Chapter 1

INFOSTATIONS: NEW PERSPECTIVES ON WIRELESS DATA NETWORKS

Ana Lúcia Iacono
Christopher Rose
WINLAB – Wireless Information Network Laboratory
Department of Electrical and Computer Engineering
Rutgers, The State University of New Jersey

Abstract

We discuss file delivery issues for a new approach to inexpensive, high rate wireless data called *Infostations*. As opposed to ubiquitous coverage, infostations offer geographically intermittent coverage at high speed (1Mbps to 1Gbps) since data, as compared to voice, can often tolerate significant delay. The infostations paradigm flips the usual "slow-radio/fast-network" scenario upside down and offers intriguing new design problems for wireless data networks. Collectively, we at WINLAB believe that the infostations scenario, especially with the emergence of the World Wide Web as both a communications medium and defacto standard is one way to obtain low cost wireless data. And perhaps controversially, we offer arguments that currently proposed extensions to cellular systems (such as the coming Third Generation) will not be able to offer data as inexpensively. In this chapter we describe the infostations concept and then concentrate on issues above the physical layer. Specifically, we worry about delay bounds on information delivery for variety of simple user mobility scenarios and infostation geometries. We then provide heuristic algorithms which closely approach these bounds.

Keywords: wireless communications, mobile computing, wireless data, wireless internet, scheduling algorithms, delay bounds, mobility management

1. INTRODUCTION

Over the past 10 years, wireless voice communication has grown from a rarity to a necessity. In contrast, wireless data services at rates and price sufficient to generate equal excitement remain elusive. In response, the wireless industry has proposed third generation systems with rates in the hundreds of kilobit per second range. However, the dominant traffic on such systems will probably be voice at least initially, and here lies a "Catch-22" first observed by Roy Yates here at WINLAB [1].

Consider that the bit rate currently associated with voice communications is on the order of 10 kbps and let us use this voice channel rate as our unit of measure. This channel costs v cents per minute. It therefore costs approximately $13v$ cents to transmit a one megabyte file – prohibitively expensive at current rates. In addition, what is particularly interesting is that this basic fact does not change with the introduction of higher rate services as long as voice is the dominant traffic. One megabyte of data *always* costs $13v$ cents since the basic voice channel rate is unlikely to change drastically for both economic and legacy reasons. Thus, unless normal voice communications becomes essentially free, it seems that wireless data will never be inexpensive when provided using a cellular architecture.

This conundrum causes us to re-examine the cellular paradigm. Specifically, cellular wireless was built to carry voice traffic for people accustomed to the reliability and ubiquity of fixed telephone service. Thus, the goal of the cellular industry was coverage anytime and anywhere. However, to provide large coverage the system must be designed so that users both near and far from the access point (a base station) achieve some minimum quality of service. From a systems perspective however, it would be more efficient to serve users closer to the base station at higher rate, be done with them and then serve users farther away. However, for voice systems the implication is intermittent coverage which is incompatible with continuous interactive traffic such as voice.

In contrast, data can tolerate delay and the system throughput could be increased by offering rates commensurate with achievable signal to interference ratios. Add to this that customers are often in motion the basic (somewhat surprising) infostations design precept emerges for single non-dispersive, non-directional channels:

Infostations should not be shared between users

That is, at any point in time, only one user should be attached to an infostation. This basic idea has roots in information theory and water-filling of channels in space, time and frequency (see [2] for a development on multiple user dispersive channels). If we consider different frequency or spatial sub-channels, then the precept still holds if each sub-channel is considered to be an infostation unto

itself and users attempt to use the "infostation(s)" with the best channel(s), even though these infostations might be co-located.

A possibly non-obvious consequence of such spatio-temporal-frequency water-filling results in another defining characteristic of the infostations paradigm. For users in (ergodic) motion, the places at which transmissions should occur are where the channel quality is above some threshold – a result first shown by Joan Borras [3, 4] and based in part on work by Andrea Goldsmith [5] on fading channels. This implies that a user traveling with uniform velocity in an isotropic environment should transmit or receive only when it is close to an infostation, and from this the notion that

Infostation coverage areas are spatially discontinuous

emerges naturally.

Thus, we define infostations as a wireless communication system characterized by sequential user access with discontinuous coverage areas and high data rate transmissions. As opposed to the moderate rate ubiquitous coverage in cellular systems, infostations offer high speed discontinuous coverage which may be accessed by users in transiently close proximity to an infostation, and in fact can maximize system capacity. Furthermore, the removal of the need to coordinate channels among multiple users and over the system as a whole should lead to simple inexpensive realizations. And owing to the bursty nature of data communications and its tolerance of moderate delay, the infostation scenario with its inherently lower associated costs might be an attractive alternative to the classical concept of anytime anywhere communications networks.

1.1 EXAMPLES

Although not specifically an infostation, consider a system introduced by Apple, called Airport [6]: a base station that costs $299 and wireless networking cards that cost $99 enable up to 10 computers to share a 11 Mbits/second Internet connection at distances up to 150 feet. As Peter Lewis describes in his article in The New York Times [7]:

"*That is so important, and it has such potential to change the way we use computers and information appliances around the house, that I'm compelled to repeat it in a different way: I'm sitting outside the house on the deck, with an iBook on my lap, enjoying a glorious autumn day, reading the current e-news, checking e-mail . . . There are no wires, cables or extension cords in sight. As the stars come out, I simply stroll back into the house and continue working from the sofa in the living room.*"

The salient feature of this narrative is a perceived desire for manytime many-where web access as opposed to the more traditional anytime anywhere access

we expect for voice services. Note in particular that Lewis did not suggest he was *using* the computer during his journey from deck to sofa.

There are a variety of possible infostation system architectures. For example, many infostations may be owned by a single company and they may be clustered and connected to cluster controllers according to their location, creating a hierarchical architecture, as shown in Figure 1.1. This is somewhat analogous to the large telephone company cellular systems where many base stations are connected to mobile telephone switching offices through dedicated high speed lines.

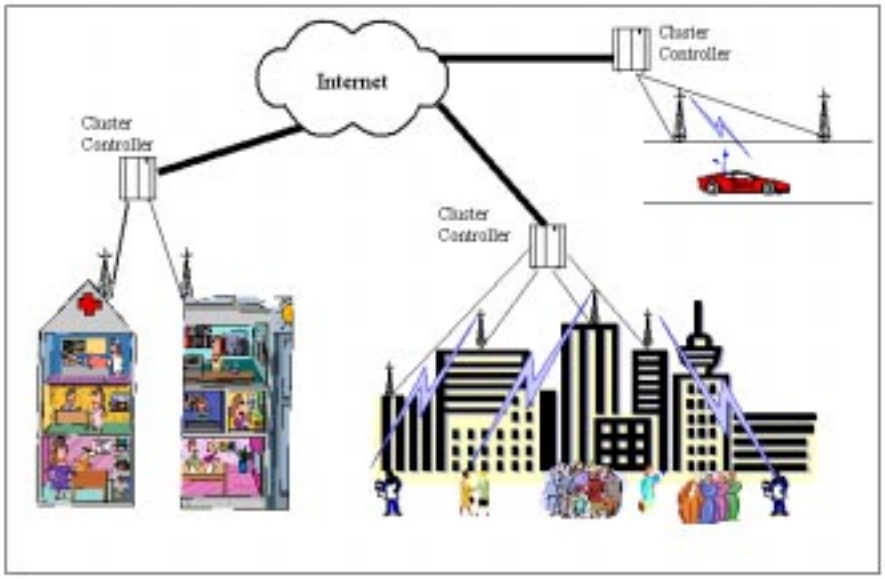

Figure 1.1 Cluster of Infostations

Another possible scenario might have small businesses such as convenience stores carry infostation service as a sideline – analogous to lottery sales agents. This architecture is shown in Figure 1.2. To be economically attractive, the start up cost to such a "Mom and Pop" operator should be low. There could also be a mixture of the two, as in a franchise setting where infostation operators leased the infrastructure from the founding company.

The network could also be isolated from the Internet and could be used for local communications, as in an office building or home. This is shown in Fig-

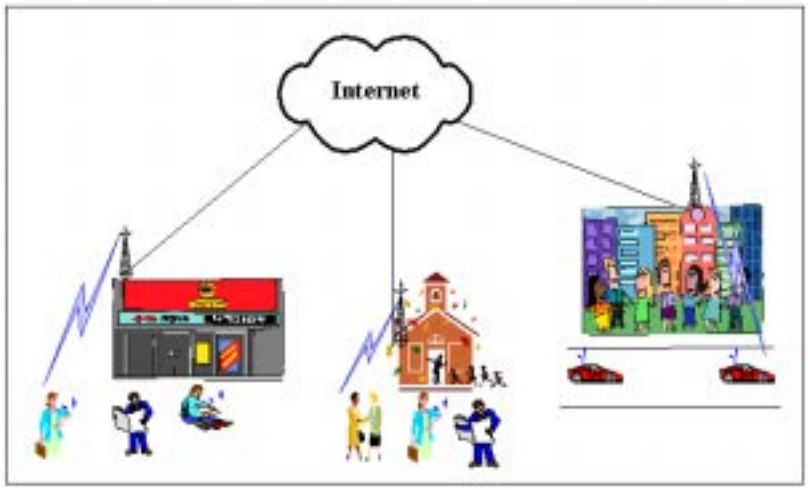

Figure 1.2 Independent Infostations

ure 1.3. Yet another architecture is to integrate infostations with a ubiquitous, low data-rate system (e.g. CDPD or other [8]) and use them as bandwidth boosters. Figure 1.4 shows one example of a hybrid infostation network. According to where infostations are placed, the user mobility can be characterized by three situations [9]: mobile users moving with high speed, such as in a highway, characterize what is called a "drive-through" scenario; users with medium speed, such as in a sidewalk or a mall, characterize a "walk-through" scenario; finally stationary users, such as in an airport lounge or a classroom, characterize "sit-through" scenario.

At WINLAB we have been studying several different problems related to an infostation network. A study of the infostation system performance in terms of capacity, throughput and delay was presented in [4] where various models and different power allocation, symbol rate adaptation and modulation schemes are presented. A medium access scheme called WINMAC has been proposed in [10] to support efficient packet communications between an infostation and mobile terminals. This protocol adapts to the radio channel condition and achieves enhanced communication reliability through packet retransmission and data rate adjustment. Due to the fact that one of the main services that infostation will provide is Internet access, another area of interest is the design of a link layer protocol to transmit IP packets efficiently via the wireless link. An error control scheme for the Radio Link Protocol is proposed in [11]. The

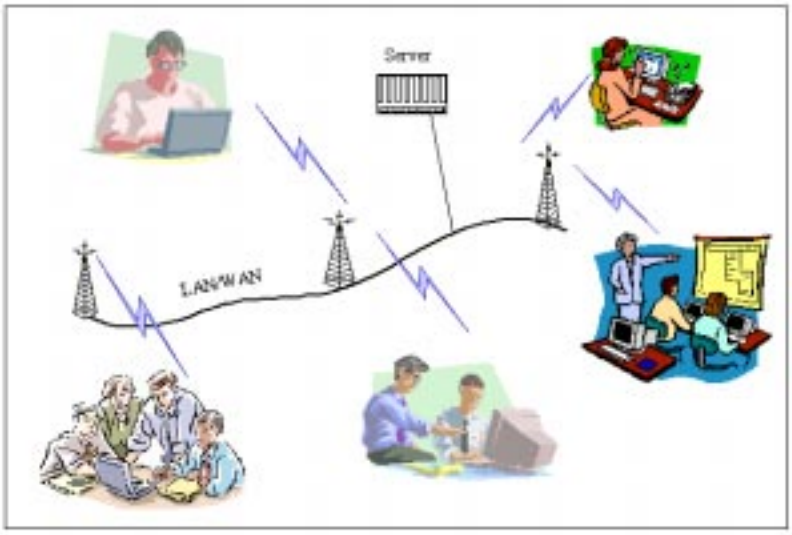

Figure 1.3 Isolated Infostation System

scheme uses multicopy and error threshold detection to improve the system performance. Infostation operation issues such as registration, authentication and billing are addressed in [12]. Some radio design issues are examined in [13]. There is also a variety of other work both at WINLAB and elsewhere ranging from physical layer issues up through applications [3, 12, 14, 15, 16, 4, 17, 18, 19, 1, 8].

1.2 USER MOBILITY AND INFOSTATIONS

One might wish to place an infostations system in an airport lounge, in a conference room or in a small office at an affordable price. One common characteristic of these situations is relatively low user mobility. Although the coverage area is small, the system is designed based on the fact that the user will stay in the coverage area during the time of the connection and in fact, using beam steering techniques one might "move the infostation" as opposed to moving the user. However, from the perspective of the fixed network, users are in relatively fixed locations.

