Network Assisted Mobile Computing with Optimal Uplink Query Processing

Carri W. Chan, *Member, IEEE,* Nicholas Bambos, *Member, IEEE,* and Jatinder Singh, *Member, IEEE,*

Abstract—Many mobile applications retrieve content from remote servers via user generated queries. Processing these queries is often needed before the desired content can be identified. Processing the request on the mobile devices can quickly sap the limited battery resources. Conversely, processing user-queries at remote servers can have slow response times due communication latency incurred during transmission of the potentially large query. We evaluate a network-assisted mobile computing scenario where mid-network nodes with "leasing" capabilities are deployed by a service provider. Leasing computation power can reduce battery usage on the mobile devices and improve response times. However, borrowing processing power from mid-network nodes comes at a leasing cost which must be accounted for when making the decision of where processing should occur. We study the tradeoff between battery usage, processing and transmission latency, and mid-network leasing. We use the dynamic programming framework to solve for the optimal processing policies that suggest the amount of processing to be done at each mid-network node in order to minimize the processing and communication latency and processing costs. Through numerical studies, we examine the properties of the optimal processing policy and the core tradeoffs in such systems.

Index Terms—Dynamic Programming (DP), Network-Assisted Mobile Computing, Network Optimization

1 Introduction

The processing and storage capabilities of mobile consumer devices are becoming increasingly powerful. A gamut of new mobile applications has thus emerged for providing a better quality of experience for the end users. A class of such applications commonly referred to as mobile augmented reality [1]–[3] includes ones that enable delivery of content in response to the user-generated queries for enhancing user's experience of the environment. Text to speech conversion and optical character recognition (OCR) based applications for mobile devices follow a similar paradigm. Several interesting usage scenarios thus arise. A user clicks a picture or shoots a video of a desired object–a building, painting in a museum, a CD cover, or a movie poster–through a camera phone. The video or image is then processed and sent over the network to an application server hosting a database of images. The extracted query image is then matched with a suitable entry and the resulting content–object information, location, title song from a CD, or movie trailer–is then streamed back to the user. A number of existing commercial product provide this type of service [4]–[6]. The processing of query image or video on the phone often involves computationally demanding processes like pattern recognition, background extraction, feature extraction, and feature matching [7]–[10], which when done often can diminish the battery lifetime of the mobile device. Similarly running a text to speech conversion application or an OCR engine for usage scenarios such as listening to a book on mobile device while driving or text extraction from pictures is computationally and battery intensive.

Alternatively, the raw data could be transmitted to the application server where the processing could be done. However this would increase the bandwidth demand over the network with several users using such an application and competing for spectrum along with voice and data traffic generated by users of the wireless network. The first-hop wireless link between the mobile device and base station is often bandwidth constrained and backhaul connections in mobile networks have high capital and operation expenditures per bit. Several wireless carriers have also reported a staggering increase in data traffic over mobile networks because of unprecedented use of mobile data applications [11], [12]. Backhaul links that carry the traffic from edges to the core using copper, fiber or wireless links are associated with significant cost for the carriers [13], [14]. Moreover, the transmission latency on the uplink will be higher as larger query data is transmitted through the network. Thus there is an inherent tradeoff between battery usage and latency. As mobile devices become more sophisticated with higher resolution image and video capabilities, the query data will continue to grow resulting in more demand for intelligent navigation of this tradeoff.

Consider the scenario in Fig. 1. A user request originates at the Mobile Station (MS). In order to be completed, the request must be transmitted upstream to a remote Application Server (AS) via a Base Station (BS) and a series of relay nodes. We refer to the node at the first hop as the base station, but emphasize that the links between the BS, relay nodes, and AS may be wired or wireless. If the request processing is entirely done at the MS, the limited battery power can be drained. On the other hand, if the processing is done at the AS, communication latency can be high due to limited bandwidth of the wireless access link and large query size.

- *C. W. Chan is at Columbia Business School, New York, NY 10023. E-mail: cwchan@columbia.edu*
- *N. Bambos is at Stanford University, Stanford, CA. Email: bambos@stanford.edu*
- *J. Singh is at the Palo Alto Research Center, Palo Alto, CA. Email: jatinder@stanford.edu*

There are a number of systems which enable distributed processing across multiple nodes [15]–[24]. We consider systems with leasing servers which are deployed at mid-network nodes to offer processing capability for the user queries before they reach the AS. Deployment of servers by Akamai [25] constitutes an instance of server leasing capabilities in the network, where uplink queries requesting content are processed without these uplink data having to travel all the way to backend servers. Content Centric Networking (CCN) [26] promulgates an architecture that optimizes uplink bandwidth by aggregating data interest queries on the uplink via intermediate CCN-compliant node processing using name-based addressing internet data. An offshoot of the architecture is deployment of intermediate node caches that process queries for data and respond with content if they have it. Similar methodologies like transparent caching where intermediate nodes in the network respond to queries to data, fall in the intermediate leasing paradigms.

We consider how to utilize network assisted computing to alleviate the processing burden on the MS thereby reducing its battery consumption and extending its operational lifetime. Leasing processing power from mid-network nodes can help lower communication latency because rather than transmitting the entire, large request message over multiple congested links to the AS, mid-network processing will reduce the message size. Introducing the ability to lease processing power from mid-network nodes brings in the tradeoff of leasing cost. As discussed, battery consumption and latency can be reduced by leasing processing power. However, if leasing is costly because of scarce processing capability available at the mid-network nodes or if the users are averse to their data being accessed by the leasing servers, then battery usage and latency will increase. Depending on the relative costs between battery usage, latency, and leasing, it may or may not be beneficial to lease. We examine these tradeoffs in this paper. Using the dynamic programming framework, we solve for the optimal processing policies that suggest amount of processing to be done at a node in the network. The optimization objective is to minimize the processing and communication latency and processing costs. We consider cases where the processing times and leasing costs have linear or concave variation with the amount of processing and assess the properties of the optimal processing policy and the core tradeoffs between leasing cost, latency, batter power, and communication over the wireless access link.

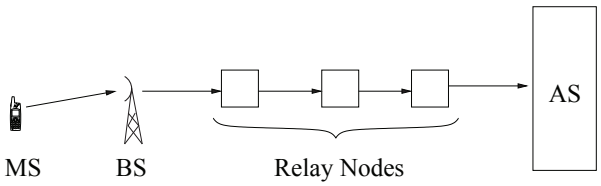


Fig. 1. System Model: Mobile Stations (MS) transmit data to the Application Server (AS) via the Base Station (BS) and relay nodes. The requested data is transmitted back to the mobile device. Links may be wired or wireless.

1.1 Related Work

As mobile applications become more sophisticated and demanding, system operators are utilizing the network to improve service. A substantial amount of work has examined Network-Assisted Computing. However, the main distinction between the previous works and ours is that we focus on allowing processing power to be leased from mid-network nodes and how to make this decision in an optimal manner.

In [27]–[29], Network-Assisted Computing has been examined in the case of cache management. The focus of these works is to determine how to pre-fetch information from a remote server in order to maximize quality of service. Due to the varying quality of the wireless channel, data may not be able to be retrieved at the precise instant it is needed. If that data is not available to the wireless device when needed, the processor will idle until it can be fetched. Pre-fetching is done in a manner to minimize service latency. These works focus on the downlink transmission to make data available and minimize processing times. In contrast, there are applications where the data necessary to complete a request is too large to store at the mobile device. In Mobile Augmented Reality applications, it is infeasible to store even part of the large database required. In the applications we consider, we assume that the request *must* be transmitted uplink to an Application Server in order to be fully satisfied. We focus on the uplink scheduling of how much processing to perform at each node in order to minimize latency, battery usage, and leasing costs.

Even without the ability to lease processing power from mid-network nodes, limited battery resources present a substantial challenge. For a survey of energy efficient protocols for wireless networks, see [30] and the references therein. While batteries are becoming more efficient, the growing sophistication and abundance of applications makes power saving necessary. There has been an extensive body of research on reducing power usage via hardware (see [31]–[34]) and software (see [35], [36]) design. These designs can significantly reduce the amount of battery resources required to process a request. However, a hardware design optimized for one application may be highly inefficient for another. A single device may have a Mobile Augmented Reality application which requires speech processing, while another application requires video processing. As the number of mobile applications increase, all options to save battery resources will prove to be useful.

In most standard Mobile Augmented Reality systems, processing is performed either entirely at the Mobile Station, quickly draining its limited battery resource, or entirely at the Application Server, leading to large communication delays. Most closely to our work is [31], [37]–[41]. These works examine load splitting where processing is split between Mobile Station and Application Server. In [37], [38], the potential battery savings by splitting processing between Mobile Station and Application Server are examined experimentally. In [42], the tradeoff between battery usage and latency is closely examined. Girod et. al. provide an overview of these types of challenges in mobile visual search [43]. Over a 3G network, the transmission of a 50kB image would timeout more than 10% of the time while the transmission of a small 3-4kB query

vector never timed-out. As the sophistication of mobile devices increase, the tradeoff between latency and energy usage will become more critical. A developer at oMoby stated that high latency is the main reason for the use of 50kB queries, but they hope to eventually include high definition images and videos on the order to 1-2MB[1]. In these works, the decision is between local and remote execution of processing tasks. The networks considered are single-hop while we consider multi-hop networks. The main distinction between our work and these works is the idea of cooperating with the mid-network nodes in order to improve the battery versus latency tradeoff. Rather than relying solely on the Mobile Station and Application Server to process a request, we allow for mid-network processing. In this work, an extension to [44], we introduce the idea of "leasing" processing power from mid-network nodes in order to improve quality of service to users.

There has been a steady stream of work on developing systems which allow leasing of processing power which we require. These works focus on the software/OS implementation of an "Active Network" where intermediary nodes can be used to perform computations [15]–[19]. As applications become more demanding and sophisticated, use of such Active Networks will significantly improve system performance. In contrast to this body of work which is primarily centered around system design and deployment, our work focuses how to use such system in an efficient manner. Our work aims to develop a systematic framework to utilize the capabilities of intermediary nodes in such systems.

There has also been some work considering energy and delay sensitive scheduling and partitioning of tasks in collaborative networks [20]–[24]. However, the tradeoffs considered in these works is quite different from ours. The communications saving due to reducing the number of nodes to communication with comes at the cost of reducing the lifetime of the network by draining battery power at each additional node required for communication and processing. In contrast, we do not affect the number of nodes to transmit to, but are able to vary the amount of information that is required to be transmitted by utilizing mid-network processing.

