A Cost Driven Approach to Information Collection for Mobile Environments

Qi Han and Nalini Venkatasubramanian
Department of Information and Computer Science, University of California-Irvine, Irvine, CA 92697
Email: {qhan,nalini}@uci.edu

Abstract— **Mobile multimedia applications have QoS requirements; resource provisioning algorithms utilize current system resource availability information to ensure that these applications meet their QoS requirements. Information collection algorithms collect and maintain current system resource information and are vital in performing efficient resource provisioning. In this paper, we present a cost-driven approach to the information collection problem for mobile environments. Our approach aggregates mobility patterns of mobile hosts and tries to minimize the overhead involved in the collection process. We compare its performance with existing techniques for collecting system state information.**

Keywords— **information collection; resource provisioning; aggregation mobility model.**

I. INTRODUCTION

In the future, we are likely to see a tremendous rise in mobile computing and communications as ubiquitous applications incorporate multimedia information. Achieving an advanced level of tetherless mobile multimedia services requires (1)the development of a wireless network that supports the integrated multimedia services; and (2) the development of intelligent network management middleware services that can provide agile interfaces to mobile multimedia services. Prior work has focused on protocols for mobility management for various types of wireless architectures. Our objective is to provide support for mobility and Quality of Service management (QoS) at the middleware layer independent of the underlying specific network architecture.

To ensure the QoS requirements of mobile multimedia applications and allow more concurrent mobile hosts, resource provisioning techniques are employed to allocate suitable resources for each incoming request. Information required for efficient QoS provisioning includes an accurate representation of resource state information. There exists an inherent tradeoff between information accuracy and system performance. More accurate information leads to better QoS provisioning, however, higher overhead is introduced to maintain the accuracy. In our previous work, we have studied approaches for information collection in non-mobile environments [3].

In this paper, we present a cost-driven approach to designing an information collection framework for mobile environments that addresses the cost/quality tradeoff. Specifically, we characterize the cost involved in each step of the collection process. Incorporating detailed information about each of these costs into the collection process can be time consuming and resource-intensive in itself. We analyze the possibilities to reduce the cost of information collection without degrading the overall QoS delivered by the system. We compare existing approaches with our cost driven approach for network and server related information under different levels of host mobility and user request patterns. Our results are used in the development of an adaptive information collection framework AutoSeC (Automatic Service Composition) [3] that determines (a) which information collection policies to use for specific data elements collected and (b) techniques for determining when the deployed information collection mechanisms must be modified or switched based on user requirements and host characteristics.

The rest of this paper is organized as follows. Section II describes the overall architecture of the system with a focus on the middleware components, briefly presents the previous approaches to information collection. In Section III, we propose a cost driven approach to information collection CDIC and present its optimized variation Opt-CDIC. We present a comparative performance evaluation of various information collection algorithms in Section IV and analyze the obtained results. We conclude with future research directions in Section V.

II. INFORMATION COLLECTION FRAMEWORK

An information collection architecture suited for mobile multimedia environments shown in Figure 1 consists of three components:

- *Information Source:* This corresponds to the managed entity, such as the server, link , mobile host or stationary host. In our system, we use the directory service to hold system state information about information sources. This state information includes network parameters(such as residual link bandwidth, end-to-end delay on links etc.), server parameters (such as CPU utilization, buffer capacity, disk bandwidth, etc.), and mobile host parameters (such as mobile host location, connectivity, power level etc.).
- *Information Consumer:* This module consumes data collected from the information sources (stored in the directory service) for application and system level tasks. For instance, resource provisioning modules consume information about network and system status to perform admission control and resource allocation.
- *Information Mediator:* This module serves as the decision point of the information collection. It listens to notifications from sources or consumers and invokes suitable actions so that the directory service maintains information at a suitable level

0-7803-7605-6/02/$17.00 ©2002 IEEE.

of accuracy satisfactory to the consumers. It initiates sampling at pre-defined sampling frequency and decide policies to alter sampling frequency suitably.