However, when designing system where users roam, then user mobility must be considered since users may visit several infostations during a single connection. For example, a user might roam over a shopping mall with stores offering

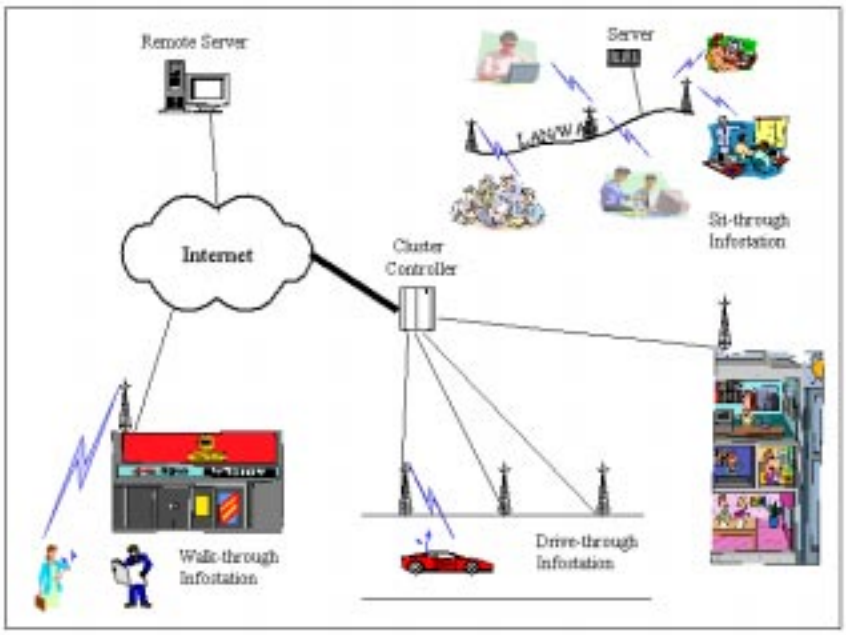

Figure 1.4 The Infostation Network.

local infostation services and a user would not stay connected to a single infostation while shopping. Likewise on a highway with infostations at regular intervals users might traverse great distances (from the fixed network perspective) between infostation contacts.

Now consider that data communication, such as messaging systems or web applications, is inherently asymmetric with much greater volume occurring on the downlink from network to user. Under this scenario, if the information is available at the infostation, then the main issue is to send it to the mobile as rapidly as possible. Since the data rate is high, as long as the user is in the coverage area, this can be done in a few seconds. However, if the information is not available at the infostation and has to be transfered from a server, then the information has to pass through the fixed network before reaching the mobile. Thus, in the "drive-through" scenario, since the coverage area is small, the time which the mobile spends in coverage at a given infostation may not be sufficient to transfer the information from the server to the infostation.

This is a situation peculiar to infostations where the radio rate is assumed much higher than the fixed network rate. Given an inexpensive high-speed

radio, there are a number possible reasons for this inversion of the status quo. For economy, one could have a low cost relatively low rate connection (i.e. commodity telephone modems) to each infostation. Alternately, even if one connects the infostations with high speed links, some types of services (i.e. HTTP request) have typical transmission rates of the order of Kbits/second. The server transmission rate and network congestion play an important role in determining the speed of the connection. Another scenario would have fixed network links servicing some primary traffic with the infostation as an add-on service sharing these links. Regardless, in all these cases although the radio rate is high, the user would have to restart a request at the next infostation in the path, and this process will increase the delivery delay, especially if only a small fraction of time is spent in coverage by any one user.

1.3 PROBLEM OVERVIEW, MOTIVATION

The obvious solution to this radio/fixed-net mismatch is to cache or prefetch information at the infostations. As an example, an intelligent prefetching algorithm which attempts to predict *what* the user will need was proposed in [14] as a solution to a location dependent application (map request). The algorithm uses location and speed information to select which of a set of maps should be prefetched. Based on location, time or user dependency, different types of applications would need different schemes for prefetching. However, suppose the information needed is *known* and can be of any sort such as a web page, a map, or personal e-mail. Then, the issue becomes how to partition the information, and then *when* and *where* to send the packets over the fixed network so that they arrive at the user with minimal overall delay.

Thus, consider a system where the infostations are connected as a *cluster* in a hierarchy where there is a higher level with a *cluster controller*, as shown in Figure 1.5. The cluster controller is the entity that has information on all requests that were made in that cluster and how many users are being served at every infostation in that cluster. Note that a cluster would be a natural way of connecting different infostations in the same geographical area, but the cluster controller does not have to be necessarily in the same geographical area.

The cluster controller can coordinate the delivery of some packets to the next infostation in the mobile path, so that they are locally available at that infostation when the mobile user arrives in its coverage area. If the path is not known then the cluster controller can send the packets to infostations that are most likely to be in the mobile user path. Therefore, during the time the user is going between two infostations, the system can download the information to the next coverage area, reducing the delivery delay, as shown in Figure 1.6.

The optimization problem is then, given some parameters and system configuration, to deliver the information from its current location(s) to the mobile

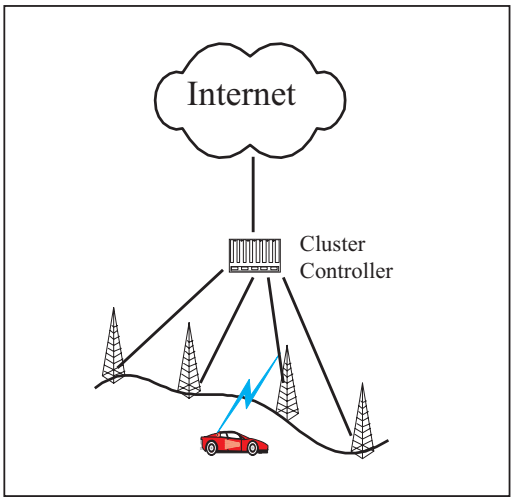

Figure 1.5 The Drive-through Scenario.

user in a minimum amount of time. The important parameters that have to be taken into account are:

- the overall amount of information that is requested, or file size;
- the location of the file, which can be stored at the infostation, at the cluster controller, at the home server (Internet) or distributed over a number of locations;
- the data rate of the wired and wireless network;
- the number of infostations at the cluster;
- the infostations' location;
- the user mobility model.

To better understand the approach used here, consider the single user case where there is a cluster with a given number of infostations, M. Let R_i be the rate between the cluster controller and the Internet, R_b the rate of each link between the cluster controller and the infostations and R_r the radio rate. Assume that the user requests a file, which is then divided into packets, and each packet can be sent to different infostations.

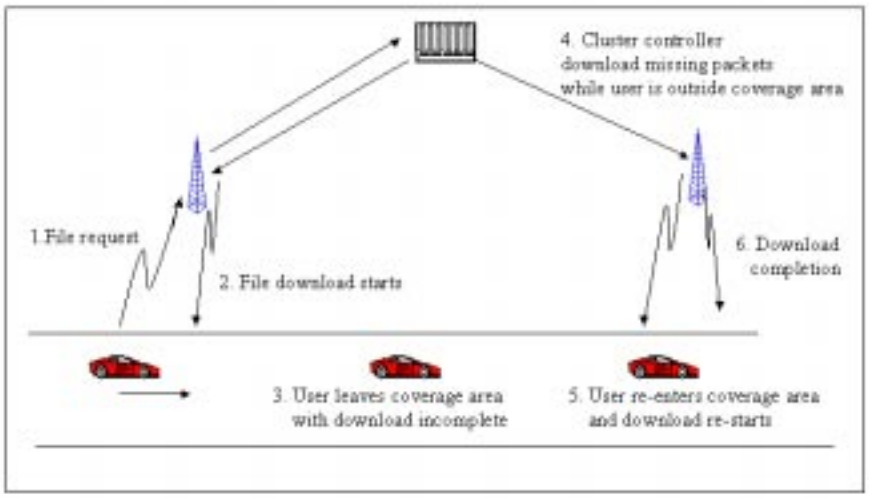

Figure 1.6 The Problem Approach.

Note that if the file is stored in "The Internet" then the network will be able to download the file to the user at the lowest link speed of the network. To take advantage of the fast radio, the cluster controller will prefetch file packets to infostations in the user path. Note that if radio data rate (R_r) were low, then every request should be re-initiated at every infostation. That is, with a slow radio, there is little use in prefetching information to the infostations. Therefore we are interested in the case where $R_r > R_b$ and $R_r > R_i$. In this case the prefetching approach is helpful and only the radio rate will restrict the *maximum* amount of information that should be prefetched at a given infostation since there is a limit on how much can be downloaded to the user in the coverage area.

Given that the radio data rate is large, the specific delivery problem ranges from the trivial to the difficult. Consider the case when information is stored at some server on the Internet. Since we assume $R_r \gg R_i, R_b$, then the fixed network is the limiting factor. In the case when $R_i \leq R_b$ then the cluster controller can broadcast *all* the packets received to *all* the infostations, as shown in Figure 1.7. All the infostations will have the same information that the cluster controller has, as a copy network. As the user passes through the infostations the radio can then download as many packets as possible to the user and discard packets that were already received.

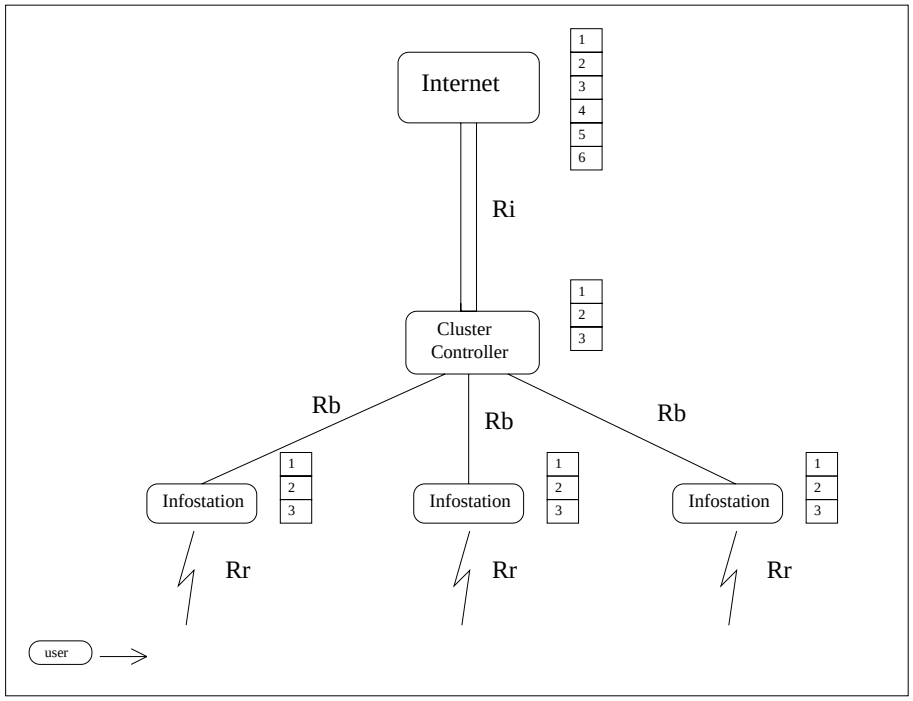

Figure 1.7 An example of a copy network.

However if $R_i > R_b$, then although the cluster controller cannot copy all the packets to all infostations, it can send different packets to different infostations, as shown in Figure 1.8. As the user passes through new infostations, it can get *new* packets.

In the same way, if the requested file is locally stored at the cluster controller, then again different file packets can be divided among all infostations in the path, taking advantage of the parallelism of the network. The cluster controller has to decide which packets to put in which infostations so that, when the user passes through the infostations, it obtains the most amount of information possible. The number of packets that can and should be prefetched is a function of the backbone rate (R_b) and the radio rate (R_r). Note that if the Internet rate (R_i) is very high then the scenario is similar and we can assume that the file is locally stored at the cluster controller, as shown in Figure 1.9

In general, the cluster controller has a buffer where it queues all the file packets. According to how large R_i is, the size of the largest buffer (if the file is locally stored at the cluster controller then the buffer contains all the file packets). Thus, the cluster controller can coordinate the delivery of these packets to different places and it can send *copies* and/or *different* packets to

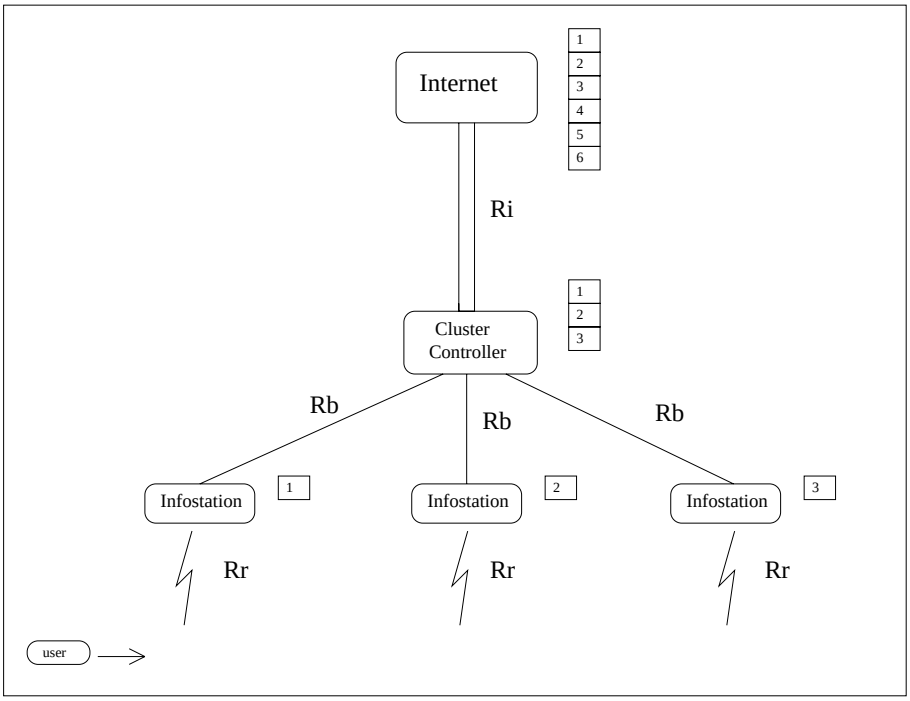

Figure 1.8 The parallelism of a network.

every infostation in the cluster. According to the radio rate and time spent in the coverage area, each infostation should only store some maximum amount of information, since the user will not be able to transfer all the packets during its brief visit to the coverage area. This more general case is shown in Figure 1.10

Let us assume that there are N infostations in the user path before completion of file delivery. Note that the value of N will be a function of the file size, the delivery algorithm used, the user mobility model and the rates R_b, R_i and R_r. In any case, we call this number N, although the value could be different in each situation.