The rest of the paper proceeds as follows. In Section 2, we formally introduce the system model and the idea of Network-Assisted Mobile Computing via leasing. In Section 3, we formulated the optimal processing problem as a shortest path problem and use Dynamic Programming to solve for the optimal policy. While the optimal processing policy can be difficult to solve in general, we identify a number of interesting and useful properties of the optimal policy in Section 4. In Section 5, we examine some of these properties via numerical analysis. Finally, we conclude in Section 6.

2 Problem Formulation

A typical application where Network-Assisted Mobile Computing would be useful is in media applications such as Mobile Augmented Reality. Many mobile devices are equipped with a small camera. In Mobile Augmented Reality, a picture captured by a mobile device corresponds to a request, such as streaming a desired video or audio stream to the mobile device. One of the main technical difficulties of MAR is matching the original picture to the desired media content. A series of image processing techniques are used to do this. The final step requires matching the processed image to the requested content in a large database. It is often the case that this database is so large it cannot feasibly be stored on the limited memory of the mobile device. Therefore, a request must be transmitted uplink to the Application Server. Once the request has been fully processed, the desired content can be streamed downlink to the requesting handheld device. There has been an extensive body of work focusing on the problem of downlink streaming of media content (see [45] and references therein). In this paper, we focus on the uplink transmission and processing of a single original request.

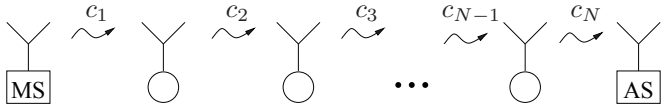


Fig. 2. System Diagram: A request originates at the Mobile Station (MS) and it transmitted over a multihop network to the Application Server (AS). Once the request has reached the Application Server and has been fully processed, it can be satisfied.

The uplink pathway from Mobile Station (MS) to Application Server (AS) is shown in Fig. 2. A request originates at the Mobile Station. In order to locate and stream the desired content, a request message must traverse multiple mid-network hops before arriving at the Application Server. Due to the large file sizes (video/audio streams) which the requests correspond to, as well as the vast number of these files, it is infeasible to store them all on a memory limited mobile device. As such, they are stored in a large database at the remote Application Server and the request must be transmitted upstream in order to be satisfied. The request message must be processed (i.e., speech processing or image processing, feature extraction, feature matching, etc.) before the media stream can be transmitted downstream. See Girod et. al. for an overview of this process [43]. Some tasks are quite simple while others are more complex. There are also a number of scalable media standards which allow simple transcoding by simply discarding bits [46], [47]. In current systems, all of this processing is either done at the MS or the AS. The original request message can be a very large image file and transmitting it over multiple congested links to the AS will result in large delays. If the request were processed prior to transmission, the information needed to be transmitted may be smaller, significantly reducing the communication delay. However, limited computation power and battery resources makes it undesirable to process the entire request at the MS.

The motivation of Network-Assisted Mobile Computing is to improve the Quality of Service of clients subscribing to mobile applications which are often computationally and memory intensive. As the request message traverses network hops, we propose to allow for some processing to be performed at these mid-network nodes. This will mitigate the power drain

1. private communications with developers at oMoby [6]

at the Mobile Station by alleviating the amount of processing required to be executed on the mobile device. Additionally, the large communication delays may be reduced as processing can reduce the message size. The use of Network-Assisted Mobile Computing removes some of the processing burden off the Mobile Station while reducing the size of the request message, and in turn, reducing the communication delays. Certainly, "leasing" the processing power at the mid-network nodes does not come for free, and we examine how to balance the battery life, latency, and leasing costs. In order to study these tradeoff, we must begin by defining the system which we are studying.

2.1 Request Size and Processing Model

A request originates at the Mobile Station. Each request consists of M stages of processing before the desired content can begin streaming to the MS. For instance, M can represent the amount of time required to fully process the request at the MS. Because the processing power at the MS may differ from that at the AS due to different processor types and loads, M is *not* the amount of time required to fully process the request at the Application Server. Therefore, M is a normalized quantity which represents the total amount of processing required to satisfy the request. Certainly M will depend on the particular request and type of data that requires processing.

If z stages of processing have been performed, $M-z$ stages remain. At each node, n, in the network, some processing $0 \leq \delta z \leq M-z$ can be executed. The processing time required to do this is given by:

$$\tau_p(\delta z, n)$$

which is dependent on the amount of processing performed as well at the node at which it is executed. In general, τ_p can take on any functional form. However, we assume that for fixed n, $\tau_p(\delta z, n)$ is increasing in δz, which corresponds to larger processing times as more processing is done.

As more processing is completed, the request message/query data will decrease in size. For instance, the original image may be reduced to a compressed image or an image with the background extracted after some processing is done. In both cases, processing reduces the amount of information that must be relayed to the Application Server to complete the request. Given that z stages of processing have been completed, the size of the request message is given by

$$V(z)$$

which is decreasing in z and is strictly positive. The positivity is required because, even if all processing is completed ($z = M$), a small message must be transmitted to the Application Server so that it knows what content to begin streaming downlink. Without the reception of a request message, the Application Server cannot satisfy a request.

2.2 Networking Model

We now describe the network topology of the system we consider. In order to emphasize the benefits of Network-Assisted Mobile Computing, we assume a tandem network. This allows us to utilize mid-network nodes without unnecessarily complicating the approach with routing decisions, though our framework can be extended to incorporate them. Therefore, our system may reside in a much more complex network with arbitrary topology; however, we will assume that the route from Mobile Station to Application Server is known once the request is made. This is equivalent to assuming the routes are fixed.

Because routes are fixed, we can model the network as an upstream path of $N+1$ network processing nodes in tandem. The request originates at the wireless Mobile Station and must traverse N links to reach the Application Server. The first few hops may be wireless prior to reaching the Base Station/Access Point that connects to the Internet and the next series of hops are wired along the Internet path to the Application Server. At minimum, there is one wireless link between the Mobile Station and Base Station, but there may be others over wireless relays/sensors/etc. Also, at minimum, there is no wireline link; for instance, the Application Server is co-located at the Base Station. However, in general the wireline path to the Application Server could be multihop.

Each link, n (connecting the n^{th} and $(n+1)^{st}$ nodes), is characterized by the capacity of this link, c_n, in bits per second. Therefore, if a message with volume V bits is transmitted along the n^{th} link, it requires V/c_n seconds. Hence, the latency incurred on the n^{th} link after z stages of processing has been performed is:

$$\tau_c(z, n) = \frac{V(z)}{c_n}$$

It is easy to see that τ_c is decreasing in c_n as the link becomes less congested. It is also decreasing in z since $V(z)$ is decreasing in z as mentioned in Section 2.1.

Due to varying path loss, interference, and fading, a wireless channel may be highly varied and randomly varying over time. A wired channel may also be varied due to random congestion in the network. In order to account for this unavoidable physical phenomenon, we assume that the capacity of link n is a random variable with known distribution. We make no assumptions on this distribution other than its expectation, $E[c_n]$, exists and is finite. Therefore, the communication time is a random variable with expectation:

$$E[\tau_c(z, n)] = \frac{V(z)}{E[c_n]}$$

2.3 Leasing Model

Utilizing the processing power of intermediary nodes is the main idea behind Network-Assisted Mobile Computing. Leasing processing power from mid-network nodes can be extremely beneficial to reduce latency and to extend the battery life of a mobile device. However, it comes with a cost. These costs can capture the fee required to lease CPU power from the mid-network nodes. Additionally, these costs may capture potential security risks by giving access of client data to these nodes. Some operations, such as transcoding, can be done on encrypted data, while other would require decrypting the data [48], [49]. We represent these leasing costs by the following

function which is dependent on the amount of processing done, δz, and the node at which it is performed, n:

$$\phi(\delta z, n)$$

On a given node, n, ϕ is increasing in δz, as it is more costly to process more stages. More client data is available to the processing node which could be undesirable. Also, processing more stages requires more processing time so that more power is expended and more congestion is clogging the processors at the mid-network node. If $n = 1$, ϕ represents the cost of processing on the Mobile Station. So rather than encompassing leasing costs, which there are none, it represents the cost of draining battery power as well as tying up the MS processor and preventing the use of other applications. Similarly, if $n = N + 1$, ϕ represents the cost of processing at the Application Server. These costs do not represent leasing costs, as leasing cannot be done at the AS, but can represent the computation power required to process the request which prevents requests from other clients from being completed in a timely manner.

The control dilemma we examine is how much processing should be done at each node given the processing latency, τ_p, communication latency, τ_c, and leasing costs, ϕ. Note that we make no restrictions on the relationships between delay and costs. These relationships should be adjusted according to the types of customers of a particular application and network system. For instance, for customers with strong aversion to delay and are willing to pay extra for fast service, the leasing costs ϕ will be small compared to any delay, τ_p and τ_c. The goal is to determine a computing and transmission control to minimize delay and costs.

3 Optimal Computing/Transmission Control

In order to determine the optimal computing and transmission control, we cast this as a shortest path problem and use Dynamic Programming to find the optimal control [50].

The optimization problem we are trying to solve is to find δz_n, the amount of processing to do at node n given z_n stages have already been processed in order to minimize the total latency and processing costs. The total cost is given by the processing latency, processing costs, and communication latency. The goal is to minimize the expected costs to process the entire request.

$$\min_{\delta z_n} \Big\{ \sum_{n=1}^{N+1} \Big[\tau_p(\delta z_n, n) + \alpha_n \phi(\delta z_n, n)\Big] + \sum_{n=1}^{N} E[\tau_c(V(z_n + \delta z_n))]\Big\} \quad (1)$$
$$\text{s.t.} \sum_{n=1}^{N+1} \delta z_n = M$$

In order to study the core tradeoffs we introduce a scale factor, α_n, to weigh the processing costs at each node. For instance, we can have $\alpha_1 = \beta$, $\alpha_{N+1} = 1$, and $\alpha_n = \alpha$ for $n \neq 1, N+1$. For $\beta = 0$, there is no cost for draining battery at the MS and for $\beta \to \infty$ battery costs are extremely expensive and subsequently little, if any, processing should be done at the MS. If $\alpha = 0$, leasing comes for free and we are mostly concerned with latency. Conversely, if $\alpha \to \infty$, then we are not concerned with latency and processing should be done at the node with the lowest leasing costs.