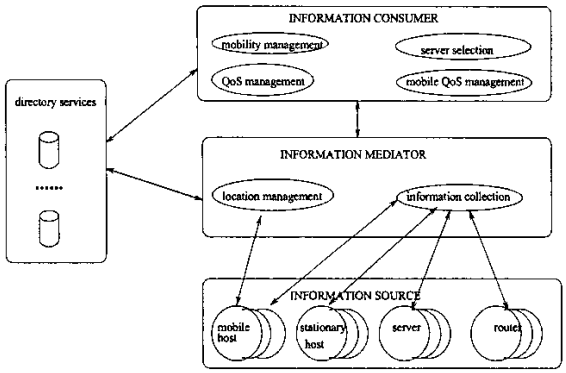


Fig. 1. Information Collection System Architecture

We have studied the impact of combining several information collection policies (snapshot based, static range based, throttle based and time series driven dynamic range based approaches) with resource provisioning techniques under varying network/server conditions and different application workloads [3]. Furthermore, initial studies on information collection for mobile environments indicate that an aggregation based approach [4] (where we use the total number of mobile hosts to drive information collection) outperforms those strategies suitable for non-mobile environments. We define a *Generalized Aggregation Based Information Collection (Gen-ABIC)* approach that uses a range R with an upper bound U and a lower bound L to represent the monitored parameter in the directory service. To begin with, we first partition the underlying topology into non-overlapping regions. Each region is equipped with a collection point that accumulates all the state information of the mobile hosts, servers, links for that region. The Gen-ABIC algorithm consists of two phases: Phase 1 derives the aggregate mobility patterns from individual user mobility patterns and utilizes the aggregation status and current resource utilization status to adjust the collection parameters such as sampling frequency and range size; Phase 2 utilizes feedback from the sources and the consumers (the resource provisioning process in our case) for further customization of the collection process. Every source (managed entity) maintains the current exact value and the approximation held in the directory service. When the current value at the source changes unexpectedly and falls outside of the current range, a *source-initiated trigger* is issued to the information mediator for future action. The feedback from the information consumer may indicate that the current range is not accurate enough to make decisions, this causes a *consumer-initiated trigger*. The information mediator responds by collecting more accurate information.

The CDIC Algorithm:
//phase 1: aggregation driven coarse-grained adjustment of
// collection parameters (invoked periodically)
compute host aggregation level;
compute resource utilization level;
switch(resource utilization)
{
case high: set SF and R to be minimum;
case low: set SF and R to be minimum;
case medium: increase/decrease SF and R
based on current aggregation level;
}

//phase 2: fine-grained adjustment of range size
every certain time period:
compute R based on Equation 1.

Fig. 2. The CDIC Algorithm

III. COST DRIVEN INFORMATION COLLECTION(CDIC)

In this section, we analyze the cost factors involved in a generic information collection process. For instance, servicing and managing source and consumer initiated triggers can be quite expensive; for applications such as resource provisioning, this overhead may even be unnecessary. Careful study of a generic information collection process reveals that the communication cost involved in the whole process consists of:

- regular sampling overhead C_{rs}: from probing messages issued by the mediator to sources;
- regular update overhead C_{ru}: from the update messages which report current source values to the mediator and update directory service with the new values.
- source/consumer-initiated trigger overhead C_{st} and C_{ct}: from the sources/consumers to the mediator that can potentially cause range expansion and tightening.
- source/consumer-initiated directory update overhead C_{su} and C_{cu}: this comes from the messages sent by the mediator to update the directory service with the new range.

We characterize the total cost as the sum of the above costs. For simplicity, we assume the cost of information collection is proportional to the sum of the number of regular samplings N_{rs}, regular updates N_{ru}, source-initiated triggers N_{st}, consumer-initiated triggers N_{ct}, source-initiated directory service updates N_{su} and consumer-initiated directory service updates N_{cu}. So the total overhead is simplified to be