Assume that $M\,R_b$ bits/second is the maximum data rate necessary so that the radio is able to download all packets that are prefetched to a given infostation. Note that the value of M will be a function of the total file size, the delivery algorithm used and the user mobility model.

For a given value of R_b, the problem space diagram is shown in Figure 1.11. Region (1) is the case where $R_i < R_b$ and the cluster controller should just broadcast every information received to all infostations. Region (2) is where the radio is the bottleneck of the network and there is no reason for caching any

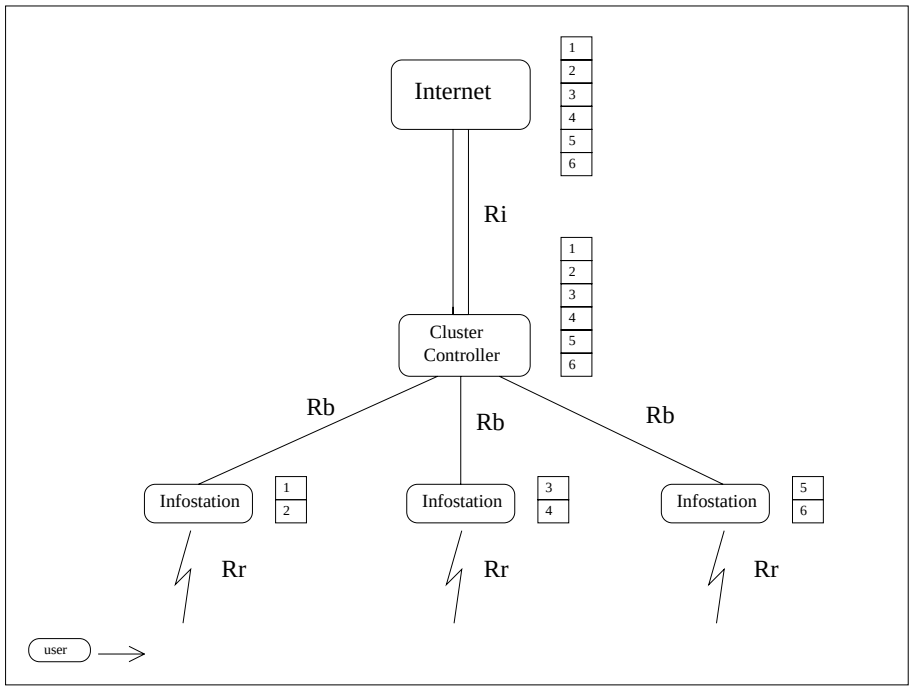

Figure 1.9 The cluster controller has all the packets and distributes them through the infostations.

information since it will not be delivered to the user anyway. Therefore our study lies in region (3).

In region $(3a)$ the rates R_i and R_r are as large as necessary to take the most possible advantage of parallelism of the network for a given file size. That implies that either the file is stored at the cluster controller or $R_i \geq NR_b$. It also implies that the radio rate is very fast and *all* packets prefetched to the infostations can be downloaded to the user during the time it is in the coverage area.

If the rate R_i is not as high then it is not possible to bring the the desired amount of information to the cluster controller in order to spread it over the many slow links. The cluster controller can then send different packets to some infostations and copy others in more than one place. Regions $(3b)$ and $(3d)$ represents this situation. In other words, the queue at the cluster controller will have a small number of packets and they can be copied to some infostations. In regions $(3c)$ and $(3d)$, the radio imposes a limit on the number of packets that should be prefetched to the infostations since only some maximum number of

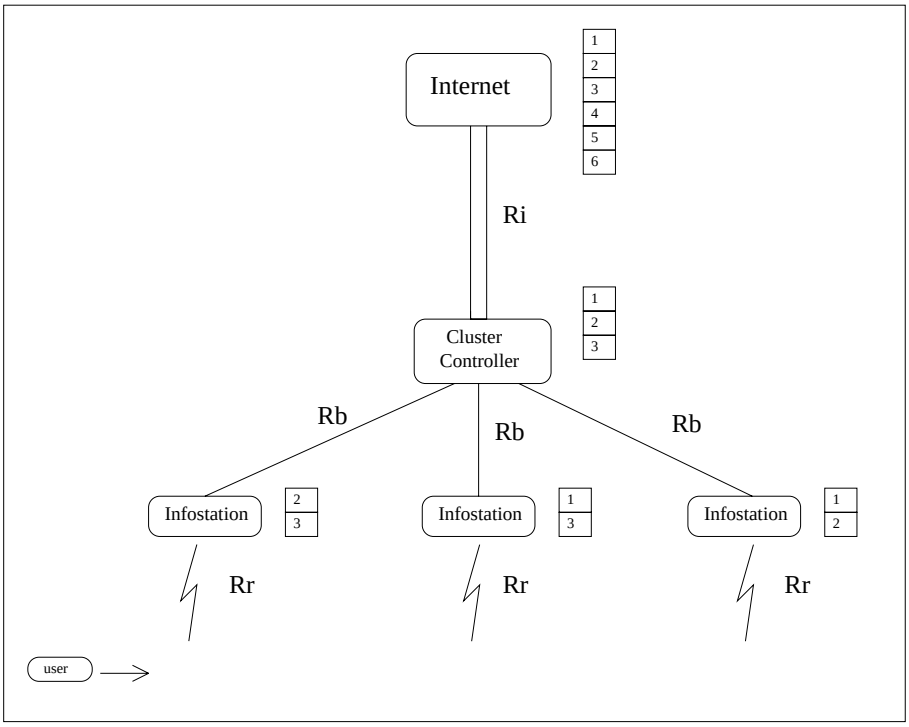

Figure 1.10 The cluster controller coordinates the delivery of packets through the cluster.

packets can be downloaded to the user during its passage through the coverage area.

Having exercised the various overall model parameters and identified a number of scenarios – some trivial, some not, we can state the file delivery problem for infostations simply. Let $\mathcal{A}$ be a set of algorithms which transmit parcels of information to each infostation for delivery to a user. Let D be the delivery delay seen by that user, defined as the total time between the initiation of the request and delivery of the final parcel. The optimization problem is then to

- Find absolute lower bounds on delivery delay

- Find algorithms which approach or meet these lower bounds

In this chapter, we will consider these problems for different mobility models and infostation structures for single users. The multiple users case is treated elsewhere [17] and will be the subject of near-future publications .

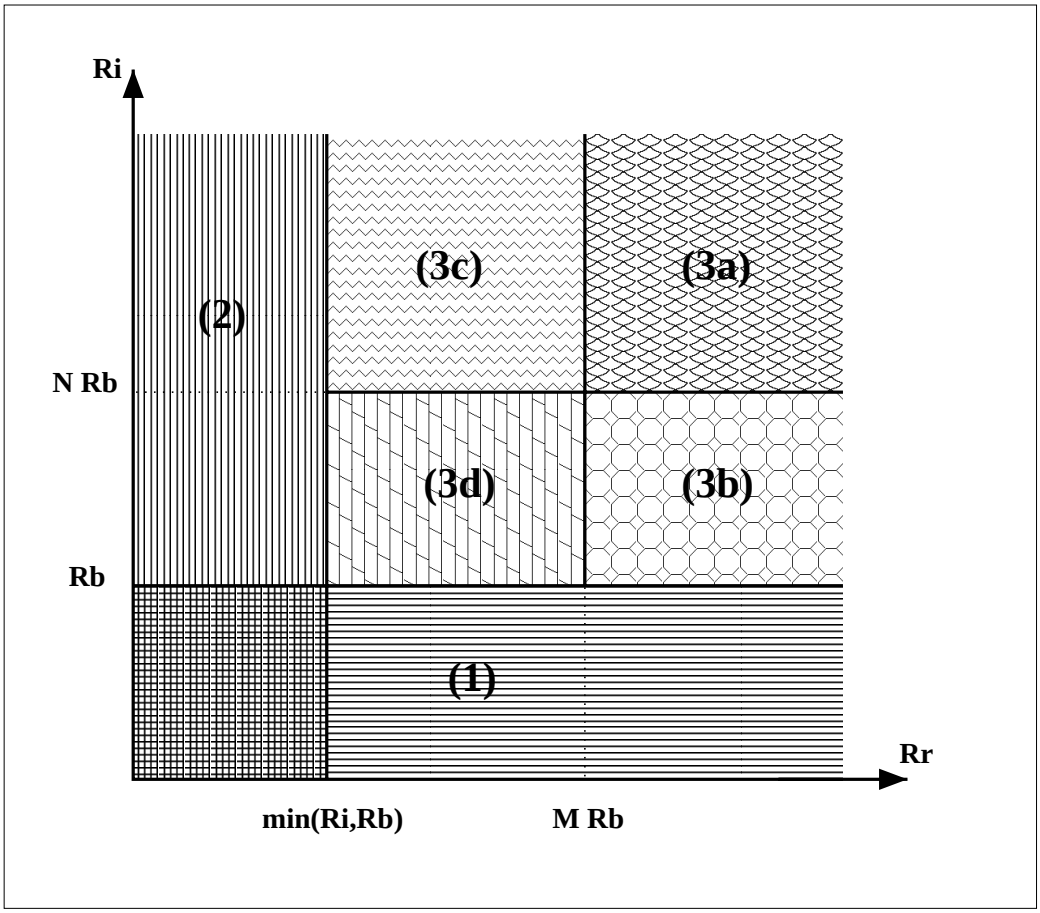

Figure 1.11 The Problem Space.

2. TOURS IN ONE DIMENSION: THE HIGHWAY SCENARIO

Consider a one-dimensional model where users move along a line populated with equidistant infostations. Our objective is to derive lower bounds on information delivery delay. In the first section we study the case of constant velocity, where users move with a fixed velocity and fixed travel direction. In the second section we study the case where users travel with constant speed but random travel direction – a one-dimensional random walk. This constant velocity model, although simple, covers important situations such as highway or railroad travel. Furthermore, it serves to illustrate some of the basic con-

cepts of file delivery under the infostation model. In this section we assume that information can be delivered from the backbone to the clusters at or above the backbone rate and that likewise, the radios are speedy enough that were an entire file available at the infostation, it could be downloaded during one passage through the coverage area.

2.1 CONSTANT VELOCITY

Consider a system with many infostations equally spaced at distance d meters, as shown in Figure 1.12. Assume that the mobile travels at constant velocity v m/s, the size of the coverage area is d_c and that the wired backbone transmits at a rate of $R_b < R_r$ bits/sec.

The mobile will arrive in a given infostation and request a file of size F bits. If

$$\frac{R_b d_c}{v} < F \tag{1.1}$$

then the mobile can receive only a part of the file at the first infostation. Completion of the transaction must be deferred until it arrives at the next infostation. Assuming a start/stop protocol where incremental requests must be initiated at each new infostation visited, the number of infostations, I, that the mobile has to pass to receive the whole file is

$$I = \left\lceil \frac{Fv}{R_b d_c} \right\rceil \tag{1.2}$$

where $\lceil x \rceil$ represents the smallest integer greater than or equal to x.

The time required for travel between two infostations is d/v and therefore the delay D in seconds, to transmit the file is

$$D = \left\lceil \frac{Fv}{R_b d_c} \right\rceil \ \frac{d}{v} \tag{1.3}$$

which we can rewrite as

$$D \leq \frac{Fd}{R_b d_c} + \frac{d}{v} \tag{1.4}$$

Note that to decrease the delay one could decrease the distance between infostations thereby increasing the infostations density and moving toward a more ubiquitous coverage area scenario.

In our approach we assume that $R_r > MR_b$, which means that all information that is available at the infostation can be downloaded to the mobile before it leaves the coverage area *regardless of the amount*. We also assume that $R_i > NR_b$, which means that the cluster controller has all the file pieces

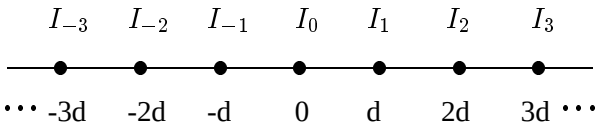

Figure 1.12 The Highway Scenario

necessary to be able to prefetch any amount of information necessary to the infostations. Note that if the information is locally stored at the cluster controller, then the value of R_i is irrelevant and the condition always holds.