We can solve the constrained optimization in (1) problem using Dynamic Programming. To begin, we define the state of the system as:

$$(z, n)$$

where $0 \leq z \leq M$ is the amount of processing that has already been completed and $n \in \{1, 2, \ldots, N+1\}$ is the node at which the request message is currently located.

At each state (z, n), the control that needs to be selected is $\delta z \in [0, M - z]$, the amount of processing to perform at node n prior to transmitting the message uplink along the n^{th} link to the $(n+1)^{st}$ node. This decision results in processing latency, τ_p, processing costs, ϕ, and communication latency, τ_c. We can group these into latency ($\tau_p + \tau_c$) and processing costs ϕ. Executing this control changes the system state to $(z + \delta z, n + 1)$.

Define the total expected cost-to-go under policy π starting in state (z, n) by:

$$\begin{aligned} J^{\pi}(z,n) &= E\Big[\sum_{l=n}^{N} \Big\{\tau_p(\pi(z_l,l),l) + \alpha_n\phi(\pi(z_l,l),l) \\ &\qquad + \tau_c(z_l + \pi(z_l,l),l)\Big\} \\ &\quad + \tau_p(M - z_N, N+1) \\ &\quad + \alpha_{N+1}\phi(M - z_N, N+1)\Big] \\ &= \sum_{l=n}^{N} \Big\{\tau_p(\pi(z_l,l),l) + \alpha_n\phi(\pi(z_l,l),l) \\ &\qquad + E\big[\tau_c(z_l + \pi(z_l,l),l)\big]\Big\} \\ &\quad + \tau_p(M - z_N, N+1) + \alpha_{N+1}\phi(M - z_N, N+1) \end{aligned} \quad (2)$$

Then we can define $J^*(z, n)$ as the minimum cost-to-go given that z stages of processing have already been completed and the request resides at node n. $J^*(z, n)$ is given by:

$$\begin{aligned} J^*(z,n) = \min_{0 \leq \delta z \leq M-z} \Big\{&\tau_p(\delta z, n) + E\big[\tau_c(z + \delta z, n)\big] + \\ &\alpha_n\phi(\delta z, n) + J^*(z + \delta z, n+1)\Big\} \end{aligned} \quad (3)$$

Once the request reaches the Application Server, the remaining processing stages must be completed. Therefore, it is easy to see that

$$J^*(z, N+1) = \tau_p(M - z, N+1) + \alpha_{N+1}\phi(M - z, N+1) \quad (4)$$

The optimal policy can be calculated via backward recursion and using Eqn. 3 and 4.

The total cost for servicing a request is given by $J^*(0, 1)$ as a request originates at the Mobile Station, node 1, and no processing has been performed on it yet. This can be broken

into the different components of cost:

$$\begin{aligned} J^*(0,1) &= C^p_{\text{Latency}} + C^c_{\text{Latency}} + \alpha C_{\text{Leasing}} + \beta C_{\text{Battery}} \\ &= C_{\text{Latency}} + \alpha C_{\text{Leasing}} + \beta C_{\text{Battery}} \end{aligned} \quad (5)$$

Where latency can be split into processing and communications latency. The tradeoff factors, α and β shown here demonstrate the competing objectives. For large β, battery is very limited at the Mobile Station, and little processing should be executed there. Conversely, if large α corresponds to large leasing costs and little, if any, processing power should be leased from mid-network nodes.

Define δz_n^* as the amount of processing done at stage n under the optimal policy π^*. Define z_n^* as the amount of processing that has been completed prior to arrival at node n. So,

$$\begin{aligned} J^*(0,1) &= \sum_{n=1}^{N+1} \Big[\tau_p(\delta z_n^*, n) + \alpha_n \phi(\delta z_n^*, n)\Big] \\ &+ \sum_{n=1}^{N} E\left[\frac{V(z_n^* + \delta z_n^*)}{c_n}\right] \end{aligned} \quad (6)$$

In general, it is difficult to determine the optimal processing policy in closed form. In Section 4, we discuss properties of the optimal control under different scenarios. For any cost functions, the optimal solution can be found using numerical analysis. In Section 5, we study the core tradeoffs in Network-Assisted Mobile Computing through numerical analysis.

4 Properties of Optimal Control

The optimal solution of where to process the request, and how much processing to do, is highly dependent on the functional form of the processing times (τ_p), leasing costs (ϕ), message volume (V), as well as communication bandwidth (c_n). However, we can identify some key structural properties of the optimal policy. These properties allow us to determine the optimal processing policy under certain circumstances.

4.1 Monotonicity

We begin by shown some monotonicity results of the optimal value function and optimal processing/transmission policy.

Intuitively, fewer processing stages that remain to be completed will correspond to lower costs. The following proposition formalizes this idea.

Proposition 1: (Monotonicity of J^*) For fixed n, $J^*(z,n)$ is decreasing in z.

Proof: Suppose $z < z'$. Let $\pi^{*'}$ correspond to the optimal policy starting in state (z',n). Now suppose in state (z,n), we use a policy $\tilde{\pi}$ which mimics $\pi^{*'}$ as long as $z < M$. While $z < M$, the processing time and costs for the $\tilde{\pi}$ policy are equal to that of the $\pi^{*'}$ policy and 0 afterwards. Likewise, the communication costs for the $\tilde{\pi}$ policy for z system will be less than those under the $\pi^{*'}$ policy for the z' system. This is because at each node, the total amount of processing completed for the z system is less than that of the z' system since $z < z'$ and the additional amount of processing at each node is equal in each system. Because $V(z)$ is decreasing in z, the communication latency is less. Therefore, $J^{\tilde{\pi}}(z,n) \leq J^*(z',n)$. The result follows by the optimality of J^*, $J^*(z,n) \leq J^{\tilde{\pi}}(z,n)$. □

While one may expect a similar monotonicity result to hold for increasing n, in general it does not hold. It is easy to see this if the processing time and costs at node $n < n'$ is very small and at nodes $m \geq n'$ it is very large. Then, not being able to process any stages at n becomes very costly for the system starting at (z,n').

As the communication link between the Mobile Station and the first network node degrades, communication latency will increase. By processing more stages at the Mobile Station, the request size will decrease, subsequently decreasing the communication latency. Define δz_{MS}^* as the number of stages completed at the Mobile Station.

Proposition 2: (Monotonicity in c_1) For fixed costs, δz_{MS}^* is decreasing as the expected capacity of the first link, $E[c_1]$, increases.

Proof: This is shown via a proof by contradiction. Consider two systems with identical parameters and cost structures, except $c_1 < c_1'$. Let J^* and $J^{*'}$ denote the optimal value function for the c_1 and c_1' systems, respectively. Define δz_{MS}^* and $\delta z_{MS}^{*'}$ as the number of stages processed at the Mobile Station in each system. Assume that $\delta z_{MS}^* < \delta z_{MS}^{*'}$.

By the optimality of δz_{MS}^*:

$$\begin{aligned} J^*(z,1) &= \tau_p(\delta z_{MS}^*, 1) + \alpha_1 \phi(\delta z_{MS}^*, 1) \\ &\quad + E[\tau_c(z + \delta_{MS}^*, 1)] + J^*(z + \delta_{MS}^*, 2) \\ &\leq \tau_p(\delta z_{MS}^{*'}, 1) + \alpha_1 \phi(\delta z_{MS}^{*'}, 1) \\ &\quad + E[\tau_c(z + \delta_{MS}^{*'}, 1)] + J^*(z + \delta_{MS}^{*'}, 2) \end{aligned}$$

which implies:

$$\begin{aligned} &\tau_p(\delta z_{MS}^*, 1) - \tau_p(\delta z_{MS}^{*'}, 1) + \alpha_1 \phi(\delta z_{MS}^*, 1) - \alpha_1 \phi(\delta z_{MS}^{*'}, 1) \\ &\quad \leq E[\tau_c(z + \delta_{MS}^{*'}, 1)] - E[\tau_c(z + \delta_{MS}^*, 1)] \\ &\qquad + J^*(z + \delta_{MS}^{*'}, 2) - J^*(z + \delta_{MS}^*, 2) \\ &\quad \leq 0 \end{aligned} \quad (7)$$

where the second inequality comes from the monotonicity of V and J^* (Proposition 1) and recalling that $\delta z_{MS}^* < \delta z_{MS}^{*'}$.

Now let π be the policy for the c_1' system that uses δ_{MS}^* at $(z,1)$ and the optimal $\pi^{*'}$ after. Then:

$$\begin{aligned} &J^{*'}(z,1) - J^{\pi}(z,1) \\ &\quad = \tau_p(\delta z_{MS}^{*'}, 1) - \tau_p(\delta z_{MS}^*, 1) \\ &\qquad + \alpha_1 \phi(\delta z_{MS}^{*'}, 1) - \alpha_1 \phi(\delta z_{MS}^*, 1) \\ &\qquad + E[\tau_c(z + \delta z_{MS}^{*'}, 1)] - E[\tau_c(z + \delta z_{MS}^*, 1)] \\ &\qquad + J^*(z + \delta z_{MS}^{*'}, 2) - J^*(z + \delta z_{MS}^*, 1) \\ &\quad \geq E[\tau_c(z + \delta z_{MS}^{*'}, 1)] - E[\tau_c(z + \delta z_{MS}^*, 1)] \\ &\qquad + J^*(z + \delta z_{MS}^{*'}, 2) - J^*(z + \delta z_{MS}^*, 1) \\ &\quad \geq 0 \end{aligned} \quad (8)$$

where the first inequality comes from (7) and the second inequality comes for the monotonicity of V and J^*. This

implies that under the c_1' system, processing δz^*_{MS} stages results in lower costs than processing $\delta z^{*'}_{MS}$, which contradicts the optimality of $\delta z^{*'}_{MS}$. Therefore, $\delta z^*_{MS} \leq \delta z^*_{MS}$. □

4.2 Linear Processing and Leasing Costs

Let's consider that case of linear processing times and leasing costs. Therefore, we can define:

$$\tau_p(\delta z, n) = k_n \delta z$$
$$\alpha_n \phi(\delta z, n) = g_n \delta z$$

for some k_n and g_n Recall that the communication time is already linear in the volume of data that must be transmitted. However, $V(z)$ is not necessarily linear.