$C_{total} = N_{rs} + N_{ru} + N_{st} + N_{ct} + N_{su} + N_{cu}$

The cost measurement can be used in optimizing any information collection process. We illustrate how such a cost analysis can improve the performance of Gen-ABIC. As we can see, in Gen-ABIC, both types of triggers/updates (source- and consumer-initiated) provide an opportunity for the information mediator to adjust the range size in the directory service. The objective in selecting a good range size is to avoid

the need for future updates since we want to minimize the communication cost. To avoid source-initiated triggers and updates, the range size should be big enough to make it unlikely that changes of the source value will exceed the range; On the other hand,to avoid consumer-initiated triggers and updates, the range size should be small enough to provide accurate enough information for consumer to make decisions. Since decreasing the probability of one type of trigger and update increases the probability of the other type, it is not obvious how best to choose a range size that minimizes the total probability that an update is required. In addition, the number of regular updates is dependent upon the sampling frequency. Therefore, to minimize the communication cost, we need to minimize the cost of both types of triggers and updates, i.e., to minimize $\Omega = C_{st} + C_{ct} + C_{su} + C_{cu}$. Let P_{st}, P_{su}, P_{ct}, P_{cu} represent the probability of source-initiated trigger, source-initiated update, consumer-initiated trigger, consumer-initiated update. Intuitively, P_{st} and P_{su} increases with a smaller range size R, while P_{ct} and P_{cu} decreases with a smaller R. Based on previous studies [5], for a given R, we have $P_{st} = K_{st}/R^2$, $P_{su} = K_{su}/R^2$, $P_{ct} = K_{ct} \cdot R$, $P_{cu} = K_{cu} \cdot R$, where K_{st}, K_{su}, K_{ct} and K_{cu} are model parameters that depend on the nature of the source and the frequency of consumer requests and the distribution of consumer requirements. Based on these, we can re-write the cost expression as follows:

$$\Omega = C_{st} \cdot P_{st} + C_{su} \cdot P_{su} + C_{ct} \cdot P_{ct} + C_{cu} \cdot P_{cu}$$

$$\Omega = C_{st} \cdot K_{st}/R^2 + C_{su} \cdot K_{su}/R^2 + C_{ct} \cdot K_{ct} \cdot R + C_{cu} \cdot K_{cu} \cdot R.$$

Our goal then is to find the value for R that minimizes this expression, which is achieved by finding the root of the derivative. Using this approach, the optimal value for R is

$$\left(\frac{2 \cdot (C_{st} \cdot K_{st} + C_{su} \cdot K_{su})}{C_{ct} \cdot K_{ct} + C_{cu} \cdot K_{cu}}\right)^{1/3}. \quad (1)$$

However, setting the range size based on this formula is not easy unless source change behaviors and application workloads are stable and known in advance since the model parameters depend on these factors. One approach is to monitor these factors at run-time to set K_{st}, K_{su}, K_{ct} and K_{cu} appropriately. Unfortunately, we observe that this monitoring complexity or overhead affects the system performance to a great extent. Therefore, we would like to evaluate the possibilities of decreasing the communication cost without sacrificing overall QoS.

A. The CDIC Algorithm

Figure 2 is the CDIC algorithm. In this algorithm, there are two parameters based on which the collection process adjusts the sampling frequency and range size: (a) mobile host aggregation level and (b) resource utilization level.

• **Mobile Host Aggregation** $A_i(t)$: It is the number of mobile hosts in region i at certain time period t. We derive aggregate mobility from individual host mobility model. We use the incremental mobility model [2] to characterize individual host mobility. In this model, mobile hosts are distributed randomly and move freely in a closed coverage area. The movement of the mobile host is represented by its velocity vector $\vec{v} = (v, \theta)$, where v is the speed and θ is its direction. The location of the mobile host (x, y) and its velocity $\vec{v}$ are updated periodically every δt time units as follows:

$$v(t + \delta t) = min[max(v(t) + \delta v, 0), V_{max}]$$
$$\theta(t + \delta t) = \theta(t) + \delta\theta.$$

where V_{max} is the maximal mobile velocity; δv, the velocity change is uniformly distributed within $(-A_{max} \cdot \delta t, A_{max} \cdot \delta t)$, A_{max} is the maximum acceleration/deceleration of the mobile host. $\delta\theta$ is the change in the mobile host's direction and is uniformly distributed in $(-\alpha \cdot \delta t, \alpha \cdot \delta t)$, where α is the maximal angular change of the mobile host's direction per unit time.

