, homeRF etc. These technologies have enabled the creation of ad-hoc peer-to-peer networking capabilities on mobile devices. Now it is possible for these devices to discover peers, form transient peer communities to exchange information and gracefully handle changes in their neighborhood. We are seeing numerous applications available on desktop PCs making their way into mobile devices. A great majority of currently available mobile devices have restrictions that hinder this migration. These include limitations on power consumption, smaller display, processing power, limited network connectivity, etc.

We present our approach to resource sharing among peer users to maximize their efficiency as a whole in a network of infostations. We believe that by coordinating such ad-hoc collaboration among mobile users, individual user data needs can be satisfied in an efficient manner. We present an intelligent data caching strategy that utilizes user mobility patterns to efficiently manage data demands in the network. Using infostations to facilitate and coordinate caching of selective future data needs on a device, more efficient usage of the available communication infrastructure can be attained. We present a novel communication mechanism that uses collaboration among mobile devices to transfer data to the user in need. Using mobility patterns, data intended for a specific user can be piggy-backed on devices heading towards that user. Through this mechanism, infostations can now actively service remote users, thereby reducing the dependency of those users on the expensive WAN. We have built a working prototype of our system which is presented in this paper.

2 Numi Framework Components

2.1 Service Portals, Mobile Hosts and Services

The key components of our Numi framework include Service Portals (SPs), Mobile Hosts (MHs), and Services. SPs are infostations running the Numi platform and hosting services that can be used by MHs. The SPs are connected by high speed links to the rest of the wireline network. The SPs use their wireless capabilities to interact with MHs that are in range and use their wireline connectivity to communicate among themselves. Mobile Hosts (MHs) are wireless mobile devices running the Numi Platform. MHs can communicate both with SPs and neighboring MHs that are within range. These MHs travel through the geographical area populated with SPs. Our network can be thought of as comprised of two distinct types of zones: *landing zones* and *transit zones*. A *landing zone* is essentially an island of connectivity around a service portal limited only by that portals wireless range. An MH can communicate with an SP when it is in a *landing zone*. In a *transit zone*, an MH can communicate only with peer MHs that are within its immediate neighborhood. Furthermore, it is assumed that MHs move along predetermined routes (like highways). An MH can request services from SPs that it encounters as well as other MHs. We assume a heterogeneous mix of mobile devices in our network with differing capabilities. Devices also vary in their level of participation in our system and the amount of resources that they are willing to share. Each Service in our system consists of a Service Specification, a Service Agent and Service Data Volume (SDV). The Service Specification is a meta description of the service. A Service Agent is the code that is capable of executing on the MH so as to realize the service to the user. Also, each service will have a portal service agent running on each SP in the network that advertises the presence of this service in the network and assists corresponding service agents running on MHs. Service Data comprises the actual data that is used by this service. A Service Data Unit (SDU) is the smallest indivisible unit of data for a service. A collection of such SDU forms a SDV. Data updates to MHs takes place in terms of these aggregate units to maximize communication efficiency.