For a general function for $V(\cdot)$, it is possible to determine if the processing power at an upstream node will never be leased. Let $\gamma_n = k_n + g_n$ so that the total processing cost at node n is:

$$\begin{aligned} C^p(\delta z, n) &= \tau_p(\delta z, n) + \alpha_n \phi(\delta z, n) \\ &= \gamma_n \delta z \\ &= (k_n + g_n)\delta z \end{aligned}$$

γ_n is the incremental cost of completing one processing stage at node n. Because processing reduces the size of data that must be transmitted ($V(z)$ is decreasing in z), there is already a propensity to process at earlier nodes. So if there is a node $m < n$ where the processing costs are cheaper, $\gamma_m < \gamma_n$, then no processing will be done at node n.

Proposition 3: (Linear Costs) Suppose processing costs are linear, such that $C^p(\delta z, n) = \gamma_n \delta z$. Let δz^*_n denote the optimal amount of processing done at stage n under the optimal policy starting from state $(0, 1)$. For all n, if there exists $m < n$ such that $\gamma_m < \gamma_n$, then $\delta z^*_n = 0$.

Proof: The proof is by contradiction. Suppose there exists $m < n$ such that $\gamma_m < \gamma_n$ and $\delta z^*_n > 0$. Now consider a policy $\tilde{\pi}$ that mimics the π^* policy, except at node m and n. Instead of processing δz^*_n at node n and δz^*_m at node m, $\tilde{\pi}$ processes $\delta z^*_n + \delta z^*_m$ at node m and 0 at node n. Because $\gamma_m < \gamma_n$, the processing costs under the $\tilde{\pi}$ policy are less than that of the π^* policy. Note also that the communication latency under the $\tilde{\pi}$ policy is lower than that of the π^* policy since more processing is done earlier, making the size of the transmitted message smaller. Therefore, the total cost under the $\tilde{\pi}$ policy is less than that of the π^* policy, which contradicts the optimality of the π^* policy. Hence, $\delta z^*_n = 0$. □

Even with the communication latency decreasing as more processing is done, it is *not* the case that all processing will necessarily be done at the Mobile Station. This is because processing costs may decrease as the message traverses network hops and so the increase in communication latency is balanced by the decrease in processing costs (both latency and leasing).

If the message volume is a linear function of the number of processing stages completed, then *all* processing will be done at one node.

Proposition 4: (Linear Costs and Volume) Suppose processing costs are linear, such that $C^p(\delta z, n) = \gamma_n \delta z$. Additionally, assume that the message size is linear in the number of stages processed, such that $V(z) = V_0 - hz$. Let δz^*_n denote the optimal amount of processing done at stage n under the optimal policy starting from state $(0, 1)$. Then, there exists one node m such that $\delta z^*_m = M$ and for all other nodes $n \neq m$, $\delta z^*_n = 0$.

Proof: We begin by showing that at any state (z, n), J^* is linear in the number of stages processed, z. That is, there exists β_n and λ_n, such that:

$$J^*(z, n) = \beta_n z + \lambda_n \tag{9}$$

We show this by induction on n. This is clearly true if $n = N$ because costs are linear, so:

$$\begin{aligned} J^*(z, N) &= \tau_p(M - z, N) + \alpha_N \phi(M - z, N) \\ &= \gamma_N (M - z) \\ &= \beta_N z + \lambda_N \end{aligned} \tag{10}$$

where $\beta_N = -\gamma_N$ and $\lambda_n = \gamma_N M$.

Now, we assume that $J^*(z, n+1)$ is linear in z and show that it holds for $J^*(z, n)$.

$$\begin{aligned} J^*(z, n) &= \min_{\delta z} \Big\{ \tau_p(\delta z, n) + E\big[\tau_c(z + \delta z, n)\big] + \\ &\qquad \alpha_n \phi(\delta z, n) + J^*(z + \delta z, n + 1) \Big\} \\ &= \min_{\delta z} \Big\{ \gamma_n \delta z + \frac{V_0 - h(z + \delta z)}{E[c_n]} + \\ &\qquad \beta_{n+1}(z + \delta z) + \lambda_{n+1} \Big\} \\ &= \min_{\delta z} \Big\{ a^0_n \delta z + a^1_n z + a^2_n \Big\} \\ &= \begin{cases} a^1_n z + a^2_n, & a^0_n > 0; \\ a^0_n (M - z) + a^1_n z + a^2_n, & a^0_n \leq 0. \end{cases} \end{aligned} \tag{11}$$

for some constants a^0_n, a^1_n, and a^2_n. We can see that $J^*(z, n)$ is clearly linear in z. Due to the linear dependence on z and δz, if $a^0_n \leq 0$ then it is optimal to process *all* remaining stages at node n; otherwise, it is optimal to process none. This immediately yields the desired result. If there exists a node n where $a^0_n \leq 0$, then all processing will be performed at that node. If there are multiple nodes with $a^0_n \leq 0$, then all processing will be performed at the earliest one. Now, if there are no nodes with $a^0_n \leq 0$, then no processing will be done at any node $n < N + 1$. Since all processing must be completed in order to process the request, all processing must be done at node $N + 1$, the Application Server. □

Linear costs are reasonable when processing is charged on a per-stage basis. However, it is sometimes that case that "processing in bulk" may reduce costs. We now turn our attention to this scenario where costs are concave.

4.3 Concave Processing Times and Leasing Costs

Let us consider the case where processing times and leasing costs are concave functions in the number of stages processed. So that

$$\frac{\partial^2 \tau_p}{\partial \delta z^2} < 0 \text{ and } \frac{\partial^2 \phi}{\partial \delta z^2} < 0.$$

For notational simplicity, let $f_n(\delta z) = \tau_p(\delta z, n) + \alpha_n \phi(\delta z, n)$. It is easy to see that f_n is also concave in δz.

Now suppose that the benefit of processing in bulk is diminishing in n. That is,

$$\max_{z'} \left.\frac{\partial^2 f_n}{\partial \delta z^2}\right|_{z=z'} \leq \min_{z'} \left.\frac{\partial^2 f_{n+1}}{\partial \delta z^2}\right|_{z=z'}$$
$$and$$
$$\left.\frac{\partial f_n}{\partial \delta z}\right|_{z=0} \leq \left.\frac{\partial f_{n+1}}{\partial \delta z}\right|_{z=0} \quad (12)$$

An example of these types of cost functions can be seen in Fig. 3 where $\tau(\delta z, n) + \alpha_n \phi(\delta z, n)$ are quadratic functions of δz. Each solid thick line corresponds to the cost of processing on that node and the lighter lines correspond to the cost of processing on earlier nodes. We can see that the cost function for later node dominates that of the earlier nodes. Under examination of these functions, the increasing costs of processing suggest that most processing is performed at the first node. In fact, under conditions (12), it is optimal to process all stages at the Mobile Station.

Proposition 5: (Concave Costs) If the first and second derivatives of concave f_n satisfies (12), then $\delta z^*_{MS} = M$ and for all $n > 1$, $\delta z^*_n = 0$.

Proof: The proof of this claim is via by contradiction. Assume there exists some intermediary node $m \neq 1$ such that $\delta z^*_m > 0$. Now consider a policy $\tilde{\pi}$ such $\tilde{\delta z}_n = \delta z^*_n$ for all $n \neq 1, m$, while $\tilde{\delta z}_1 = \delta z^*_1 + \delta z^*_m$, and $\tilde{\delta z}_m = 0$. That is instead of following the optimal policy precisely, $\tilde{\pi}$ processes the stages for node m at the Mobile Station. let $\tilde{z}_n$ and z^*_n denote the number of stages processed prior to node n.

$$\begin{aligned}
J^{\tilde{\pi}}(0,1) &= \sum_{n=1}^{N+1} \Big[\tau_p(\tilde{\delta z}_n, n) + \alpha_n \phi(\tilde{\delta z}_n, n)\Big] \\
&+ \sum_{n=1}^{N} \Big[E[\frac{V(\tilde{z}_n + \tilde{\delta z}_n)}{c_n}]\Big] \\
&= \sum_{n=1}^{N+1} \Big[f_n(\tilde{\delta z}_n)\Big] + \sum_{n=1}^{N} \Big[E[\frac{V(\tilde{z}_n + \tilde{\delta z}_n)}{c_n}]\Big] \\
&= \Big\{\sum_{n=1}^{N+1} \Big[f_n(\delta z^*_n)\Big] + \sum_{n=1}^{N} \Big[E[\frac{V(\tilde{z}_n + \tilde{\delta z_n})}{c_n}]\Big]\Big\} \\
&+ \sum_{n=1,m} \Big[f_n(\tilde{\delta z}_n) - f_n(\delta z^*_n, n)\Big] \\
&\leq J^*(0,1) + \sum_{n=1,m} \Big[f_n(\tilde{\delta z}_n) - f_n(\delta z^*_n)\Big] \\
&\leq J^*(0,1) \quad (13)
\end{aligned}$$

The first inequality comes from the fact that V is decreasing in z and $\tilde{z}_n \leq z^*_n$. The last inequality comes from the concavity property, (12), which implies that $\sum_{n=1,m}[f_n(\tilde{\delta z}_n) - f_n(\delta z^*_n)] \leq 0$. This contradicts the optimality of J^*, hence $\delta z^*_m = 0$. □

Using a similar argument, if

$$\max_{z'} \left.\frac{\partial f_n}{\partial \delta z}\right|_{z=z'} \leq \min_{z'} \left.\frac{\partial f_{n+1}}{\partial \delta z}\right|_{z=z'} \quad (14)$$

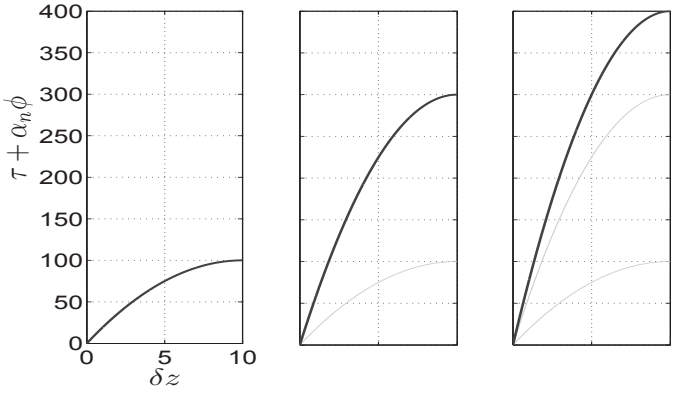


Fig. 3. Cost functions which exhibit diminishing benefits of processing in bulk. Each solid thick line corresponds to the cost of processing on that node. The lighter lines correspond to the cost of processing on prior nodes.

we can show the following proposition

Proposition 6: (Increasing Costs) If the first derivative of concave f_n satisfies (14), then $\delta z^*_{MS} = M$ and for all $n > 1$, $\delta z^*_n = 0$.