Mobile hosts are distributed randomly and move freely in a closed coverage area with the size of (X_{max}, Y_{max}) which is divided into many non-overlapping equal sized regions with the size of (X_{region}, Y_{region}). Each region has a collection point (e.g. base station) that serves as the wired network access point for all the mobile hosts in its controlled region. If we let $X_{dim} = \lceil \frac{X_{max}}{X_{region}} \rceil$ and $Y_{dim} = \lceil \frac{Y_{max}}{Y_{region}} \rceil$, then there are $N_{region} = X_{dim} \cdot Y_{dim}$ regions in the area. Mobile hosts either move or stop in this area, the population/aggregation in each region changes all the time and can be captured. At a certain time t, a mobile host located at $(x(t), y(t))$ is in region $X_{dim} \cdot \lceil \frac{x(t)}{X_{region}} \rceil + \lceil \frac{y(t)}{Y_{region}} \rceil$. At time t, the aggregation of region i is the number of mobile hosts located in region i, i.e., the cardinality of the set of the mobile hosts who are in this region.

$$A_i(t) = \|\{j | \lceil \frac{x_j(t)}{X_{region}} \rceil = i \bmod X_{dim}, \lceil \frac{y_j(t)}{Y_{region}} \rceil = \frac{i}{X_{dim}}\}\|.$$

We classify the aggregation level $AL(t)$ of region i as being one of the three levels high, medium and low as follows:

$$AL_i(t) = \begin{cases} High & \text{if } A_i(t) \geq N_{mh}/8 \\ Medium & \text{if } N_{mh}/N_{region} \leq A_i(t) < N_{mh}/8 \\ Low & \text{if } A_i(t) < N_{mh}/N_{region} \end{cases}$$

where N_{mh} is the total number of mobile hosts in the system.

• **Resource Utilization Factor** UF: It is the percentage of occupied resources, i.e., the ratio of resources used to the resource capacity. Similarly, we classify resource utilization level into level of high, medium and low as follows:

$$UL(t) = \begin{cases} High & \text{if } UF(t) \geq 0.9 \\ Medium & \text{if } 0.4 \leq UF(t) < 0.9 \\ Low & \text{if } UF(t) < 0.4 \end{cases}$$

We utilize the mobile host aggregation level and resource utilization factor to adjust the collection parameters of sampling frequency and range size.

B. Optimization of the Collectio Process (Opt-CDIC)

In Opt-CDIC, we further reduce the communication overhead by using two techniques: selective triggering and lazy sampling.

•**Selective Triggering:** The triggering process can be significantly optimized to suit the needs of information consumer, the resource provisioning process in our case. Consumer initiated triggers are used when the consumer determines that the level of accuracy of the data supplied is insufficient. Such consumer initiated triggers are impractical in the case of link related information due to the following reasons: (a) QoS-based routing techniques select a path based on aggregate link characteristics, hence it is difficult to determine whether more accurate information is needed for links on an individual basis (b) since the number of links can be large, customizing the collection process on a per-link basis is not a scalable proposition.

Furthermore, we observe that consumer initiated triggers may be necessary only when the required resources for a request closely match the available resources (if there are ample available resources, the request is likely to be accepted anyway). In such a case, we can assume that the server or the network is overloaded; at this time, there is a high likelihood that incoming requests are rejected; hence maintaining accurate directory information will have little use. For the above reasons, we expect that turning off consumer-initiated triggers and updates would not degrade QoS-provisioning performance very much; On the other hand, the collection overhead can be significantly decreased.

•**Lazy Sampling:** Once consumer-initiated triggers and updates are eliminated, the directory service modifications can only be driven by the regular probing process or by source-initiated triggers. To further reduce communication overhead, we apply lazy sampling strategies that can decrease the probing cost even more. The sampling frequency can be reduced in two cases (a) If the number of source-initiated triggers N_{st} in a given period is less than a predetermined value TH_{st}, we can infer that the current approximation of the source value is reasonable (i.e., the source value does not change very dramatically), therefore we can reduce the sampling frequency; (b)if the range is relaxed to exceed a certain value ($1/4$ of the maxima R_{max} as indicated in our experiments), it is likely that the range is large enough to accommodate reasonable changes in the source value, hence we can reduce the sampling frequency. In both cases, we decrease the sampling frequency.