2.2 Numi Platform Agents

We designed our framework with the goal of reusability. We have developed a set of agents that could be used on both MH and SP. The framework agents include the heartbeat generator, location monitor, message handler, logger, task scheduler, data handler and a service manager. A heartbeat generator agent is responsible for broadcasting device presence messages that uniquely identifies the platform. These heartbeats are broadcast periodically (every t seconds). A location monitor agent is responsible for location dependent issues. On an MH, this agent identifies whether or not that device is currently in a *landing zone* or a *transit zone* (based on whether or not a portal presence message has been received in the last t seconds). Also, this agent tracks the other peer devices that are currently in the neighborhood. A message handler agent is responsible for handling the messaging needs of the framework. Messages are routed using a combination of agent identifier, platform identifier and a message type. A logger agent records every interaction that takes place on the local device. A task scheduler agent is responsible for scheduling prescribed tasks at various times. A data handler agent is used for transferring data volumes between MHs and between an MH and an SP. Service agents run on top of our Numi platform and offer services to a user. Service agents at an SP actively wait for a user's request for a service. When a user requests a new service, the SP service manager notifies the corresponding service agent. This service agent then transmits the initial SDV to the MH and schedules subsequent volume updates throughout the network. Our current implementation can schedule volume updates only up to the next portal on the route. We are working on extending this to include all SPs on any given devices route. In addition, whenever a volume is scheduled at an SP, the service agent at that portal is responsible for ensuring that this SDV gets delivered to the MH. In case a device does not show up at a *landing zone* at it's prescribed time, the service agent at that SP then actively starts looking for other devices whose routes indicate that they are heading in the direction of that MH and attempts to use them to deliver the next data volume. The service agents continue to track the MH till it arrives and obtains a new SDV. Once this is done, the SP service agent notifies the next SP(s) on the route to schedule future volumes for this MH. A service agent on an MH offers a service to the user. Whenever a device enters a *landing zone*, the service agents on that MH receive SDV updates from their respective SP service agents. An MH service agent also monitors service data usage and detects when the service is running out of data. When this occurs, the service agent publishes queries in its neighborhood to obtain the next set of data needed to keep that service running. Service agents on other devices that have this data acknowledge these queries and using the data handler agents, data can be exchanged. In some cases, these queries cannot be handled by neighbors. However, since these interactions are being logged by the logger, a neighboring device reaching an SP can trigger this SP to attempt to deliver the data to the MH that initiated the query. A service manager agent is responsible for managing service agents on a platform. The SP service manager hosts the service agents representing services available at that SP. This manager uses each service agents' service specification to generate a list of available services with their description. The service manager at the MH, can activate, suspend or terminate other service agents. Also, this manager monitors system usage by each service agent including statistics like the amount of memory used, running time, messaging overhead incurred etc.

3 Implementation and Experimental Results

We have implemented a prototype of our framework using Java programming language. We installed our platform on three PCs and three iPAQs. The PCs run the SP platform and the iPAQs run the MH platform. All devices used were equipped with 802.11b wireless LAN cards. The iPAQs were running the Jeode Embedded Virtual Machine. Each SP also runs a Tomcat Apache servlet engine. To simulate the mobility of the devices (moving in range and out of range of each other) we divided each *transit zone* into non-overlapping cells. Each cell has a unique cell ID. MHs are able to communicate with each other only if they are in the same cell. Messages have been augmented to carry a cell ID. Since we are using 802.11, broadcast messages will be heard by all devices. However, the message handler filters out all messages that do not match a device's current cell ID. By using this notion of cells, we can simulate neighborhoods and by changing a MH's cell ID, its neighborhood can be changed thereby simulating movement. We have developed an additional simulation component called the Mobility Coordinator. Using this, control messages can be sent to any MH to change its current cell ID.

3.1 Experimental Setup

We set up network of 3 SP: P1, P2 and P3. *Transit zones* between SPs were divided into cells. There are three mobile users Susan, Bob and Jim with iPAQs. Susan has MH1; she starts her trip at P1 and plans to go to P3 through P2.

Bob has MH2; he starts his trip at P2 and plans to go to P1. Jim has MH3; he starts his trip at P3 and plans to go to P2. There are three services available on the network: Music Service, Newspaper Service and eBooks Service. Susan is using Music Service and eBooks Service; Bob is using Newspaper Service; Jim is using eBooks Service.