Proof: The proof of this claim is via a proof by contradiction, similar to the proof of Proposition 5. Under condition (14), Eqn. 13 still holds. □

If all processing costs and times are equal and concave, then $f_n = f, \forall n$. In this case, f_n clearly satisfies (12).

Proposition 7: (Identical Concave Costs) If $\tau_p(\delta z, n) = \tau_p(\delta z)$ and $\phi(\delta z, n) = \phi(\delta z)$ for all n and $\tau_p(\cdot)$ and $\phi(\cdot)$ are concave, then $\delta z^*_{MS} = M$ and for all $n > 1$, $\delta z^*_n = 0$.

Proof: This is a direct consequence of Proposition 5. □

Now suppose that instead of all nodes satisfying (12), there exists a series of nodes $m, m+1, \ldots, m_n$ which satisfy (12). Then if any processing is done on these nodes, it is *all* done at node m.

Proposition 8: (Series of Concave Costs) If there exists $m, m+1, \ldots, m_n$ whose f_m are concave and satisfy (12), then $\delta z^*_n = 0$ and for all $n \in \{m+1, m+2, \ldots, m_n\}$.

Proof: This can be shown via a proof similar to that of Proposition 5. We omit the details to avoid repetition and for the sake of space. □

A scenario where this may apply is if all intermediate network nodes are identical. If the processing times and leasing costs on these nodes are concave and equal, then $m = 2$ and $m_n = N$. If any processing is leased, all of it is leased from the first intermediary node, $m = 2$. The remaining processing is done at the Mobile Station and Application Server.

All of the preceding results corresponding to concave cost functions are independent of the volume function, $V(z)$. It may very well be the case that processing times are concave since processing multiple stages at once can eliminate some file input/output overhead. It is also likely that the midnetwork nodes will be identical, so that Proposition 8 will apply.

4.4 Constant Communication Times

In some cases, communication times may be independent of the message size. This may occur if the original message size

(before processing) fits into the size of a single network packet. Often, a single packet is the finest granularity with which information can be transmitted. So, while further processing may reduce the message size, the amount of information transmitted must be placed into a standard network packet with padding if the message is very small. Hence, no matter how much processing is done, the transmission times are given by the size of a network packet. Therefore, communication latency will be constant and independent of the policy and we can ignore communication times in the optimization.

We can also ignore communication times if processing does not affect the query data size. For instance, if processing corresponds to linear transformations of the original image (rotation, wavelet decomposition, etc.) so that processing requires time and computation power, but does not modify the amount of information that needs to be transmitted, then we can ignore communication times. This is because the total communication latency will be independent of how much processing is done and at which node it is performed.

Here we will assume that $c_n \to \infty$ so that $\tau_c(z,n) \to 0$. As we have mentioned, if $\tau_c(z,n) = K_n$ is some constant independent of z, then we can ignore it in the optimization problem, so it is similar to assuming $\tau_c(z,n) = 0$. We can rewrite the optimization problem in (1) as:

$$\begin{aligned} &\min_{\delta z_n} \Big\{ \sum_{n=1}^{N+1} \Big[\tau_p(\delta_z, n) + \alpha_n \phi(\delta z, n)\Big] + \sum_{n=1}^{N} K_n \Big\} \\ &\text{s.t. } \sum_{n=0}^{N+1} \delta z_n = M \end{aligned} \tag{15}$$

which results in the same δz_n^* as the following constrained minimization problem without communication costs:

$$\begin{aligned} &\min_{\delta z_n} \sum_{n=1}^{N+1} \Big[\tau_p(\delta_z, n) + \alpha_n \phi(\delta z, n)\Big] \\ &\text{s.t. } \sum_{n=0}^{N+1} \delta z_n = M \end{aligned} \tag{16}$$

Bellman's equations can be rewritten as:

$$\begin{aligned} J^*(z,n) = \min_{0 \le \delta z \le M-z} \Big\{ &\tau_p(\delta z, n) + \alpha_n \phi(\delta z, n) + \\ &J^*(z + \delta z, n+1) \Big\} \end{aligned} \tag{17}$$

Again, once the request reaches the Application Server, the remaining processing stages must be completed.

$$J^*(z, N+1) = \tau_p(M-z, N+1) + \alpha_{N+1}\phi(M-z, N+1) \tag{18}$$

If $\delta z_n \in [0, M-z]$ can be fractional, the minimization problem in (16) can be solved using Lagrangian techniques.

Proposition 9: (Constant Communication Costs) If for some constant $k > 0$, $\tau_c(z,n) = k$ for all z and n, there exists λ^* such that δz_n^* satisfies for all n:

$$\frac{\partial(\tau_p + \alpha_n \phi)}{\partial \delta z}\Big|_{\delta z_n^*} = \lambda^*$$

Proof: This can be shown via a proof by contradiction. Let's suppose that the claim does not hold true. Then, there exists m and m' such that:

$$\lambda_m^* = \frac{\partial(\tau_p + \alpha_n \phi)}{\partial \delta z}\Big|_{\delta z_m^*} < \frac{\partial(\tau_p + \alpha_n \phi)}{\partial \delta z}\Big|_{\delta z_{m'}^*} = \lambda_{m'}^*$$

Define $\tilde{\pi}$ as the policy which mimics the optimal policy π^*, except at nodes m and m'. Therefore, $\tilde{\delta} z_n = \delta^* z_n$ for all $n \neq m, m'$ and $\tilde{\delta} z_m = \delta^* z_m + \epsilon$ and $\tilde{\delta} z_{m'} = \delta^* z_{m'} - \epsilon$ for some small $\epsilon > 0$. For notational simplicity let $f(\delta z, n) = \tau_p(\delta z, n) + \alpha_n \phi(\delta z, n)$.

$$\begin{aligned} J^*(0,1) &- J^{\tilde{\pi}}(0,1) \\ &= \sum_{n=0}^{N+1} \Big[f(\delta z_n^*, n) - f(\tilde{\delta} z_n, n) \Big] \\ &= -\Big[f(\delta z_m^* + \epsilon, m) - f(\delta z_m^*, m) \Big] \\ &\quad + \Big[f(\delta z_{m'}^*, m') - f(\delta z_{m'}^* - \epsilon, m') \Big] \\ &= -\frac{\partial(\tau_p + \alpha_n \phi)}{\partial \delta z}\Big|_{\delta z_m^*} + \frac{\partial(\tau_p + \alpha_n \phi)}{\partial \delta z}\Big|_{\delta z_{m'}^*} \\ &= \lambda_{m'}^* - \lambda_m^* \\ &> 0 \end{aligned} \tag{19}$$

This contradicts the optimality of J^*. Therefore, there does not exist m, m' such that $\lambda_m^* < \lambda_{m'}^*$. □

The lack of affect on communication delays generates a very interesting contrast to the optimal policies defined by concave costs given in Section 4.3. Consider a system with 3 nodes, a Mobile Station, a Base Station, and an Application Server as in Fig. 4.

Suppose that costs are identical on each node so that for all n, $\tau_p(\delta z, n) = \tau_p(\delta z)$ and $\alpha_n \phi(\delta z, n) = \alpha \phi(\delta z)$. Therefore, the cost function at each node is $f(\delta z) = \tau_p(\delta z) + \alpha \phi(\delta z)$. Let's consider the case where there are 9 processing stages and $f(\delta z)$ is concave, as in Fig. 5. For this example, we consider the case of $f(\delta z) = 20\delta z - \delta z^2$, for $\delta z \in [0, 9]$.

Under variable communication costs, $\tau_c > 0$, by Proposition 7, all processing is done at the Mobile Station, and no processing is done on other stages. Conversely, when communication costs are constant, $\tau_c = k$, Proposition 9 implies that equal processing is done at the MS, BS, and AS. This is because the cost functions are identical, and so the δz_n^* which achieves derivative λ^* are identical. The two policies are compared

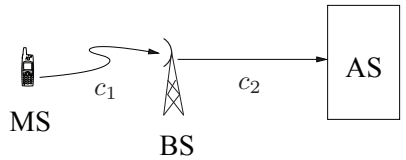


Fig. 4. Simple System Diagram: A request originates at the Mobile Station (MS) and is transmitted over one hop to the Base Station (BS) and finally to the Application Server (AS). Once the request has reached the AS and has been fully processed, it can be satisfied.

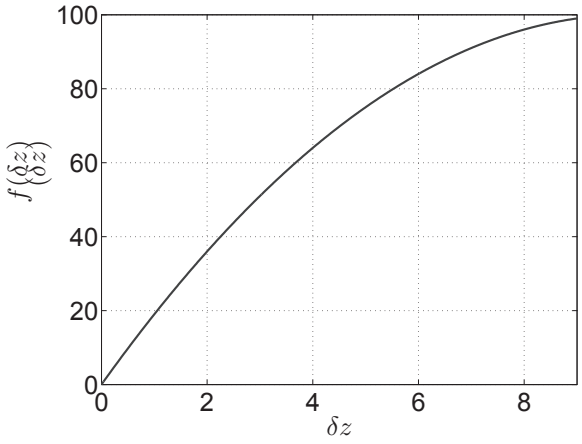


Fig. 5. Cost Function: total processing latency and costs as a function of amount of processing completed.

in Fig. 6. Because of latency due to communication of the request message, there is a propensity to process stages at earlier nodes. This will reduce the message size and, in turn, the amount of latency. However, when communication latency is not a factor, the location of each node is irrelevant–the influencing factor is the difference in incremental cost of processing at each node.