In the case of constant velocity, given the initial position, the path is known. Therefore, rather than initiating new transfers at each infostation, the time the mobile is traveling between infostations may be used to download part of the file to the other infostations along the mobile tour. Since the time spent traveling between two infostations is given by d/v, the system can download B (different) bits to each infostation in the tour, where

$$B = \frac{R_b d}{v} \tag{1.5}$$

Likewise when the mobile leaves the second infostation the same amount can be downloaded to the remaining infostations and so on. Therefore, to transmit the whole file we require

$$F \leq \sum_{i=1}^{I} iB = \sum_{i=1}^{I} i\frac{R_b d}{v} = \frac{I(I+1)R_b d}{2v} \tag{1.6}$$

Thus, the smallest number of infostations, I, required for delivery of the file of size F is obtained by determining the smallest integer I such that

$$\frac{I(I+1)}{2} - \frac{Fv}{R_b d} \geq 0 \tag{1.7}$$

Therefore

$$I = \left\lceil \frac{\sqrt{1 + 8Fv/R_b d} - 1}{2} \right\rceil \tag{1.8}$$

and the delay $D_{prefetch}$, in seconds, is given by

$$D_{prefetch} = \left\lceil \frac{\sqrt{1 + 8Fv/R_b d} - 1}{2} \right\rceil \ \frac{d}{v} \tag{1.9}$$

which we rewrite as

$$D_{prefetch} \leq \sqrt{\frac{d^2}{4v^2} + \frac{2Fd}{R_b v}} + \frac{d}{2v} \tag{1.10}$$

Note that the delay now does not depend on the size of the coverage area. In Figure 1.13 we can see the delay as a function of the file size for both situations: The delay without the prefetching approach, which is given by equation 1.3 and the delay obtained when the prefetching approach is used, given by equation 1.9. We can see that there is a large improvement when the prefetching approach is used. We also note that the delay stays invariant for a larger range of file sizes when the prefetching approach is used.

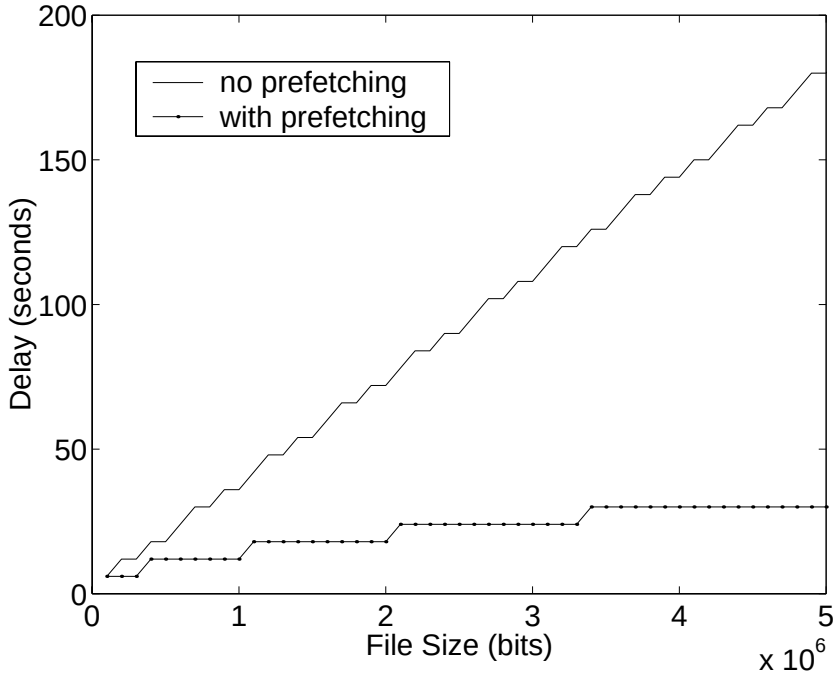

Figure 1.13 Delay as a function of the file size; $d_c = 0.05$ miles, $d = 0.2$ miles, $R_b = 56000$ bits/second, $v = 60$ miles/hour

Another interesting fact is that the velocity, v, affects the delay. Equations 1.3 and 1.9 show that the delay *decreases* as the velocity *increases*. Thus, it behooves the mobile to move rapidly, passing through a large number of infostations. As can be seen Figure 1.14, equation 1.9 is more sensitive to the velocity. In this case, the network-to-infostation bottleneck is essentially re-

moved by spreading the communication over many slow links. The ripples in the curve are due to the fact that the ceil function ($\lceil x \rceil$) remains constant for a range of values of v, while the denominator increases with v

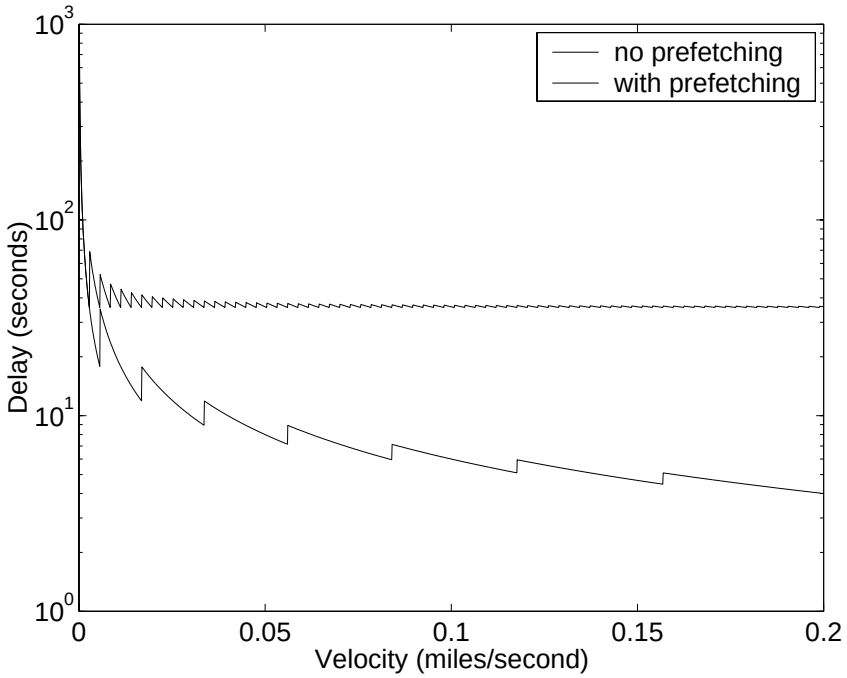

Figure 1.14 Delay as a function of the file size; $d_c = 0.05$ miles, $d = 0.1$ miles, $R_b = 56000$ bits/second, $F = 1$ Mbits

Certainly there is a limit to the improvement. It should first be noted that we assumed that there were as many infostations as needed in the user path. This may not be the case since the number of different infostations along a given tour may be limited. Furthermore, since the network usually transmits packets and frames, there may be some minimum number of bits, B_{min}, that can be delivered during a transmission. Thus, deliveries could only be made by every infostation along the tour if v is such that $B_{min} \leq R_b(d/v)$. However, even for infostation placement as close as one city block (0.05 mile), a typical $B_{min} = 128$ bytes, and a line transmission rate of 56Kbits/s, v would have to exceed 2.7 miles per second. For comparison, jetliner velocities are typically on the order of 0.1 miles per second.

2.2 THE RANDOM WALK

Now consider the scenario where the mobile moves with constant velocity, but at each step the direction is random – it can go to the right or to the left with probabilities p and $q = 1 - p$, respectively. Note that in this case the path is not known *a priori*. We would like to have bounds for file delivery delay for the random walk scenario. Let a *step* be a motion between two infostations and, at every step, the mobile either goes to the left or to the right, it does not stay at the same infostation.

2.2.1 Delay Bounds. Assume that we have an optimum algorithm, in the sense that it minimizes the delay. That is, the algorithm sends file parts to each infostation as if it knew the path. Assume the initial position is *position 0*. If the file size is F bits then it can be divided into N segments, such that

$$N = \left\lceil \frac{Fv}{R_b d} \right\rceil \tag{1.11}$$

If the optimum algorithm is used, then the maximum number of infostations needed to the left and right of the mobile so that the whole file can be downloaded is given by

$$I_{bound} = \left\lceil \frac{\sqrt{1+8N}-1}{2} \right\rceil \tag{1.12}$$

where I_{bound} represents a boundary where the mobile stays until file delivery completion. In other words, while the transaction is not completed, the mobile will be restricted to the region $[-I_{bound}, I_{bound}]$, as shown in Figure 1.15. If

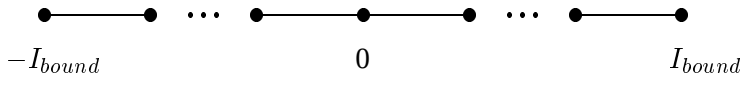

Figure 1.15 Bounds for the Mobile Path During the File Transference

the mobile actually goes on a straight line, the delay is the minimum possible given by

$$D_{min} = I_{bound}\frac{d}{v} = \left\lceil \frac{\sqrt{1+8N}-1}{2} \right\rceil \frac{d}{v} \tag{1.13}$$

If the mobile keeps hopping between two infostations then the maximum number of parts it can get is 1 in the first step and then 2 in all next steps. That

will create a situation where the mobile passes through the maximum number of infostations, I_{max} before file delivery completion. To find I_{max} will be the smallest integer such that

$$N \leq 1 + \sum_{i=2}^{I_{max}} 2 \tag{1.14}$$

Therefore, the maximum delay, in seconds, is given by

$$D_{max} = I_{max} \frac{d}{v} = \left\lceil \frac{N+1}{2} \right\rceil \frac{d}{v} \tag{1.15}$$

and thus,

$$D_{min} \leq D \leq D_{max} \tag{1.16}$$

Thus, equation 1.16 gives an upper an lower bound for the delay, in seconds, for file delivery completion using the prefetching approach.

2.2.2 Average Number of Segments. Equation 1.16 provides upper and lower bounds for the delay but it does not provide average delay for a given motion process. Therefore we will calculate the average number of segments that can be downloaded to an infostation if an optimum algorithm that minimizes the delay is used. From this we can infer the approximate average number of steps, and thereby delay, necessary to deliver files of various sizes.

Assume that the mobile passes through position i for the first time at time t and once again at time $t + n$. Assuming the download process starts at time 0, the *maximum* number of new file segments that can be downloaded to that infostation at time t is t file segments. Furthermore, the maximum number of new file segments that can be delivered at position i at time $t + n$ is n. Thus, it can be seen that for any optimal algorithm, the cumulative number of file segments obtained by a visit to location i at time t is *exactly* t regardless of the path taken to location i.

This result suggests that delay depends on the number of infostations that are revisited in a path, that is, if fewer infostations are *revisited* then more new file segments are obtained at each step and the time necessary to completely transmit the file is reduced.

Let us assume that the path is limited to s steps. Let $p(x, t|s)$ be the probability that location x was last visited at time $t < s$ given that the path is limited to s steps. If a location was visited at some time t, then the maximum number of segments that could have been downloaded, cumulatively, is t. If an optimum algorithm is used, then the number of segments downloaded would be exactly t. Given that, for an optimum algorithm, the mean number of file segments, $\bar{P}(s)$, picked up by step s is then

$$\bar{P}(s) = \sum_{t=1}^{s} \sum_{x} t\, p(x,t|s) \tag{1.17}$$

To calculate the value of $\bar{P}(s)$ it is necessary to calculate $p(x,t|s)$. We will do that using first passage times.

2.2.3 First Passage Times. Our goal is to calculate $p(x,t|s)$, the probability that location x was last visited at time $t < s$ given that the path is limited to s steps. If one looks at the motion process in the reverse direction, then $p(x,t|s)$ is simply the probability that, starting at some position n, the first passage through x is at time $s-t$. Observe that for that to be possible it is necessary that $|x-n| \leq (s-t)$.

Without loss of generality, assuming $n = 0$, the number of paths from the origin that pass through position x *for the first time* at time $s-t$, $x \neq 0$, is given by [20]

$$N_f(x, s-t) = \frac{|x|}{s-t} \begin{pmatrix} s-t \\ \frac{s-t+|x|}{2} \end{pmatrix} \tag{1.18}$$

$N_f(x, s-t)$ gives the number of possible paths. Each of these paths has a probability which is a function of p and q. The sum of all these probabilities gives the probability that the first passage time through position x is given at time $s-t$.

In order to calculate this probability we divide the problem in two cases: the positions to the left and right of the mobile. We will assume that the mobile at position 0. We first choose a position to the right of the mobile, say position r. We want to calculate the probability of a path that starts in 0 and passes through r for the first time in $s-t$ steps. We know that there are $N_f(r, s-t)$ paths with this property. But the probability of each one of them is exactly the same. This statement will become more clear with the discussion below.

The probability of a path that starts in 0 and passes through r for the first time in $s-t$ steps will be a function of the number of steps taken to the right and to the left. Assuming all steps are independent of each other, if the mobile take k_r steps to the right and k_l steps to the left, the probability, P_r, of this path is given by

$$P_r = p^{k_r} q^{k_l} \tag{1.19}$$

In order to satisfy the first passage time condition

$$k_r + k_l = s - t \tag{1.20}$$

In order to arrive to position r, we need that

$$r + 2k_l = s - t \tag{1.21}$$

which implies that

$$k_l = \frac{s-t-r}{2} \tag{1.22}$$

for all the $N_f(r, s-t)$ paths. The number of steps to the right, k_r, is given by

$$k_r = s - t - k_l = (s-t) - \frac{s-t-r}{2} = \frac{s-t+r}{2} \tag{1.23}$$

and then we conclude that the number of steps to the left and the number of steps to the right is the same for all $N_f(r, s-t)$ paths that start in 0 and end in r after $s-t$ steps.