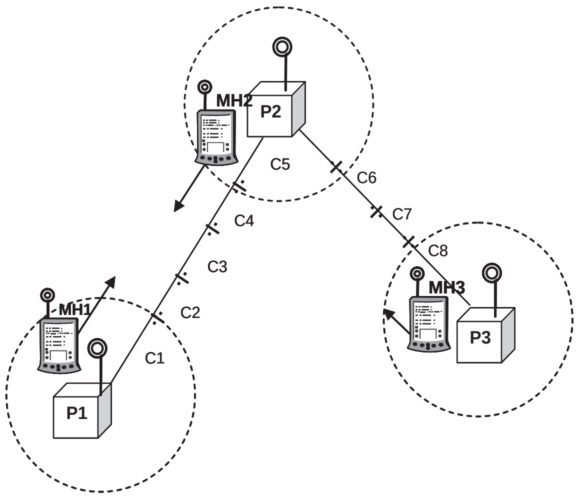

Figure 1. Experimental Setup

"Memory Shortage" scenario occurs when the user requests one or more services but his or her MH is unable to store all the needed data to last all the way form one *landing zone* to the next. Suppose Susan requested the Music Service and the Newspaper Service through a MH to Portal interaction. MH1 gets two SDV and Susan starts her trip form P1 to P2. MH1 has enough data to last Susan 80% of her trip. She will run out of data in C3. P1 notifies P2 about service provided to Susan through portal to portal interaction. P2 determines that she does not have enough data to last her till she reaches P2. Meanwhile, Bob with MH2 is getting ready to depart from the *landing zone* of P2 (towards P1). He is using the Newspaper Service. P2 gives MH2 the volumes for Bob's Newspaper Service and it also gives the next set of SDVs needed for Susan's Newspaper and Music Services. Bob passes by Susan in C3. MH1 contacts MH2 and obtains the next set of SDVs for the Newspaper Service and Music Service. Susan now can reach P2 with out any service disruptions.

"Delay" scenario deals with cases when a user takes longer then expected to pass thought the *transit zone*. The user is initially given enough data to last from one *landing zone* to the next *landing zone*. However for some unexpected reason the user takes longer than initially was expected. Suppose Susan continues on her way from P2 to P3 though C5, C6, C7 and C8. She is continuing to use Music Service and Newspaper Service. MH1 has enough memory to store SDVs for her services to last her from P2 to P3. She departs P2's *landing zone* towards P3 with no plans for detours or delays. Along the way in C7 she sees a coffee shop. As she stands in line to place her order she continues to read her newspaper and listen to her music. Meanwhile P3 determines that Susan did not arrive on time to P3's *landing zone*. P3 receives a request for a Newspaper Service from Jim who is planning to depart P3's *landing zone* and head towards P2. His MH3 gets SDVs for Jim's Newspaper Service and also receives music and newspaper SDVs for Susan. By this time, service agents on Susan's iPAQ would have realized that they are running out of data and will start querying the neighborhood. As Jim passes by the coffee shop, MH1 and MH3 discover each other and MH1 uploads needed SDVs from MH3. Susan gets her coffee and resumes her trip. She reaches P3 with out service interruptions.

4 Conclusion and Future Work

We have presented a novel approach to manage data needed by a mobile device in an infostation network. Our model augments hoarding schemes with the ability of users to share load among themselves so that collectively they can satisfy individual user data needs. Through this, we attempt to minimize the amount of interaction a mobile device needs with the WAN network. Also, by sharing load, sophisticated applications can be offered even on less capable devices as long as their neighborhood is sufficiently resource rich to satisfy the applications needs. The infostations in our model facilitate this collaboration by equipping devices that are likely to meet with data that the other may require. This allows our infostations to offer services even to devices that are not in range. Unlike existing schemes that do not gracefully handle deviations in a devices expected route, we provide a mechanism for our infostations to detect such deviations and react by actively trying to route needed data to these devices. Through our framework, data needs for mobile devices can be managed across a network of portals in a highly distributed manner that scales and is cost efficient with little dependency on a cellular WAN.

Currently we are extending our framework to support *n*-hop scheduling of data on SPs on an MH's route. We are working on implementing intelligent services that can monitor usage patterns so as to adapt the data delivery mechanism to offer optimal efficiency. Ability for users to modify the level of collaboration for their MH is being developed. We are adding support in our framework to allow an MH to issue a query in a *transit zone*, whose result will be made available to that MH when it arrives at its next *landing zone* or will be carried to it by other MHs. We will also be migrating our framework to use Bluetooth protocol for ad-hoc interactions. We are also building a simulation model of our approach to study characteristics of our framework such as different scheduling mechanisms, scalability, performance when MH routes are not known and performance when there is a combination of MHs whose routes are known and others whose routes are not. Also, using mobility patterns, number of MHs passing through an SP and their levels of collaboration, we would like portals to generate an analytical estimate of the bandwidth that is available so as to offer QoS guarantees to its users.