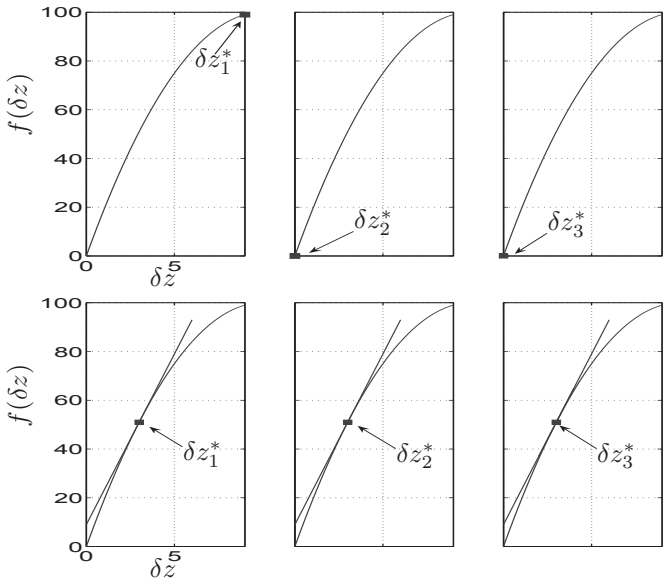


Fig. 6. Optimal processing policy for $\tau_c > 0$ varying (top) and $\tau_c = k$ constant (bottom)

4.5 General Costs

In general, communication latency will depend on the request message size which depends on the amount of processing completed. In this case, communication costs are not negligible and optimality condition in Proposition 9 must be relaxed.

Proposition 10: (Non-Negligible Communication Costs) If $\delta_n^* \neq 0, M$, then λ_n^* is non-increasing in n where λ_n^* is defined as:

$$\frac{\partial(\tau_p + \alpha_n\phi)}{\partial \delta z}\Big|_{\delta z_n^*} = \lambda_n^*$$

Proof: This can be shown via a proof by contradiction similar to the proof for Proposition 9. Let's suppose that the claim does not hold true. Then, there exists m and m' such that, $m < m'$ and:

$$\lambda_m^* = \frac{\partial(\tau_p + \alpha_n\phi)}{\partial \delta z}\Big|_{\delta z_m^*} < \frac{\partial(\tau_p + \alpha_n\phi)}{\partial \delta z}\Big|_{\delta z_{m'}^*} = \lambda_{m'}^*$$

Again $\tilde{\pi}$ mimics the optimal policy π^*, except at nodes m and m'. Therefore, $\tilde{\delta}_n = \delta_n^*$ for all $n \neq m, m'$ and $\tilde{\delta}_m = \delta_m^* + \epsilon$ and $\tilde{\delta}_{m'} = \delta_{m'}^* - \epsilon$ for some small $\epsilon > 0$. Define $\tilde{z}_n$ and z_n^* as the amount of processing that has been completed up to and including node n under policy $\tilde{\pi}$ and π^*, respectively. Note that by the definition of $\tilde{\pi}$ and π^*, $\tilde{z}_n \leq z_n^*$ because processing is done earlier under the $\tilde{\pi}$ policy. Again, let $f = \tau_p + \alpha\phi$.

$$\begin{aligned}
J^*(0,1) &- J^{\tilde{\pi}}(0,1) \\
&= \Big\{ \sum_{n=0}^{N+1} f(\delta z_n^*, n) + \sum_{n=0}^{N} \tau_c(z_n^*, n) \Big\} \\
&\quad - \Big\{ \sum_{n=0}^{N+1} f(\tilde{\delta z}_n, n) + \sum_{n=0}^{N} \tau_c(\tilde{z}_n, n) \Big\} \\
&= -\frac{\partial(\tau_p + \alpha_n\phi)}{\partial \delta z}\Big|_{\delta z_m^*} + \frac{\partial(\tau_p + \alpha_n\phi)}{\partial \delta z}\Big|_{\delta z_{m'}^*} \\
&\quad + \sum_{n=0}^{N} \Big[\tau_c(z_n^*, n) - \tau_c(\tilde{z}_n, n) \Big] \\
&= \lambda_{m'}^* - \lambda_m^* \\
&\quad + \sum_{n=0}^{N} \Big[\tau_c(z_n^*, n) - \tau_c(\tilde{z}_n, n) \Big] \\
&> \lambda_{m'}^* - \lambda_m^* \\
&> 0 \qquad (20)
\end{aligned}$$

The first inequality comes because $\tilde{z}_n \leq z_n^*$ and because τ_c is decreasing in z as described in Section 2.2. This contradicts the optimality of J^*. Therefore, there does not exist $m < m'$ such that $\lambda_m^* < \lambda_{m'}^*$. □

The communication latency has a significant affect on the optimal processing policy which we saw in our example with quadratic processing costs in Fig. 5. It is easy to see in Fig. 6 that with non-negligible communication times, $\tau_c > 0$, λ_n^* is non-decreasing in n, which seems to contradict Proposition 10. However, it does not because $\delta_1^* = 9 = M$ and $\delta_2^* = \delta_3^* = 0$. These boundary cases make it impossible to use the interchange to policy $\tilde{\pi}$ because we cannot increase δ_1^* nor can we decrease δ_2^* or δ_3^*.

5 Numerical Analysis

In the previous section we identified special properties of the optimal processing policy under various scenarios. We now examine some of these properties through numerical studies with example cost functions and systems. Latency, battery

usage, and leasing costs have a tightly woven relationship. Increasing battery usage will decrease latency and leasing costs, but also limits the lifetime of the mobile device. Conversely, the lifetime of the device can be extended by increasing leasing costs which will decrease latency and battery usage.

For our studies, we assume a request requires 10 stages of processing. The size of the original request is 500 kilobytes (roughly the size of a JPEG image) and after completing all stages of processing, it is 1000 bytes, for a reduction in size by a factor of 500. Note that this query may be a JPEG image, short video or audio clip, or some other type of data. The decrease in request size is quadratic in the number of stages that have been completed, z, so that $V(z) = 5(z-10)^2 + 1$ kilobytes. The processing time is linear in the number of stages completed and is dependent on the node it is being processed on, so that $\tau_p(\delta z, n) = k_n \delta z$ for some set of k_n.

We consider a network with 10 nodes, including the Mobile Station and Application Server. Therefore there are 8 intermediary nodes where processing power can be leased. Each mid-network is identical in that the processing time and leasing costs are identical. We also assume they are linear in the number of stages processed so that, for $n \neq 1, N+1$, $\tau_p(\delta z, n) = k\delta z_n$ and $\phi(\delta z, n) = g\delta z_n$. In this case, Proposition 8 applies to the series of mid-network nodes. Therefore, if any processing is leased, then it will *all* be leased from the first intermediary node, node $n = 2$.

We examine the case where the leasing costs $\phi = 1$ for all n. Therefore, the resulting leasing cost is equal to the number of processing stages leased. The processing time for one stage at the Mobile Station is 100 milliseconds, while it is a constant ratio less, $\frac{100}{r} < 100$ms, at the intermediary nodes, and $\frac{100}{r^2}$ms at the Application Server. The bandwidth of the wireless links is uniformly distributed between $5 - 10$ Mbits/second.

In Fig. 7, we see the tradeoff between leasing, in terms of the number of processing stages performed on mid-network nodes, and latency, in terms of processing and communication

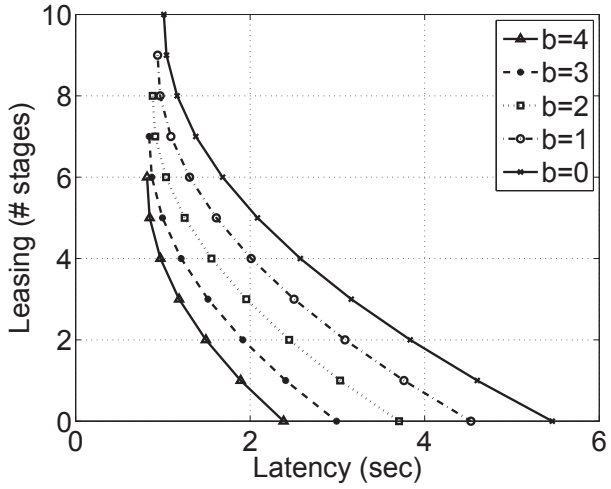


Fig. 7. Leasing vs. Latency for different number of stages (b) processed on the battery limited Mobile Station, i.e. $b = 0$ means no stages are processed at the MS.

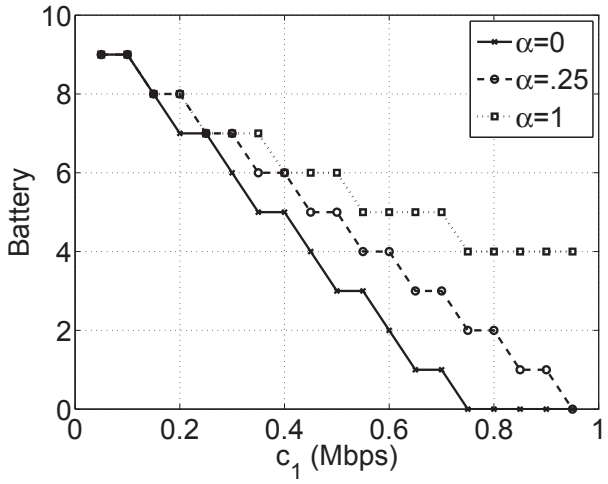


Fig. 8. Battery Usage vs. c_1, throughput of first network hop. For various tradeoff levels between Leasing costs and Latency.

time in seconds, for different amounts of battery usage, in terms of number of stages processed on the Mobile Station. As expected, as the battery usage increases, leasing and latency both decrease. Despite the slow processing times at the Mobile Station, processing stages at the MS can still reduce latency because of the decrease in communication latency that comes with reducing the message size. In this case, the reduction in communication latency is larger than the increase in processing latency. It's interesting to note that for extremely delay sensitive applications where response times must be around one second, leasing should be done very aggressively. In fact, all remaining processing should be leased from the intermediary nodes in order to avoid high delays due to communication over the potentially congested wireless links.