It is important to note that k_l and k_r must be integers and therefore $s-t-r$ must be even or, equivalently, $s-t+r$ must be even.

The probability of a given path that starts in 0 and passes through position r for the first time in $s-t$ steps is then given by

$$P_r = p^{(s-t+r)/2} q^{(s-t-r)/2} \tag{1.24}$$

And there are

$$N_f(r, s-t) = \frac{r}{s-t} \left(\begin{array}{c} s-t \\ \frac{s-t+r}{2} \end{array} \right) \tag{1.25}$$

of these paths.

A similar analysis for a given position r to the left of the mobile gives that the probability, P_l, of a path that starts in 0 and passes through position $-r$ for the first time in $s-t$ steps is given by

$$P_l = p^{(s-t-r)/2} q^{(s-t+r)/2} \tag{1.26}$$

The total number of these path is also given by

$$N_f(-r, s-t) = \frac{r}{s-t} \left(\begin{array}{c} s-t \\ \frac{s-t+r}{2} \end{array} \right) \tag{1.27}$$

Note that this case also requires $s-t-r$ to be even.

Finally, the total average number of file segments picked by the mobile after s steps, $\bar{P}(s)$ is given by

$$\begin{aligned} \bar{P}(s) = & \sum_{t=1}^{s-1} \sum_{r=1, x \in \chi}^{s-t} t \frac{r}{s-t} \left(\begin{array}{c} s-t \\ \frac{s-t+r}{2} \end{array} \right) p^{(s-t-r)/2} \, q^{(s-t+r)/2} + \\ & \sum_{t=1}^{s-1} \sum_{r=1, x \in \chi}^{s-t} t \frac{r}{s-t} \left(\begin{array}{c} s-t \\ \frac{s-t+r}{2} \end{array} \right) p^{(s-t+r)/2} \, q^{(s-t-r)/2} + s \end{aligned} \tag{1.28}$$

where

$$\chi = \{r : (s - t - r) \bmod 2 = 0\} \tag{1.29}$$

where the first summation represents the positions to the left of the mobile and the second summation represents the positions to the right of the mobile.

2.2.4 Special Cases. Two special cases of interest are the symmetric case, where $p = q = 1/2$, and the straight line case, where $p = 1$, $q = 0$.

Assuming the symmetric case, the probability of each path is given 0.5^{s-t}. Then

$$p(r, t|s) = \frac{|r|}{s-t} \begin{pmatrix} s-t \\ \frac{s-t+|r|}{2} \end{pmatrix} 0.5^{s-t} \tag{1.30}$$

and

$$\bar{P}(s) = \sum_{t=1}^{s-1} \sum_{r=-(s-t), r\in\chi}^{s-t} t\frac{|r|}{s-t} \begin{pmatrix} s-t \\ \frac{s-t+|r|}{2} \end{pmatrix} 0.5^{s-t} + s \tag{1.31}$$

where

$$\chi = \{r : (|r| - (s - t)) \bmod 2 = 0\} \tag{1.32}$$

For the case where $p = 1$ ($q = 0$) we have the situation described in Section 2.1 (constant velocity), where the mobile comes from the left to the right in a straight line. In this situation the second summation will be zero, and the first summation is only non zero for $r = s - t$, since there is only one possible path. Therefore we have

$$\bar{P}(s) = \sum_{t=1}^{s-1} t + s = \frac{(s-1)s}{2} + s = \frac{s(s+1)}{2} \tag{1.33}$$

which is similar to the result in Section 2.1. In Equation 1.6 the variable T represents the number of infostations required for delivery of F bits and plays the role of the variable s in Equation 1.33. The difference between Equations 1.6 and 1.33 is the factor $(R_b d/v)$ used to transform from number of segments to number of bits.

Figure 1.16 shows the bounds on average number of pieces picked after a given number of steps, $\bar{P}(s)$, as a function of s. We start with $p = q = 0.5$ and we increase the value of p. Because of the symmetry of the problem, the results increasing the value of q are similar, or, in other words. p and q can be exchanged.

As can be seen, the results for the straight line ($p = 1.0$, $q = 0.0$) give highest bound. As the probability p increases the bound approaches the lower bound, which is achieved with the symmetric case ($p = q = 0.5$). This suggests that the average number of pieces is closely related to the number of revisits to a given place. In the straight line case the mobile never revisits a given position, and therefore at every step a higher number of pieces can be downloaded. As we mentioned in Section 2.2.1, the minimum delay is achieved when the mobile moves in a straight line and the maximum delay is achieved when the mobile hops between two infostations. In the next section we will discuss this topic further.

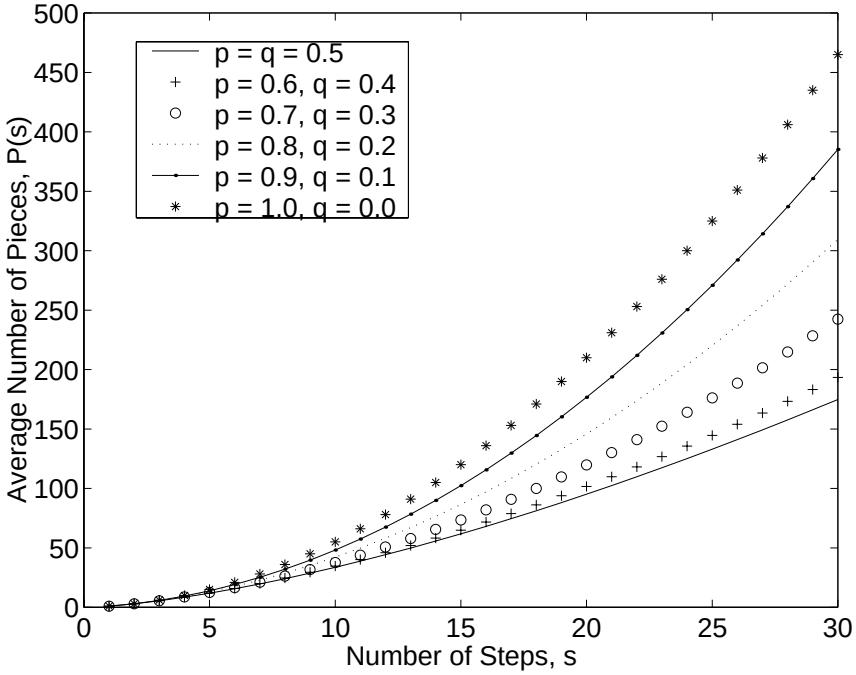

Figure 1.16 Average number of segments picked after a given number of steps, $\bar{P}(s)$ for different values of p and q.

3. EXTENDING TO HIGHER DIMENSIONS: THE GRID, THE CUBE, ETC.

Results for first passage times helped us to derive the bounds for the one-dimensional case. Since we were unable to find first passage time results for higher dimensions are not available in the literature, these results are provided here so that we may derive bounds on the average number of pieces delivered.

We assume the same radio and links rate scenario as in the single dimensional case.

3.1 THE TWO-DIMENSIONAL PROBLEM

We first extend the one-dimensional case to two-dimensions. Let us consider that the infostations are equally spaced in a rectangular grid as shown in Figure 1.17. The mobile can go right, left, up or down with probabilities p_r, p_l, p_u and p_d, respectively. At every step the mobile chooses one direction and does not stay at the same position. The velocity is still assumed constant for internode moves.

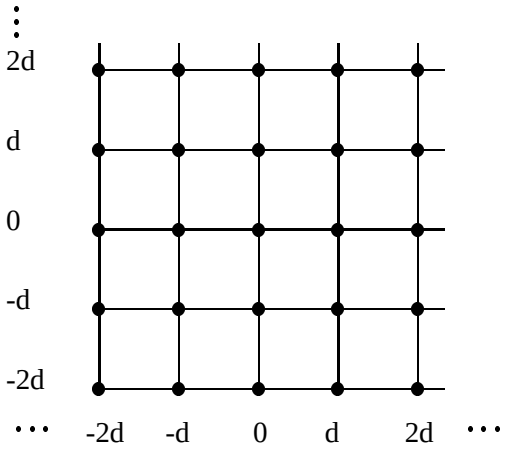

Figure 1.17 Two-dimensional Scenario

We want to calculate the average number of file segments that can be picked up by step s, $\bar{P}(s)$. Let $p(x, y, t|s)$ be the probability that location (x, y) was last visited at time $t < s$ given that the path is limited to s steps. Similar to the one-dimensional case, we have

$$\bar{P}(s) = \sum_{t=1}^{s} \sum_{(x,y)} t\, p(x, y, t|s) \tag{1.34}$$

Observing the motion process backwards, as we did in the last section, then $p(x, y, t|s)$ is simply the probability that, starting at some position (n_x, n_y), the first passage through (x, y) is at time $s - t$. Observe that for that to be possible it is necessary that $|x - n_x| + |y - n_y| \leq (s - t)$.

We need to calculate the first passage time through some position (x, y). Since we were unable to find this result in the literature we will derive it here,

starting with the multinomial distribution. Assume x and y positive. Without loss of generality, assume $n_x = n_y = 0$. We will first calculate the total number of paths that start in $(0,0)$ and pass through (x,y) in s steps, $N_{2d}(x,y,s)$. Note that a necessary condition is that $x + y \leq s$.

Let k_u be the number of steps taken upwards, k_d the number of steps downwards, k_r the number of steps taken to the left and k_r the number of steps taken to the right. The total number of paths that start in the origin and pass through (x,y) in step s is given by

$$N_{2d}(x,y,s) = \left(\begin{array}{c} s \\ k_u \end{array} \right) \left(\begin{array}{c} s-k_u \\ k_d \end{array} \right) \left(\begin{array}{c} s-k_u-k_d \\ k_r \end{array} \right) \left(\begin{array}{c} s-k_u-k_d-k_r \\ k_l \end{array} \right) \tag{1.35}$$

where

$$\left(\begin{array}{c} s \\ k \end{array} \right) = \frac{s!}{k!(s-k)!} \tag{1.36}$$

In order that the mobile arrives at position y, it is necessary that

$$k_d = k_u - y \tag{1.37}$$

And since it will also have to get to position x,

$$k_l = \frac{s-x-k_u-k_d}{2} = \frac{s-2k_u+y-x}{2} \tag{1.38}$$

and

$$k_r = \frac{s-x-k_u-k_l}{2} + x = \frac{s-2k_u+y+x}{2} \tag{1.39}$$

The number of steps upwards, k_u needs to satisfy

$$y \leq k_u \leq \frac{s-x-y}{2} + y \tag{1.40}$$

Because the number of steps has to be an integer, we also need that $(s+x+y)$ mod 2 = 0. Therefore, for $x > 0$ and $y > 0$, we have

$$N_{2d}(x,y,s) = \tag{1.41}$$

$$= \sum_{k_u=y}^{(s-x+y)/2} \frac{s!}{k_u!(k_u-y)!((s-2k_u+y-x)/2)!((s-2k_u+y+x)/2))!}$$

Note that the same holds for x or y non positive, depending only on the absolute value of x and y. Therefore the number of paths with s steps, from the origin

to (x, y) is given by

$$N_{2d}(x, y, s) = \tag{1.42}$$

$$= \begin{cases} \displaystyle\sum_{k=|y|}^{(s-|x|+|y|)/2} \frac{s!}{k!\,(k-|y|)!((s-2k+|y|-|x|)/2)!\,((s-2k+|y|+|x|)/2))!} : \\ \quad \text{if } |x|+|y| \leq s;\ (s+|x|+|y|) \bmod 2 = 0 \text{ and } (x, y, s) \neq (0, 0, 0) \\ 0 : \text{otherwise} \end{cases}$$

$N_{2d}(x, y, s)$ gives the total number of paths from the origin to some position (x, y) in s steps. That includes paths that pass through (x, y) at some time $t < s$ also. In order to obtain the number of paths that have the *first passage* through (x, y) at step s, it is necessary to remove the paths that pass through (x, y) at s *and before* s *also*.

Let $N_e(i)$ be the number of paths that start at some position and return to that same position *for the first time* in i steps. Note that i must be even. Then, the total number of paths starting at the origin that pass *for the first time* through position (x, y) in s steps, $N_{f_{2d}}(x, y, s)$, is given by

$$N_{f_{2d}}(x, y, s) = N_{2d}(x, y, s) - \sum_{i=2, i\ even}^{s-1} N_{2d}(x, y, s-i)\, N_e(i) \tag{1.43}$$

where we eliminated the paths that also pass through that position before time s. These are the paths that pass through that position for the first time at time $s-i$, and then return to that position in i steps, for all i even, $i \geq 2$ and $s - 1 \geq 2$.

To calculate $N_e(i)$ we start with the total paths from some position to itself in i steps, $N_{2d}(0, 0, i)$. To calculate the total number of paths that start at some position and return to the same position *for the first time* in i steps it is necessary to eliminate the paths that also pass through that position before time i. Similarly to before, these are the paths that pass through that position for the first time at time $i - p$, and then return to that position in p steps, for all p even, $p \geq 2$ and $i - p \geq 2$.

$$N_e(i) = N_{2d}(0, 0, i) - \sum_{p=2, p\,even}^{i-2} N_e(i-p)\, N_{2d}(0, 0, p) \tag{1.44}$$

And we know that

$$N_e(2) = 4 \tag{1.45}$$

therefore the value of $N_e(i)$ can be calculated, using recursion, for any $i \geq 2$, i even.