By combining these two optimization strategies, we enhance the phase II of theCDIC algorithm.

IV. PERFORMANCE EVALUATION

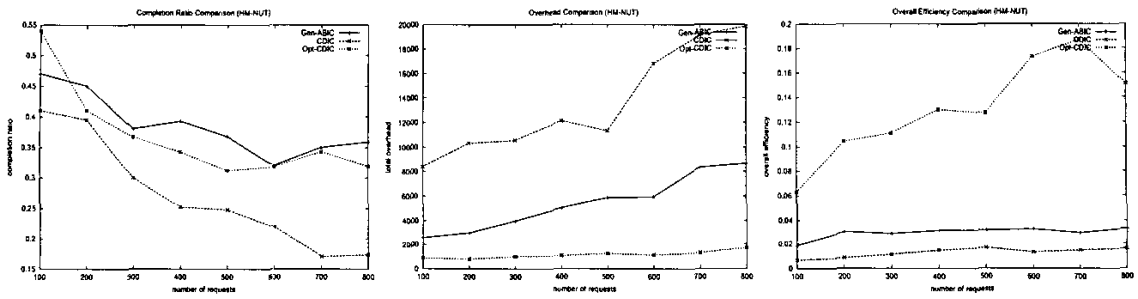
The goal of our simulation is to evaluate and compare performance of resource provisioning under various information collection policies. In the simulation, we use the typical ISP network topology with 18 nodes and 30 links [4] .

Request and Traffic Generation Model: We model request arrival at the source nodes as a Poisson distribution, and the request holding time is exponentially distributed with a pre-specified average value. We pre-define a set of client request templates to capture typical multimedia connection request patterns in terms of network bandwidth, CPU, memory, disk bandwidth and end-to-end delay. For each request generated, the requested parameters are randomly selected from the set of request templates, with the mean requested bandwidth being 2.5Mbps, mean end-to-end-delay being 400ms and CPU, memory and disk bandwidth being 150, 374 and 271 calibrated units respectively. To represent non-uniform traffic, we designate some sets of candidate destinations as being "hot", (i.e. serving popular videos, web sites etc), and they are selected by the clients more frequently than others. To reduce the effect of local and nearby requests, we choose three pairs of source-destination sets from the topology. The requests arrive to these hot pairs, as foreground traffic, at a higher rate than other background traffic. In our non-uniform traffic pattern, we set the foreground arrival rate to be 5 times higher than the background rate and in uniform traffic, we set them to be equal to each other. Specifically we set the foreground arrival rate to be 10 seconds, and the background rate to 50 seconds.

Simulation Scenarios: We study the performance of the three information collection policies (Gen-ABIC, CDIC, Opt-CDIC) under different conditions of mobility. In this simulation, we set $V_{max} = 65mph$, $A_{max} = 0.9meter/sec^2$, $\alpha = 0.1745radian/sec$. We model two levels of mobility: high mobility and low mobility. With high mobility, the velocity change (δv and $\delta\theta$) is twice that of low mobility. Two request patterns described above are considered: uniform traffic and non-uniform traffic. Four different cases we studied in this paper are: HM-UT(High Mobility, Uniform Traffic), HM-NUT(High Mobility, Non-uniform Traffic), LM-UT(Low Mobility, Uniform Traffic), LM-NUT(Low Mobility, Non-uniform Traffic).