In some instances, the first link may be highly congested and processing at the Mobile Station becomes imperative otherwise large delays will ensue. This particularly may occur if the Base Station is also the Access Point to the wired network. Therefore, the connection between MS and first node is wireless, while the rest of the links are wired with much larger capacity. In Fig. 8, we see how the amount of processing done on the MS varies with the average throughput of the first hop between MS and intermediary nodes. As given by Proposition 2, the number of stages processed on the Mobile Station, and subsequently the amount of battery energy that is drained, decreases as the quality of the first communication link improves. As the channel improves, even large messages can be transmitted without incurring large communication delays. Therefore, in order to save battery power, less processing is done at the MS while communication latency is not vastly affected. When the channel quality is very high, no processing will be performed at the MS. Each line corresponds to different α values to weight the importance between leasing costs and latency. For larger α, leasing becomes more expensive and less desirable. Therefore, to avoid lengthy delays due to the transmission of such a large file, more processing must be

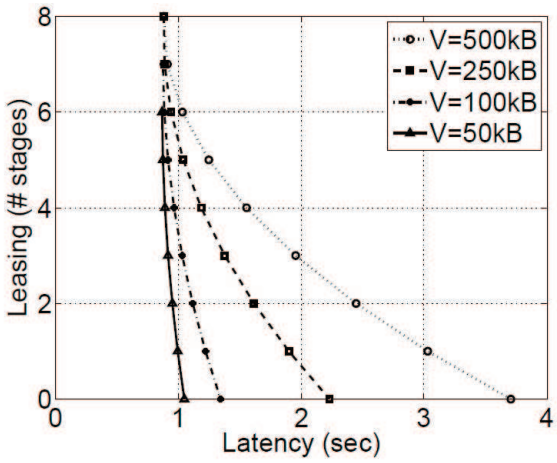


Fig. 9. Leasing vs. Latency for different files sizes (V).

done at the MS to reduce the size of the request message.

Query sizes may vary due to the diversity in mobile devices and applications. We explore how the tradeoff between leasing and latency and the battery usage versus throughput of the first network hop changes with the size of the original query request. We consider the same scenario as before; however, we vary the size of the original request varies from 500 kilobytes to 50 kilobytes. There are still 10 stages of processing and after completing all stages, the request is reduced to 1000 bytes. Hence, after z stages have been completed: $V(z) = V(z-10)^2+1$ for $V = 500, 250, 100, 50$. Fig. 9 is analogous to Fig. 7 with battery usage $b = 2$ and varying file sizes. We can see that even with smaller initial file sizes, leasing is still used sometimes, though much less frequently than when the file size is large. Fig. 10 is analogous to Fig. 8 with tradeoff factor $\alpha = .25$ and varying file sizes. As expected, with smaller file sizes, there is less battery usage for the same throughput of the first hop link. We see that even for the smallest original file size, 50 kilobytes, some processing may be done at the base station when the throughput is very low and communication latency is high. Despite the quantitative differences which arise for varying query sizes, we can see that the fundamental tradeoffs which we have discussed in this paper are irrespective of the specific file size. For all subsequent numerical experiments, we assume that $V = 500$ so that the original query size is 500 kilobytes, recognizing that the qualitative results will hold for other query sizes.

Processing times on the nodes vary due to the different types of processors they may have. For instance, the processor in the Mobile Station may be very limited compared to that of the remote Application Server which may have access to a high powered rack of CPUs. Because $\tau_p(\delta z, 1) = 100\delta z$ms, $\tau_p(\delta z, n) = \frac{100}{r}\delta z$ms ($n \neq 1, N+1$) and $\tau_p(\delta z, N+1) = \frac{100}{r^2}\delta z$ms, r captures the variance between these processing times. The larger the value of r, the more disparate the processing times on each node. Because the processing times per stage improve from the MS to the intermediary nodes to the AS, one suspects that as r increases, latency will decrease significantly. Fig. 11 shows this trend when no processing is done at the MS. It is interesting to note that when jumping from $r = 1$ to $r = 2$ the decrease in latency is much more significant than the jump from $r = 4$ to $r = 20$. Despite the fact that the increase in r corresponds to a decrease in delay, for very large r, the delay is mostly due to communication of the request message rather than processing times.

We now consider nonlinear processing costs in the case of 4 network nodes, 2 from which processing power can be leased. The following experimental setup is identical as before; however, now $\phi(\delta z, n) = \xi_n(20\delta z - \delta z^2)$ where $\xi_1 = 1$, $\xi_2 = \frac{1}{3}$, $\xi_3 = \frac{1}{4}$, and $\xi_4 = \frac{1}{10}$. Fig. 12 and 13 shows the optimal leasing versus latency tradeoff for various battery usages for the first and second mid-network nodes, respectively. Because $\xi_3 < \xi_2$, the leasing costs on the second mid-network node is less than that for the first mid-network node. However, due to communications latency, the first mid-network may still be used. We can see that in order to decrease latency, more processing should be performed at the first mid-network node. Conversely, if leasing costs are more important than latency, it is beneficial to incur an increase in communication latency in order to process at the second mid-network node for lower costs.

We have seen that battery usage, latency (both due to processing and communication), and leasing costs are highly intertwined. These costs are also highly dependent on system parameters such as communication bandwidth; processor speeds at the MS, AS, and intermediary nodes; as well as request message size as a function of the number of stages processed. By studying these tradeoffs, we can gain a better understanding of the relationships between each cost. This knowledge will help future system design. From a user's perspective, one must determine how much processing power to lease from mid-network nodes in order to satisfy delay constraints and extend battery life. From a network administrator's perspective, one must determine how much to charge for leasing processing power in order to encourage users to

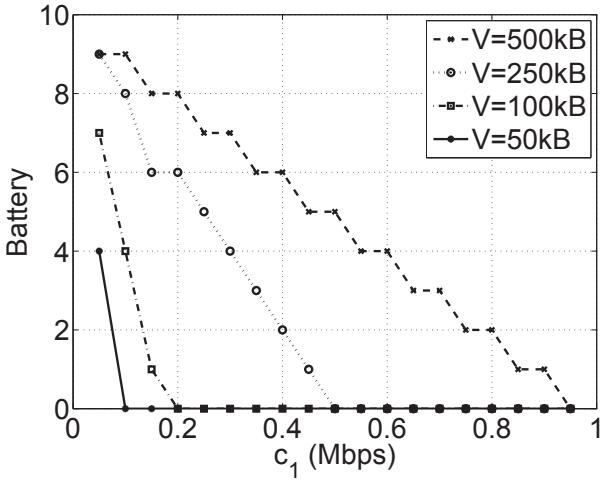


Fig. 10. Battery Usage vs. c_1, throughput of first network hop. For various tradeoff levels between Leasing costs and Latency.

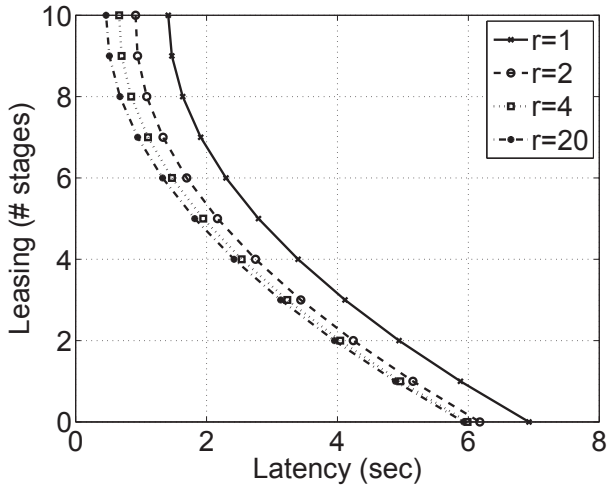


Fig. 11. Leasing vs. Latency for various values of the ratio between processing times on each node, $\frac{1}{r}$.

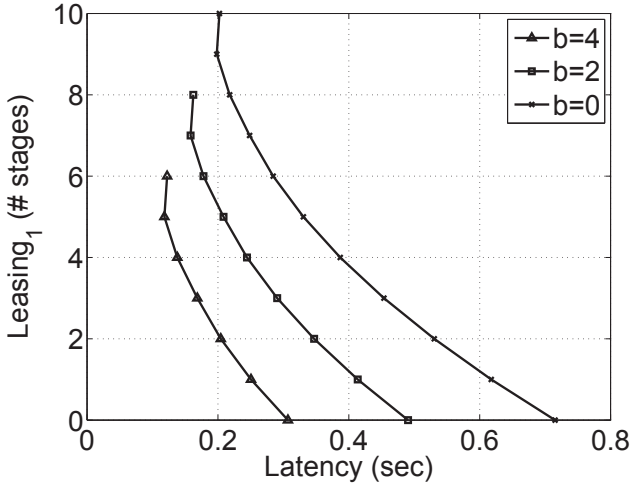


Fig. 12. Concave costs: Leasing of 1st mid-network node vs. Latency for different number of stages (b) processed on the battery limited Mobile Station, i.e. $b = 0$ means no stages are processed at the MS.

use the new feature while generating revenue.

6 Conclusion

The popularity of mobile applications is steadily increasing. Many of these applications require significant computation power, especially in the case of multimedia applications. As the demand, as well as the sophistication and required computation power, for these types of applications increases, battery and communication bandwidth limitations may prevent the use of many of these applications. By "leasing" processing power from mid-network nodes, the battery drain and communication latency may be diminished. Network-Assisted Mobile Computing can help alleviate the processing burden off the Mobile Station without increasing the service latency. Using Dynamic Programming, we identified the optimal processing

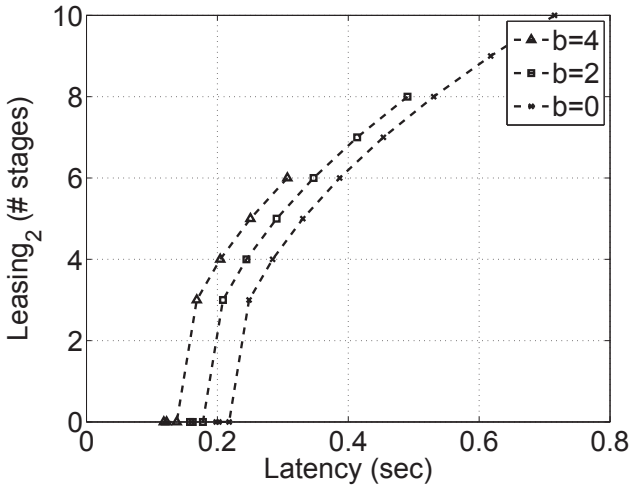


Fig. 13. Concave costs: Leasing of 2nd mid-network node vs. Latency for different number of stages (b) processed on the battery limited Mobile Station, i.e. $b = 0$ means no stages are processed at the MS.

policy. We identified some important properties of the optimal policy which can be used to guide future system design. Through numerical studies we examine the core tradeoffs and relationships between battery usage, latency, and leasing costs.

A number of factors must be considered for deployment of such a network-assisted mobile computing system. While there exist technology for collaborative networks, one must consider the amount of processing and data that will be permitted to be shared at mid-network nodes. If high security is required, there may be additional costs required to handle mid-network processing. The design challenges will be application and system dependent. For instance, if the processing only requires transcoding, this can be done on fully encrypted data by simply dropping packets, making mid-network processing simple and secure [48], [49]. However, it is certainly the case that query partitioning will be limited if the data must remain encrypted during the whole query processing. Much as transcoding encrypted media has been an interesting area of research, one may want to consider developing processes which allow for other query processing on encrypted data.

