International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Topology Control in Mobile Computing with Optimal Uplink Query Processing

Syeda Husna Mehanoor[1], Syeda Ayesha Thainiath[2]

[1]DR V.R.K College of Engineering & Technology (Affiliated to JNTUH)
[2]Nawab Shah College of Engineering & Technology (Affiliated to JNTUH)

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 trade-off 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.***

Keywords: Mid network, Transmission latency, Communication latency, Processing cost, optimal processing policy

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 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. 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, 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. 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. 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.

2. Literature Survey

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 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, 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 and software, 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, these works examine load splitting where processing is split between Mobile Station and Application Server. The potential battery savings by splitting processing between Mobile Station and Application Server are examined experimentally. The trade-off between battery usage and latency is closely examined. Girod et. al. provide an overview of these types of challenges in mobile visual search . 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 trade off 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 1-2MB1. 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 midnetwork nodes in order to improve the battery versus latency trade-off. Rather than relying solely on the Mobile Station and Application Server to process a request, we allow for midnetwork processing. In this work, an extension, we introduce the idea of “leasing” processing power from midnetwork nodes in order to improve quality of service to users.

3. Problem Definition

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.

4. Methodologies

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. In this paper, we focus on the uplink transmission and processing of a single original request.

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. The mobile station send one sentence for ex: (how are you), in the application server receive the sentence into audio.

Relaying Strategies Model:

- Amplify-and-Forward
- Decode-and-Forward

In amplify-and-forward, the relay nodes simply boost the energy of the signal received from the sender and retransmit it to the receiver. In decode-and-forward, the relay nodes will perform physical-layer decoding and then forward the decoding result to the destinations. If multiple nodes are available for cooperation, their antennas can employ a space-time code in transmitting the relay signals. It is shown that cooperation at the physical layer can achieve full levels of diversity similar to a system, and hence can reduce the interference and increase the connectivity of wireless networks.

Multi-Hop Transmission Model

Multi-hop transmission can be illustrated using two-hop transmission. When two-hop transmission is used, two time slots are consumed. In the first slot, messages are transmitted from the mobile station to the relay, and the messages will be forwarded to the Application Server in the second slot. The

outage capacity of this two-hop transmission can be derived considering the outage of each hop transmission.

5. Technique Used

Managed Code

The code that targets .NET, and which contains certain extra Information - "metadata" - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

Common Type System

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

The Class Library

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary. The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity. The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

6. Conclusion and Future Work

In this paper 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 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. 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] J. Laneman, D. Tse, and G. Wornell, "Cooperative Diversity in Wireless Networks: Efficient protocols and Outage Behavior," *IEEE Trans. Info. Theory*, vol. 50, no. 12, 2004, pp. 3062–80.
[2] P. H. J. Chong *et al.*, "Technologies in Multihop Cellular Network," *IEEE Commun. Mag.*, vol. 45, Sept. 2007, pp. 64–65.
[3] K. Woradit *et al.*, "Outage Behavior of Selective relaying Schemes," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, 2009, pp. 3890–95.
[4] Y. Wei, F. R. Yu, and M. Song, "Distributed Optimal Relay Selection in Wireless Cooperative Networks with Finite-State Markov Channels," *IEEE Trans. Vehic. Tech.*, vol. 59, June 2010, pp. 2149–58.
[5] Q. Guan *et al.*, "Capacity-Optimized Topology Control for MANETs with Cooperative Communications," *IEEE Trans. Wireless Commun.*, vol. 10, July 2011, pp. 2162–70.

Author Profile

Syeda Husna Mehanoor received the M. Tech degree in computer Science Engineering from DR. V.R.K College of engineering.

Syeda Ayesha Thainiath received the M. Tech degree in computer Science Engineering from Al-Habeeb College of Engineering and Technology. Working in Nawab Shah College of Engineering & Technology as an Assistant Professor in CSE dept.