The probability of the paths that start at the origin and pass through position (x, y) at time s, $Pr_{2d}(x, y, s)$, will be a function of x, y and s and is given by

$$Pr_{2d}(x, y, s) = \tag{1.46}$$
$$= \sum_{k_u=|y|}^{(s-|x|+|y|)/2} \frac{s! \quad p(k_u)}{k_u!(k_u-|y|)!((s-2k_u+|y|-|x|)/2)!((s-2k_u+|y|+|x|)/2))!}$$

where

$$p(k_u) = p_u^{k_u}\, p_d^{k_d}\, p_r^{k_r}\, p_l^{k_l} = p_u^{k_u}\, p_d^{k_u-|y|}\, p_r^{(s-2k_u+|y|-|x|)/2}\, p_l^{(s-2k_u+|y|+|x|)/2)} \tag{1.47}$$

But calculating the probability of the paths that start at the origin and pass *for the first time* at position (x, y) at time s is not trivial. For that reason we will consider here the symmetric case where $p_r = p_l = p_u = p_d = 0.25$. Observing the motion process backwards, as we did in the last section for the one-dimensional case, and considering the symmetric case, one can derive the average number of file segments picked up by step s is then given by

$$\bar{P}_2(s) = \; s \; + \sum_{t=1}^{s-1} \sum_{\substack{(x,y)=(t-s,t-s) \\ (x,y)\neq(0,0)}}^{(x,y)=(s-t,s-t)} t\, N_{f_{2d}}(x, y, s-t)\, 0.25^{s-t} \tag{1.48}$$

The condition $(x, y) \neq (0, 0)$ is included because we assume that the user is at the origin at step s and therefore it got cumulatively s segments, which are added separately in equation 1.48.

In Figure 1.18 we present the average number of segments picked after s steps, $\bar{P}(s)$ for the symmetric two-dimensional case. We also present $\bar{P}(s)$ for the one-dimensional case, for different values of p and q (please refer to Section 2). One could expect that the curve for two-dimensional case would match with the case of $p = 0.75$, $q = 0.25$. That is not the case because in the one dimensional case the probability of revisits to a given position when $p = 0.75$, $q = 0.25$ is higher than for the two-dimensional case. As we can see from the figure, the value that most approaches the 2-d case is $p = 0.8$, $q = 0.2$.

3.2 THE N-DIMENSIONAL PROBLEM

Extending the two-dimensional to the n-dimensional problem is straightforward. Of course, one might ask "why bother?" First, one could easily imagine 3-dimensional infostations models. However, although the infostations problem was couched in terms of moving matter (people, vehicles, etc.) and radios,

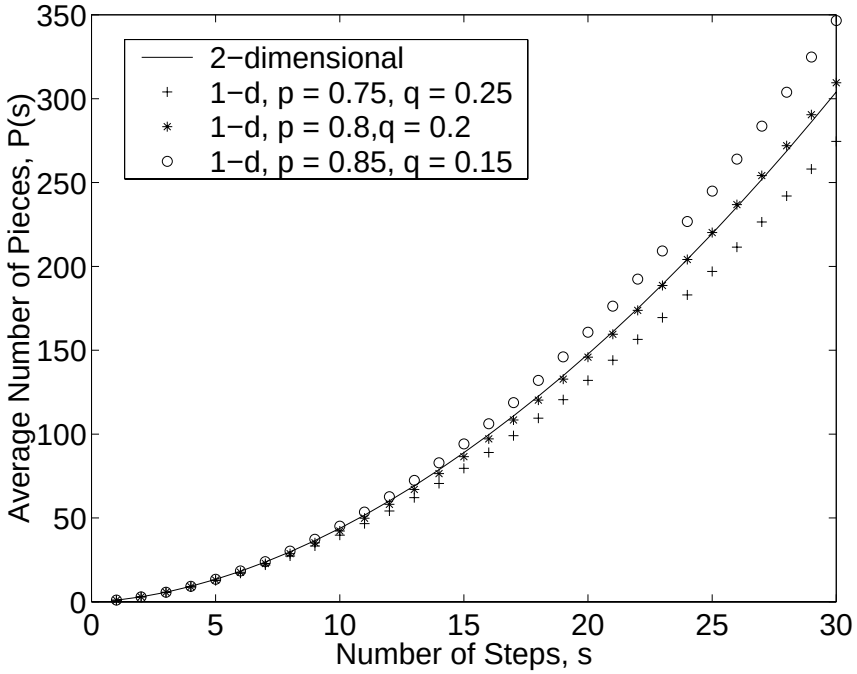

Figure 1.18 Comparison between average number of segments picked after a given number of steps, $\bar{P}(s)$, for one-dimensional and two-dimensional cases.

a possibly quixotic generalization might include migrant programs (mobile agents) operating over a computer network in an almost arbitrary dimensional data space and other fixed programs which need to pass large data files to the agents as they move over the network. In addition, practical utility aside, the same machinery to consider three dimensions allows consideration of N dimensions. Thus, since the incremental effort necessary for generalization is minimal, it is therefore provided here.

Equation 1.44 is still valid for the n-dimensional case, where now $N_{nd}(x_1, x_2, \ldots, x_n, s)$ is the number of paths with s steps from the origin to position $(x_1, x_2, \ldots, x_n)$. This number is easily calculated using the multinomial distribution as in Equation 1.42, but now with $n-1$ summations, as

$$N_{nd}(x_1, ..., x_n, s) = \tag{1.49}$$

$$= \begin{cases} \sum_{k_i \in \alpha} \frac{s!}{k_1!(k_1-|x_1|)!k_2!(k_2-|x_2|)!\cdots k_n!(k_n-|x_n|)!} : \\ \text{if } |x_1|+...+|x_n| \leq s; (s-|x_1|-...-|x_n|) \text{ mod 2 = 0} \\ \text{and } (x_1,...,x_n,s) \neq (0,...,0,0) \\ 0 : \text{otherwise} \end{cases}$$

and

$$\alpha = \{k_i : \sum_{j=1}^{n} (2k_j - |x_j|) = s \text{ and } k_i \geq |x_i| \text{ for } i = 1,...,n\} \tag{1.50}$$

For the case where $n = 3$, for example, we would have

$$N_{3d}(x,y,z,s) = \sum_{k_2} \sum_{k_3} \frac{s!}{k_1!\,(k_1-|x|)!\,k_2!\,(k_2-|y|)!\,k_3!\,(k_3-|z|)!} \tag{1.51}$$

But we know from condition 1.50 that

$$2k_1 + 2k_2 + 2k_3 - x - y - z = s \tag{1.52}$$

which implies that

$$k_1 = \frac{s + x + y + z - 2k_2 - 2k_3}{2} \tag{1.53}$$

We also know that the maximum value of k_2 is obtained when $k_1 = |x|$ and $k_2 = |y|$ and therefore from condition 1.50 the maximum value of k_2 is obtained when

$$2k_2 - |y| + |x| + |z| = s \tag{1.54}$$

which implies that

$$k_2 \leq \frac{s - |x| + |y| - |z|}{2} \tag{1.55}$$

and similarly, the maximum value of k_3 is obtained when

$$2k_3 - |y| + |x| + 2k_2 - |z| = s \tag{1.56}$$

which implies that

$$k_3 \leq \frac{s - |x| - (2k_2 - |y|) + |z|}{2} \tag{1.57}$$

and finally we can write

$$N_{3d}(x,y,z,s) = \tag{1.58}$$
$$= \sum_{k=|y|}^{k_{max}} \sum_{j=|z|}^{j_{max}} \frac{s!}{k!(k-|y|)!\, j!(j-|z|)!(\frac{s+|y|+|z|-2k-2j+|x|}{2})!(\frac{s+|y|+|z|-2k-2j-|x|}{2})!}$$

where

$$k_{max} = \frac{s-|x|+|y|-|z|}{2} \tag{1.59}$$

and

$$j_{max} = \frac{s-|x|-2k+|y|+|z|}{2} \tag{1.60}$$

Returning to the general case, the number of paths that start at the origin and pass through $(x_1, ..., x_n)$ *for the first time* after s steps, $N_{f_{nd}}(x_1, ..., x_n, s)$, is given by

$$N_{f_{nd}}(x_1, ..., x_n, s) = N_{nd}(x_1, ..., x_n, s) - \sum_{i=2,\, i\, even}^{s-1} N_{nd}(x_1, ..., x_n, s-i)\, N_e(i) \tag{1.61}$$

where

$$N_e(i) = N_{nd}(0, ..., 0, i) - \sum_{p=2,\, p\, even}^{i-2} N_e(i-p)\, N_{nd}(0, ..., 0, p) \tag{1.62}$$

and

$$N_e(2) = 2n \tag{1.63}$$

In general, for the symmetric n-dimensional case, we can write the average number of file segments picked at step s as

$$\bar{P}_n(s) = s + \sum_{t=1}^{s-1} \sum_{\substack{(x_1,...,x_n)=(t-s,...,t-s) \\ (x_1,...,x_n)\neq(0,...0)}}^{(x_1,...,x_n)=(s-t,...,s-t)} t\, N_{f_{nd}}(x_1, ..., x_n, s-t) \left(\frac{1}{2n}\right)^{s-t} \tag{1.64}$$

Figure 1.19 shows the average number of file pieces picked in s steps for one-, two- and three-dimensional problems, for the symmetric cases. $\bar{P}(s)$

increases as n increases since the number of infostations revisited in a path decreases. Also shown in Figure 1.19 is the case where the mobile never revisits an infostation. Figure 1.20 shows the average number of times the mobile revisits a given position in a motion process along a path of length s. This is simply the number of paths that start in the origin and return to the origin after s steps times the probabilities these paths. For the two-dimensional symmetric case, for example, the average number of revisits in s steps is given by $N_{2d}(0,0,s)0.25^s$. As can be seen from the figure, as the number of dimensions increases the number of revisits decreases.

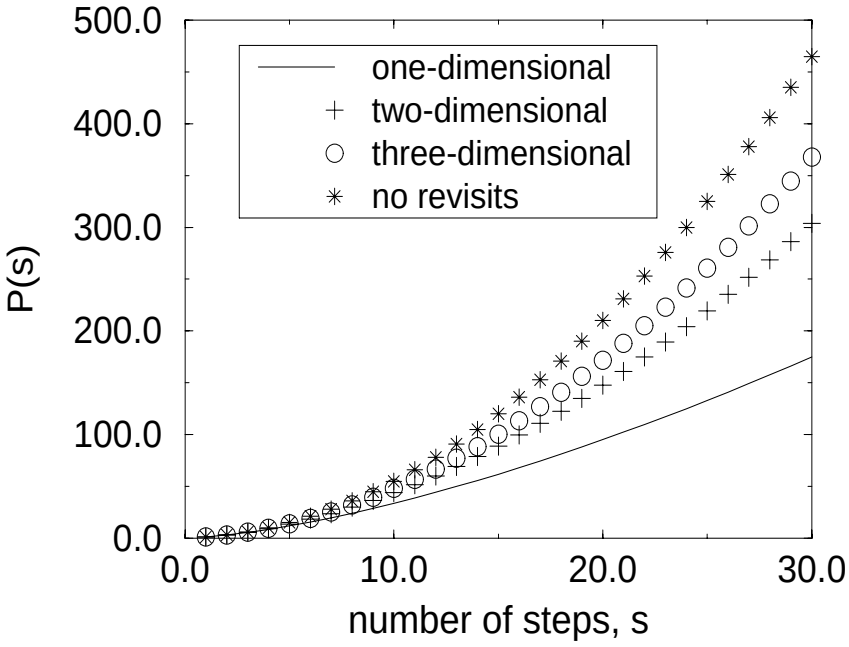

Figure 1.19 Average number of file segments picked up in s steps, $\bar{P}(s)$.