References

[1] T. Yeh, K. Tollmar, and T. Darrell, "Searching the web with mobile images for location recognition," in *Proc. IEEE CVPR*, vol. 2, pp. 76–81, July 2004.
[2] G. Fritz, C. Seifert, and L. Paletta, "A mobile vision system for urban detection with informative local descriptors," in *Proc. IEEE ICVS*, p. 30, Jan. 2006.
[3] H. Bay, B. Fasel, and L. V. Gool, "Interactive museum guide: Fast and robust recognition of museum objects," in *Proc. IMV*, May 2006.
[4] "Snaptell part of A9." http://www.snaptell.com/.
[5] "Amazon app for iphone and ipod touch." http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000291661.
[6] "omoby." http://www.omoby.com/.
[7] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, J. P. Singh, and B. Girod, "Robust image retrieval using scalable vocabulary trees," in *Proc. VCIP*, 2009.
[8] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, J. P. Singh, and B. Girod, "Tree histogram coding for mobile image matching," in *Proc. IEEE DCC*, pp. 143–152, 2009.

[9] V. Chandrasekhar, G. Takacs, D. Chen, J. P. Singh, and B. Girod, "Transform coding of image feature descriptors," in *Proc. VCIP*, 2009.

[10] S. Tsai, D. Chen, J. P. Singh, and B. Girod, "Image-based retrieval with a camera-phone," in *Technical Demo, IEEE ICASSP*, 2009.

[11] "At&t faces 5,000 percent surge in traffic." http://www.internetnews.com/mobility/article.php/3843001.

[12] "T-mobile's growth focusing on 3g." http://connectedplanetonline.com/wireless/news/t-mobile-3g-growth-0130/.

[13] H. Galeana-Zapien and R. Ferrus, "Design and evaluation of a backhaul-aware base station assignment algorithm for ofdma-based cellular networks," *IEEE Trans. Wireless Commun.*, vol. 9, pp. 3226–3237, 2010.

[14] T. Biermann, L. Scalia, C. Choi, H. Karl, and W. Kellerer, "Backhaul network pre-clustering in cooperative cellular mobile access networks," in *WoWMoM*, pp. 1–9, June 2011.

[15] D. Wetherall, U. Legedza, and J. Guttag, "Introducing new internet services: Why and how," *IEEE Network*, vol. 12, pp. 12–19, 1998.

[16] U. Legedza, D. Wetherall, and J. Guttag, "Improving the performance of distributed applications using active networks," in *Proc. IEEE INFOCOM*, 1998.

[17] D. L. Tennenhouse and J. M. Smith, "A survey of active network research," *IEEE Communications Magazine*, vol. 35, pp. 80–86, 1997.

[18] S. Merugu, S. Bhattacharjee, Y. Chae, M. Sanders, K. Calvert, and E. Zegura, "Bowman and canes: Implmentation of an active network," in *37th Annual Allerton Conference*, 1999.

[19] S. Schmid, T. Chart, M. Sifalakis, and A. Scott, "Flexible, dynamic, and scalable service composition for active routers," in *Proc. IWAN*, pp. 253–266, 2002.

[20] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, "An intelligent task allocation scheme for multi-hop wireless networks," *IEEE Transactions On Parallel And Distributed Systems*, vol. 99, no. PrePrints, 2011.

[21] Y. Tian and E. Ekici, "Cross-layer collaborative in-network processing in multi-hop wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 297–310, 2007.

[22] T. Xie and X. Qin, "An energy-delay tunable task allocation strategy for collaborative applications in networked embedded systems," *IEEE Transactions on Computers*, vol. 57, no. 3, pp. 329 –343, 2008.

[23] A. Olsen, F. Fitzek, and P. Koch, "Energy aware computing in cooperative wireless networks," in *International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, pp. 16 – 21, 2005.

[24] J. Li, M. Qiu, J.-W. Niu, and T. Chen, "Battery-aware task scheduling in distributed mobile systems with lifetime constraint," in *Proc. ASP-DAC*, pp. 743 –748, 2011.

[25] "Akamai." http://www.akamai.com.

[26] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Proc. CoNEXT 2009*, pp. 1–12, Dec. 2009.

[27] S. Gitzenis and N. Bambos, "Power-controlled data prefetching/caching in wireless packet networks," in *Proc. IEEE Infocom*, vol. 3, pp. 1405–1414, 2002.

[28] S. Gitzenis and N. Bambos, "Joint transmitter power control and mobile cache management in wireless computing," *IEEE Trans. Mobile Comput.*, vol. 7, pp. 498–512, Apr. 2008.

[29] S. Drew and B. Liang, "Mobility-aware web prefetching over heterogeneous wireless networks," in *Proc. IEEE PIMRC*, pp. 687– 691, 2004.

[30] C. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

[31] M. Weiser, B. Welch, A. J. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in *Operating Systems Design and Implementation*, pp. 13–23, 1994.

[32] K. Govil, E. Chan, , and H. Wasserman, "Comparing algorithm for dynamic speed-setting of a low-power cpu," in *Proc. of MOBICOM*, p. 1325, 1995.

[33] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. D. Michelli, "Dynamic voltage scaling and power management for portable systems," in *Proc. ACM Conference on Design Automation*, p. 524529, 2001.

[34] P. Pilai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proc. ACM symposium on operating systems principles*, p. 89102, 2001.

[35] H. Mehta, R. Owens, M. Irwin, R. Chen, and D. Ghosh, "Techniques in low energy software," in *Proc. ACM international symposium on Low power electronics and design*, p. 7275, 1997.

[36] E.-Y. Chung, L. Benini, and G. D. Michelli, "Source code transformation based on software cost analysis," in *Proc. ACM international symposium on System synthesis*, vol. 4, p. 153158, 2001.

[37] A. Rudenko, P. Reiher, G. Popek, and G. Kuenning, "Saving portable computer battery power through remote process execution," *Mobile Computing and Communications Review*, vol. 2, pp. 19–26, Jan. 1998.

[38] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning, "The remote processing framework for portable computer power saving," in *Proc. ACM Symposium on Applied Computing*, p. 365372, 1999.

[39] S. Mohapatra and N. Venkatasubramanian, "Para: power aware reconfigurable middleware," in *Proc. IEEE Int. Conf. on Distributed Computing Systems*, p. 312319, 2003.

[40] S. Gitzenis and N. Bambos, "Joint task migration and power management in wireless computing," *IEEE Trans. Mobile Comput.*, vol. 8, pp. 1189–1204, Sept. 2009.

[41] S. Narayanaswamy, S. Seshan, E. Amir, E. Brewer, R. Brodersen, F. Burghart, A. Burstein, Y. Chang, A. Fox, J. Gilbert, R. Han, R. Katz, A. L. D. Messerschmitt, and J. Rabaey, "A low-power, lightweight unit to provide ubiquitous information access applications and network support for infopad," *IEEE Personal Commun. Mag.*, vol. 3, pp. 4–17, Apr. 1996.

[42] S. Gitzenis and N. Bambos, "Mobile to base task migration in wireless computing," in *Proc. IEEE PerCom*, pp. 187– 196, Mar. 2004.

[43] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham, "Mobile visual search," *Signal Processing Magazine, IEEE*, vol. 28, no. 4, pp. 61 – 76, 2011.

[44] C. W. Chan, N. Bambos, and J. Singh, "Wireless network-assisted computing," in *Proc. IEEE PIMRC*, pp. 1–5, Sept. 2008.

[45] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, pp. 390–404, Apr. 2006.

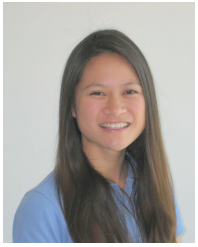
[46] *15444 JPEG-2000 Image Coding System, Part 1: Core coding system*.

[47] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, *AHG Report on Spatial Scalability Resampling, Document JVT-Q007*, Oct. 2005.

[48] S. Wee and J. Apostolopoulos, "Secure scalable streaming and secure transcoding with JPEG-2000," in *Proc. IEEE ICIP*, vol. 1, pp. 205–208, Sept. 2003.

[49] J. Apostolopoulos, "Secure media streaming & secure adaptation for non-scalable video," in *Proc. IEEE ICIP*, vol. 3, (Singapore), pp. 1763–1766, Oct. 2004.

[50] D. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1 & 2. Athena Scientific, 2nd ed., 2000.



Carri W. Chan is an Assistant Professor at the Graduate School of Business of Columbia University where she has been since 2010. She received her SB degree in Electrical Engineering & Computer Science from the Massachusetts Institute of Technology (2004). She received her M.S. (2006) and Ph.D. (2010) degrees in Electrical Engineering from Stanford University. She is a member of the IEEE and INFORMS. She is a recipient of a STMicroelectronics Stanford Graduate Fellowship (2004). Her research interests include modeling of complex stochastic systems, efficient algorithmic design for queueing systems, and dynamic control of stochastic processing systems. Applications for this line of research include telecommunication networks, healthcare operations management, and information services.

Nicholas Bambos received his Ph.D. in electrical engineering and computer science from U.C. Berkeley in 1989, after graduating in Electrical Engineering from the National Technical University of Athens, Greece in 1984. He served on the Electrical Engineering faculty of UCLA from 1990 to 1995 and joined Stanford University in 1996, where he is now a professor in the Electrical Engineering department and the Management Science & Engineering department. His current research interests are in performance engineering of communication networks and computing systems, including queuing and scheduling issues in wireless an wireline networks, as well as ergodic random processes, queuing theory and adaptive control of stochastic processing networks.

Jatinder Pal Singh is the Director of Mobile Innovation Strategy at Palo Alto Research Center and Consulting Associate Professor with the department of Electrical Engineering at Stanford University. He was previously Vice President of Research with Deutsche Telekom, one of the world's largest ISP and parent company of T-Mobile. He received his Ph.D. and M.S. in Electrical Engineering from Stanford University, where he was awarded Stanford Graduate Fellowship and Deutsche Telekom Fellowship. He received his B.S. in Electrical Engineering from the Indian Institute of Technology, Delhi, where he graduated at the top of his class with Institute Silver Medal.