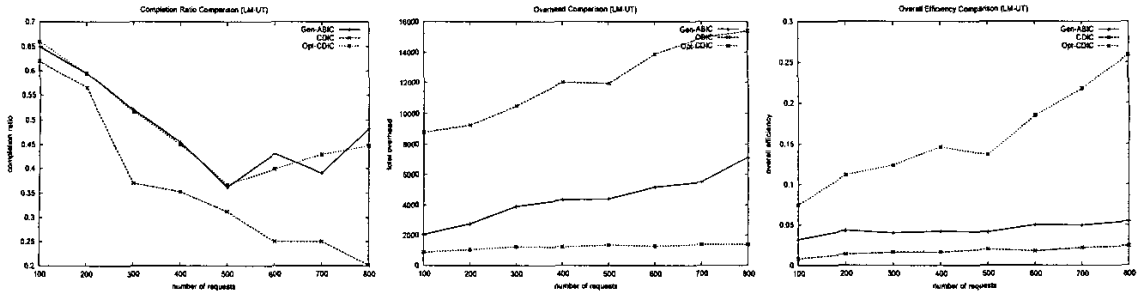
Simulation Results: We analyze the impact of information collection mechanisms on the overall resource provisioning performance. The provisioning mechanism used here is a Composite Path and Server Selection (CPSS) technique [1]. CPSS deals with path and server selection in a unified way, allows load balancing not only among replicated servers, but also among network links to maximize the request success ratio and system throughput. However, our algorithms are not limited to CPSS and they can also be applied to other resource provisioning processes in a similar way. Performance metrics used include the completion ratio (the ratio of the number of completed requests to the total number of requests), overhead and overall efficiency (the ratio of the number of completed requests to the overhead).

Figure 3 shows the performance of the three policies for collecting network and server information under high mobility and non-uniform traffic. We observe that the completion ratio of Gen-ABIC is marginally higher than that of Opt-CDIC but much higher than that of CDIC. As expected, with an increase of the number of requests in the system, the completion



A.completion ratio B.overhead C.overall efficiency

Fig. 3. Collection Policies for Network and Server Information under High Mobility and Non-uniform Traffic



A.completion ratio B.overhead C.overall efficiency

Fig. 4. Collection Policies for Network and Server Information under Low Mobility and uniform Traffic

ratio decreases. We also observe that the overhead of CDIC is the highest and the overhead of Opt-CDIC is the lowest. Although CDIC is expected to perform well theoretically, an accurate relationship between the number of triggers and the range size is difficult to capture in practice, hence the increased overhead. By combining the completion ratio and overhead, Figure 3C shows the overall efficiency. Opt-CDIC exhibits much higher efficiency than the other two policies due to its low overhead. We also studied the performance under the other scenarios: HM-UT, LM-UT (see Figure 4) and LM-NUT. The results obtained are similiar to the HM-NUT case.

We conclude from our studies that a coarse assignment of collection parameters (e.g. sampling frequency and range size) is adequate to render satisfactory completion ratios under most traffic workloads and mobility patterns. Adding source and consumer-initiated triggers and updates makes the collection process more complex and increases overhead without significantly increasing the completion ratio. Optimizations such as turning off consumer-initiated triggering and selectively issuing source-initiated triggers help reduce overhead to a great extent without lowering the completion ratio. In summary, Opt-CDIC is a desirable strategy to collect network and server information in mobile environments.

V. CONCLUSIONS

In this paper, we propose a cost-driven approach to collecting information about system components in mobile environments. We are currently developing a scalable information collection architecture suitable for wide-area environments that incorporates distributed directory services. We believe that adaptation must be incorporated in every component of a QoS-based application infrastructure. Customizable middleware frameworks such as that presented in this paper is key to ensuring system performance while delivering user QoS for multimedia applications in mobile environments.

REFERENCES

[1] Z. Fu and N. Venkatasubramanian. Directory based composite routing and scheduling policies for dynamic multimedia environments. In *Proceedings of IPDPS*, 2001.

[2] Z. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proceedings of the IEEE Int. Conf. on Universal Personal Communications*, October 1997.

[3] Q. Han and N. Venkatasubramanian. Autosec: An integrated middleware framework for dynamic service brokering. *IEEE Distributed Systems Online*, 2(7), 2001.

[4] Q. Han and N. Venkatasubramanian. Aggregation based information collection for mobile environments. *Journal of High Speed Networks*, 2002. To appear.

[5] C. Olston, B. T. Loo, and J. Widom. Adaptive precision setting for cached approximate values. In *Proceedings of the ACM SIGMOD*, 2001.