4. A NEAR-OPTIMUM ALGORITHM

In the last two sections we presented bounds on file delivery for a general n-dimensional model. Those results give bounds on the maximum number of file segments that can be downloaded for the user in the case where $R_i > NR_b$ and $R_r > MR_b$ and were derived assuming that an optimum algorithm is used. But it does not tell us how such an algorithm would work. Therefore, in this section we will provide an algorithm for the one-dimensional case. Given that the file can be divided in many different smaller segments, and we label each

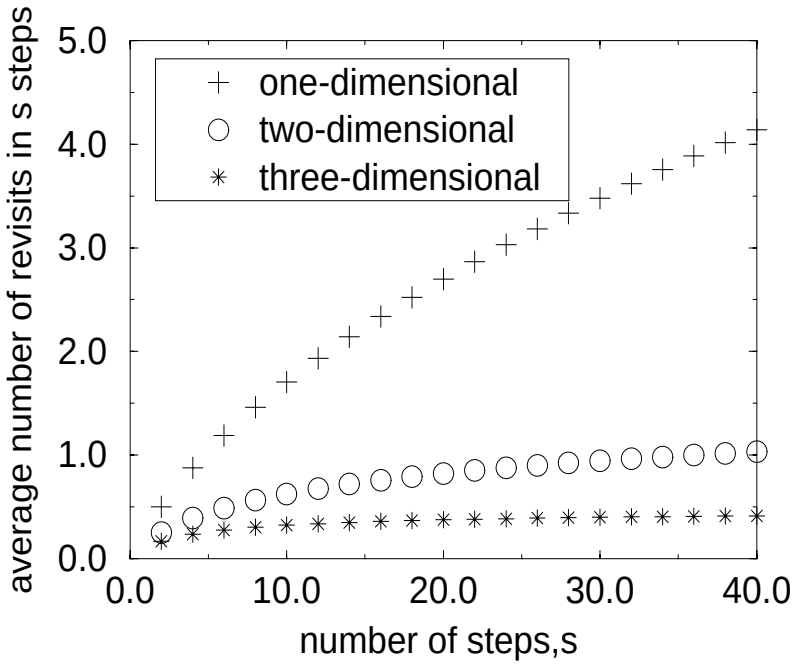

Figure 1.20 Average number of revisits for a motion process with s steps.

segment, the role of the algorithm is to decide after every step which segments should be sent to which infostations. As always, the goal is to minimize the overall file transfer delay.

4.1 OVERVIEW

We will concentrate in the one-dimensional scenario, as shown in Figure 1.21, where infostations are equally spaced at distance d.

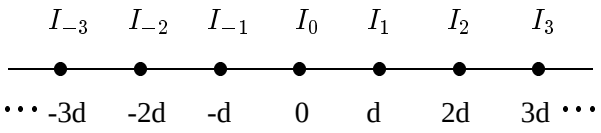

Figure 1.21 The One-dimensional Model

Assume a file of size F, and this file is divided in N segments of size B, where

$$B = \frac{R_b d}{v} \tag{1.65}$$

and

$$N = \left\lceil \frac{F}{B} \right\rceil = \left\lceil \frac{F v}{R_b d} \right\rceil \tag{1.66}$$

Given that the mobile is at some position x, an algorithm that delivers file segments to infostations around that user will have to deliver segments to a given number of infostation to the right and left of the mobile. Therefore our algorithm will work in a range of infostations. The first important characteristic of the algorithm is the calculation of the boundaries in the range. These boundaries are the maximum number of infostations necessary to the left and right of the mobile so that the mobile is able to receive the whole file. Note that they should be recalculated at every step of the algorithm.

In order to maximize the number of file segments received at each step, the algorithm schedules different segments to each infostation, so that there are no repetitions and no segments wasted. After all the segments are spread over the range of infostations, then one can no longer avoid repetitions. From that point on the algorithm avoids sending repeated segments to positions that are in a possible path for the mobile. Details of the algorithm are described in the following sections.

4.2 THE RANGE OF INFOSTATIONS

As seen in Section 2.2, as long as the mobile picks the maximum number of file segments at every step, the maximum number of infostations needed to the left and right of the starting position is given by I_{bound} where

$$I_{bound} = \left\lceil \frac{\sqrt{1+8N}-1}{2} \right\rceil \tag{1.67}$$

If the mobile is at position i then an optimum algorithm does not have to consider infostations that are outside the range $[i - I_{bound}, i + I_{bound}]$.

After the mobile moves to the left or right new boundaries must be calculated. To do so the history of the movements is considered. All segments that were already obtained by the mobile and all the segments already delivered to infostations are considered. In addition, undelivered new segments still to be scheduled must also be taken into consideration. The number of new segments that will be scheduled is calculated using the assumption that an optimum algorithm will be used. Note that the boundaries will be given by one infostation to the left of the mobile and one infostation to he right of the mobile. We define

- I_r: infostation at which the mobile user will receive the last file segment if it moves from infostation i to I_r in a straight line, taking only steps to

the right. We assume that, when this path is taken, the mobile receives all the segments already delivered to the infostations in the path plus new segments that will be delivered using an optimum algorithm (new segments would not be copies of segments delivered before).

- I_l: infostation at which the mobile user will receive the last file segment if it moves from infostation i to I_l in a straight line, taking only steps to the left. We assume that, if this path is take, the mobile receives all the segments already delivered to the infostations in the path plus new segments that will be delivered using an optimum algorithm (new segments would not be copies of segments delivered before).

For example, consider the situation where the file can be divided in 10 segments, $P_1, \ldots, P_{10}$. In this case the maximum number of infostations needed to each side of the mobile is

$$I_{bound} = 4 \tag{1.68}$$

Assume that the mobile is at position 5. Then the range of infostations is given by [1,9], as shown in Figure 1.22.

position:	1	2	3	4	5	6	7	8	9
					M				

Figure 1.22 Example for a file with 10 segments: calculating boundaries, $I_l = 1$ and $I_r = 9$

After the boundaries are calculated, then the algorithm must decide which segments to prefetch in which infostations during the first step. The algorithm will deliver different segments to every infostation in the range, as shown in Figure 1.23

step 0	P_8	P_6	P_4	P_2	P_9	P_1	P_3	P_5	P_7
position:	1	2	3	4	5	6	7	8	9
					M				

Figure 1.23 Example for a file with 10 segments: scheduling of segments in the range

After the mobile moves, say to position 4, one segment (P_2) is picked up, as shown in Figure 1.24.

step 1	P_8	P_6	P_4	X	P_9	P_1	P_3	P_5	P_7
position:	1	2	3	4	5	6	7	8	9
				M					

Figure 1.24 Example for a file with 10 segments: after first step the mobile is at position 4 and one segment is delivered

It is now necessary to calculate the new boundaries. The left boundary is still the same, and the right boundary is calculated using the following facts:

- Mobile has already received P_2. There are nine segments missing to complete the file.
- If the mobile goes in a straight line to the right it will pick all the segments that are scheduled plus new segments that will be scheduled. If we assume that the next segments will be scheduled using an optimum algorithm, then the mobile would pick two segments in position 5, three segments in position 6 and finally four segments in position 7, which gives a total of nine segments and therefore the mobile would finish the transmission at position 7 (please see Figure 1.25.

step 4							$*, o$
step 3						$*, o$	o
step 2					$*, o$	o	o
step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.25 After the first step the mobile is at position 4 and the path $4 \rightarrow 5 \rightarrow 6 \rightarrow 7$ is taken. *: represents future mobile position, o: represents a segment that will be delivered to an infostation

Therefore the new boundaries are $I_l = 1$ and $I_r = 7$ and positions 8 and 9 do not have to be considered anymore in our scheduling process.

Now the algorithm has to schedule segments to be delivered to the range $[1, 7]$. Note that the only segment delivered to the mobile is P_2 and therefore should not be scheduled. Note also that segments P_1, P_3, P_4, P_6, P_8 and P_9 are already delivered to the infostations in the range. Therefore segments P_5, P_7 and P_{10} should be scheduled. The other five segments must be a copy of segments already delivered to the infostations. One way of scheduling segments is shown in Figure 1.26

	P_{10}	P_7	P_5	P_3	P_1	P_9	P_8
step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.26 Example for a file with 10 segments: scheduling of segments for second step

Now suppose the mobile moves to position 5 and picked segments P_1 and P_9. The boundaries must be re-calculated.

The mobile has already received 3 segments (P_1, P_2, P_9). To the left the mobile can receive two segments at infostation 4, four segments at infostation 3 and five segments at infostation 2, a total of eleven segments (please see Figure 1.27) Since only seven segments are missing, the left boundary is $I_l = 2$. To the right, since segments P_1 and P_9 were already delivered to the user, they are wasted at infostation 6. Therefore at infostation 6 the mobile can receive only one new segment, four segments at infostation 7 and four segments at infostation 8, a total of nine segments (please see Figure 1.28. Thus the right boundary is $I_r = 8$.

step 5	$*, o$						
step 4	o	$*.o$					
step 3	o	o	$*.o$				
step 2	P_7	P_5	P_3	X	P_9	P_8	
step 1	P_6	P_4	X	X	P_1	P_3	P_5
position:	2	3	4	5	6	7	8
				M			

Figure 1.27 Calculating the left boundary after the second step. *: represents the future user position, o: represents a segment that will be delivered to an infostation

Note that the scheduling choice was not very smart and some segments were wasted. Choosing *where* to copy *which* segments is not an easy task, and it is the objective of the algorithm. We will discuss the scheduling of segments in the next sections.

Note that if the algorithm is optimum then the new boundaries may decrease, but never increase.

We then define, at any given step:

step 5							$*, o$
step 4						$*, o$	o
step 3					$*, o$	o	o
step 2	P_7	P_5	P_3	X	P_9	P_8	
step 1	P_6	P_4	X	X	P_1	P_3	P_5
position:	2	3	4	5	6	7	8
				M			

Figure 1.28 Calculating the right boundary after the second step. *: represents the future user position, o: represents a segment that will be delivered to an infostation

- N: total number of segments in the file.
- i: mobile user position.
- δ: total number of segments already delivered to the mobile user.
- $\eta(j)$: set of segments that were scheduled to infostation j but not delivered to the mobile user.
- $\lambda(i, j)$, $i \leq j$: set with segments that were scheduled to infostations in the range $[i, j]$. Thus

$$\lambda(i, j) = \bigcup_{x=i}^{j} \eta_x \tag{1.69}$$

And with these definitions we can write I_r and I_l as follows.

- The right boundary is the smallest integer I_r such that:

$$N - \delta > |\lambda(i + 1, I_r)| + \sum_{x=i+1}^{I_r} (x - i) \tag{1.70}$$

- The left boundary is the largest integer I_l such that:

$$N - \delta > |\lambda(I_l, i - 1)| + \sum_{x=I_l}^{i-1} (x - I_l + 1) \tag{1.71}$$

4.3 PARTIAL PATHS

To maximize the number of file segments received at each step, the algorithm should schedule different segments to each infostation, so that there are no repetitions and no segments wasted. After all the segments are spread over the range $[I_l, I_r]$ of infostations, then one can no longer avoid repetitions. The problem is then to decide which copies of segments should be sent to each infostation.

It is important to note that although we can calculate the boundaries, that does not imply that the mobile will pass through all the infostations in that range. For example, if the mobile goes in a straight line to the left, it will never visit infostations to the right of its initial position. This fact is very important and will be used in the algorithm.

To be able to describe the algorithm we will define:

- $\gamma(P)$: maximum number of file segments that can be delivered to the mobile if it takes a given path P.
- Partial Path from infostation i through infostation j: a path that starts at infostation i, passes through infostation j, and the mobile has the potential to receive *at most* the total number of file segments still needed. Thus a path P is a partial path if and only if

$$\gamma(P) \leq N - \delta \tag{1.72}$$

- $\phi(i, j)$: set of *all* infostations that belong to *all* partial paths from i through j.
- $\mu(i, j)$: set with all segments scheduled to infostations that belong to ϕ_i^j

$$\mu(i, j) = \{\eta(k) \mid k \in \phi(i, j)\} \tag{1.73}$$

To better understand the idea of partial paths, we consider the example given before. The mobile started at position 5, moved to position 4 and received one segment (P_2), as shown in Figure 1.29.

step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.29 Example: at the first step the mobile is at position 4 and picked one segment (P_2)

Let us obtain the partial paths from infostation 4 through infostation 5. The paths $A = \{4 \rightarrow 5\}$, $B = \{4 \rightarrow 5 \rightarrow 6\}$ and $C = \{4 \rightarrow 5 \rightarrow 6 \rightarrow 7\}$ are all partial paths from 4 through 5. In path A the mobile can receive at most two segments at infostation 4 ($2 \leq 9$); in path B the mobile can receive two segments at infostation 5 and three segments at infostation 6, a total of five segments ($5 \leq 9$) and in path C the mobile can receive two segments at infostation 5, three segments at infostation 6 and four segments at infostation 7, a total of nine segments ($9 \leq 9$).

Note that the path $\{4 \rightarrow 5 \rightarrow 4\}$ is also a partial path from 4 through 5, but it includes the same infostations.

The paths $D = \{4 \rightarrow 3 \rightarrow 4 \rightarrow 5\}$ and $E = \{4 \rightarrow 5 \rightarrow 4 \rightarrow 3\}$ are also a partial paths from infostation 4 through infostation 5.

In path D (please refer to Figure 1.30) the maximum number of segments that can be delivered is $\gamma(D) = 2$ segments (at infostation 3) $+ 2$ segments (at infostation 4) $+ 4$ segments (at infostation 5) $= 8 < 9$

step 4					$*, o$		
step 3				$*, o$	o		
step 2			$*, o$	o	o		
step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.30 The mobile is at position 4, picked one segment and takes the path $D = \{4 \rightarrow 3 \rightarrow 4 \rightarrow 5\}$. *: represents future mobile position, o: represents a segment that will be delivered to an infostation

In path E (please refer to Figure 1.31) the maximum number of segments that can be delivered is $\gamma(E) = 2$ segments (at infostation 5) $+ 2$ segments (at infostation 4) $+ 4$ segments (at infostation 3) $= 8 < 9$

We then know that infostations $[3, 7]$ belong to partial paths from 4 through 5, or belong to $\phi(4, 5)$. In order to find if infostation 2 belongs to $\phi(4, 5)$ we observe the paths $F = \{4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2\}$ (Figure 1.32) and $G = \{4 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5\}$ (Figure 1.33). Since $\gamma(F) = 2$ segments (at infostation 5) $+ 2$ segments (at infostation 4) $+ 4$ segments (at infostation 3) $+ 5$ segments (at infostation 2) and $\gamma(G) = 2$ segments (at infostation 3) $+ 3$ segments (at infostation 2) $+ 2$ segments (at infostation 3) $+ 4$ segments (at infostation 4) $+ 6$ segments (at infostation 5) then

step 4			$*, o$				
step 3			o	$*, o$			
step 2			o	o	$*, o$		
step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.31 The mobile is at position 4, picked one segment and takes the path $E = \{4 \rightarrow 5 \rightarrow 4 \rightarrow 3\}$. *: represents future mobile position, o: represents a segment that will be delivered to an infostation

$$\gamma(F) = 13 > 9 \tag{1.74}$$

and

$$\gamma(G) = 17 > 9 \tag{1.75}$$

and thus G and F are not partial paths from 4 through 5. Therefore infostations 1 and 2 do not belong to partial paths from infostation 4 through 5, and

$$\phi(4, 5) = \{3, 4, 5, 6, 7\} \tag{1.76}$$

step 5		$*, o$					
step 4		o	$*, o$				
step 3		o	o	$*, o$			
step 2		o	o	o	$*, o$		
step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.32 The mobile is at position 4, picked one segment and takes the path $F = \{4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2\}$. *: represents future mobile position, o: represents a segment that will be delivered to an infostation

It is important to note that after the boundaries are calculated, all the infostations to the left of the mobile, except from I_l, always belong to a partial path between the mobile and any infostation to the left of the mobile. The infostation I_l may or may not belong to a partial path. The same argument applies to infostations to the right of the mobile.

step 5						$*, o$	
step 4					$*, o$	o	
step 3				$*, o$	o	o	
step 2			$*, o$	o	o	o	
step 1	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.33 The mobile is at position 4, picked one segment and takes the path $G = \{4 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5\}$. *: represents future mobile position, o: represents a segment that will be delivered to an infostation

For example, consider the case where the file divided in a total of $N = 2$ segments. Assume the mobile is at position 3. The boundaries in this case are $I_l = 1$ and $I_r = 5$. but infostation 5 does not belong to a partial path from 3 through 4, since $\gamma(3 \rightarrow 4 \rightarrow 5) = 3 > 2$.

4.3.1 Calculating $\gamma(P)$. As seen in the last section, in order to verify if a path P is a partial path one need to find $\gamma(P)$. Therefore it is important to have a general equation for $\gamma(P)$.

Consider the scheme shown in Figure 1.34.

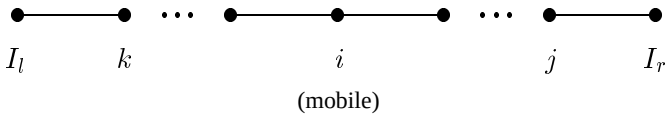

Figure 1.34 Calculating $\gamma(P)$

Consider the path A as

$$A = \{i \rightarrow k \rightarrow i \rightarrow j\} \tag{1.77}$$

$\gamma(A)$ can be written as (please see Figure 1.35):

$$\begin{aligned} \gamma(A) &= |\lambda(k,j)| + (i-k)(j-k+1) + \sum_{n=1}^{j-k} n = \\ &= |\lambda(k,j)| + (i-k)(j-k+1) + \frac{(j-k)(j-k+1)}{2} \end{aligned} \tag{1.78}$$

It is interesting to observe the example in Figure 1.35 where we can clearly see that at step s the mobile can pick cumulatively at most s segments.

step 10							$*, o$
step 9						$*, o$	o
step 8					$*, o$	o	o
step 7				$*, o$	o	o	o
step 6			$*, o$	o	o	o	o
step 5		$*, o$	o	o	o	o	o
step 4	$*, o$	o	o	o	o	o	o
step 3	o	$*, o$	o	o	o	o	o
step 2	o	o	$*, o$	o	o	o	o
step 1	o	o	o	$*, o$	o	o	o
position:	k				i		j
					M		

Figure 1.35 Calculating $\gamma(i \to k \to i \to j)$ *: represents future mobile position, o: represents a segment that will be delivered to an infostation

Consider now a path B given by

$$B = \{i \to j \to i \to k\} \tag{1.79}$$

Similarly

$$\gamma(B) = |\lambda(k,j)| + (i-j)(j-k+1) + \frac{(j-k)(j-k+1)}{2} \tag{1.80}$$

And therefore, in general, for a path $P = \{i \to x \to i \to y\}$ where x and y are at opposite sides of i and x is visited before y, $\gamma(P)$ will be a function of (i, x, y) and can be written as

$$\begin{aligned} \gamma(P) &= |\lambda(\min(x,y), \max(x,y))| + (|i-x|)(|y-x|+1) + \\ &\quad + \frac{(|y-x|)(|y-x|+1)}{2} \end{aligned} \tag{1.81}$$

where if $x < i$ then $y > i$ and if $x > i$ then $y < i$ and x is visited before y.

For example let us consider again the example given in Figure 1.29 and the paths $F = \{4 \to 5 \to 4 \to 3 \to 2\}$ and $G = \{4 \to 3 \to 2 \to 3 \to 4 \to 5\}$. Then, for path F, $i = 4$, $x = 5$, $y = 2$, $\lambda(2,5) = \{P_4, P_6, P_9\}$, $|\lambda(2,5)| = 3$. From Equation 1.80

$$\gamma(F) = \gamma(4,5,2) = 3 + (1 * 4) + (3 * 4)/2 = 13 \tag{1.82}$$

For path G $i = 4$, $x = 2$, $y = 5$, $\lambda(2,5) = \{P_4, P_6, P_9\}$, $|\lambda(2,5)| = 3$. from equation 1.78

$$\gamma(G) = \gamma(4,2,5) = 3 + (2 * 4) + (3 * 4)/2 = 17 \tag{1.83}$$

and the results match with the one presented in the last section, given by Equations 1.74 and 1.75.

4.3.2 Infostations in Partial Paths. As we mentioned before, after the boundaries are calculated, all the infostations to the left of the mobile, except from I_l, always belong to a partial path between the mobile and any infostation to the left of the mobile. The infostation I_l may or may not belong to a partial path. The same argument applies to infostations to the right of the mobile.

In order to guarantee the inclusion of I_r and I_l we define

$$\theta(i,j) = \begin{cases} \phi(i,j) \cup \{I_r\} \text{ if } i < j \\ \phi(i,j) \cup \{I_l\} \text{ if } i > j \\ \phi(i,j) \text{ if } i = j \end{cases} \tag{1.84}$$

$$\nu(i,j) = \{\eta(k) \mid k \in \theta(i,j)\} \tag{1.85}$$

Our goal now is to obtain the set $\theta(i,j)$ or $\phi(i,j)$. We want to find all the infostations that belong to all partial paths from i through j. As we saw in the example in the last section, it is not necessary to obtain *all* partial paths from i through j in order to obtain $\phi(i,j)$.

Consider again the scheme shown in Figure 1.34. The mobile is at position i and one wants to find $\theta(i,j)$, where $j > i$, or j is an infostation to the right of the mobile. We know that he set $\theta(i,j)$ is constituted of all infostations in $[k, I_r]$, where k is the left-most infostation that belongs to a partial path. In order to check if an infostation k belongs to a partial path one could check if the paths

$$A = \{i \to k \to i \to j\} \tag{1.86}$$

and

$$B = \{i \to j \to i \to k\} \tag{1.87}$$

are partial paths. Therefore it is necessary to verify, for each path, if the number of segments that the mobile can receive if the path is taken is less than or equal to the total number of file segments still needed. We need to verify if

$$\gamma(A) = \gamma(i,k,j) \leq N - \delta \tag{1.88}$$

and

$$\gamma(B) = \gamma(i,j,k) \leq N - \delta \tag{1.89}$$

Note that both paths contain the same infostations, but they may be of different sizes, or different number of steps. Define

- $\rho(P)$: the number of steps in a given path P.

But we know that

$$\rho(A) < \rho(B) \iff \gamma(A) < \gamma(B) \tag{1.90}$$

and therefore, it is necessary and sufficient to check if the *shortest path* that starts at i and passes through j and k is a partial path.

So we conclude that, in general, to check if an infostation k is in a partial path from i through j, where i is in the middle of k and j

- if $j \neq i$, to check if an infostation k is in a partial path from i through j it is necessary and sufficient to check
  - if $\min(|i-k|, |j-i|) = |i-k|$ then check if $\gamma(i,k,j) \leq N - \delta$
  - if $\min(|i-k|, |j-i|) = |j-i|$ then check if $\gamma(i,j,k) \leq N - \delta$
- if $j = i$ then to check if an infostation k is in a partial path from i through i it is necessary and sufficient to check if
  - $\gamma(i,k,i) \leq N - \delta$

Therefore:

- if $i \neq j$ then
  - if $\min(|i-k|, |i-j|) = |i-k|$ then
    - if $\gamma(i,k,j) \leq N - \delta$ then
      - k belongs to a partial path from i through j.
  - if $\min(|i-k|, |i-j|) = |i-j|$ then
    - if $\gamma(i,j,k) \leq N - \delta$ then
      - k belongs to a partial path from i through j.
- if $i = j$ then
  - if $\gamma(i,k,i) \leq N - \delta$ then
    - k belongs to a partial path from i through i.

4.3.3 Obtaining $\theta(i,j)$. As we saw in the last section, we can write

$$\theta(i,j) = \begin{cases} \{k_l, \cdots, i, \cdots, I_r\} \text{ if } i < j \\ \{I_l, \cdots, i, \cdots, k_r\} \text{ if } i > j \\ \{k_l, \cdots, i, \cdots, k_r\} \text{ if } i = j \end{cases} \tag{1.91}$$

or,

- if $j > i$ then $\theta(i,j) = \{k_l, \cdots, i, \cdots, j, \cdots, I_r\}$, where k_l is the left-most infostation that belongs to a partial path from i through j.

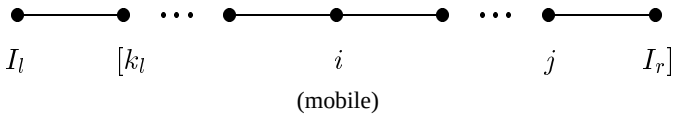

Figure 1.36 Illustration for Partial Paths from i through j

- if $j < i$ then $\theta(i,j) = \{I_l, \cdots, j, \cdots, i, \cdots, k_r\}$, where k_r is the right-most infostation that belongs to a partial path from i through j.

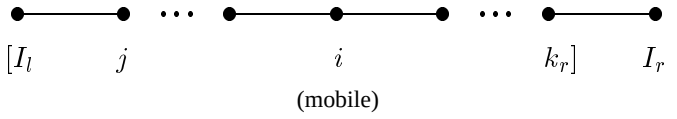

Figure 1.37 Illustration for Partial Paths from i through j)

- if j = i then $\theta(i,i) = \{k_l, \cdots, j, \cdots, i, \cdots, k_r\}$, where k_r is the right-most infostation that belongs to a partial path from i through i and k_l is the left-most infostation that belongs to a partial path from i through i.

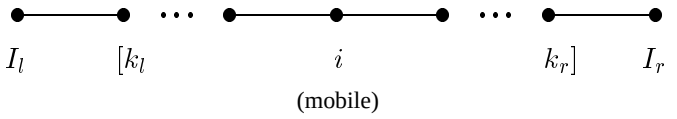

Figure 1.38 Illustration for Partial Paths from i through i

In other words, it is necessary to find the first infostation to the right (or left) to the mobile that belongs to a partial path from i through j.

4.3.4 The set $\nu(i,j)$. Once the set $\theta(i,j)$ is obtained, obtaining $\nu(i,j)$ is straightforward. If the importance of this set in the algorithm is not clear yet, it should be after this section.

Assume the mobile is at position i and all the segments that were not delivered to the mobile are already scheduled to the infostations. The algorithm has to decide what to schedule to a given infostation j.

Once we calculate the set of infostations that belong to partial paths, $\phi(i,j)$, the best solution is to schedule a segment that is *not* scheduled to any infostation in the set $\phi(i,j)$, or a segment that do not belong to $\mu(i,j)$. If a segment that belongs to $\mu(i,j)$ is scheduled to an infostation that belongs to $\phi(i,j)$ then there is a probability (different than zero) that this segment will be wasted if the mobile takes one of the partial paths. On the other hand, if the segment does not belong to $\mu(i,j)$ then this segment could be scheduled to any infostation in $\phi(i,j)$.

Assume $j > i$. As we mention before the set $\phi(i,j)$ may or may not include the the last infostation in the range, I_l. But we also do not want to copy a segment that is scheduled to I_l in j since there is a probability (different than zero) that this segment will be wasted. The same applies to infostation I_r, if $j < i$. For this reason we will work with the set $\theta(i,j)$ and $\nu(i,j)$.

In order to clarify the importance of these sets, please refer to the example shown in Figure 1.24. The file is divided in 10 segments, the mobile is at position 4 and picked one segment (P_2). The algorithm has to schedule new segments for the next step. Since segments P_5, P_7 and P_{10} are not scheduled, they can be scheduled to positions 2, 3 and 4, for example, as shown in Figure 1.39.

	?	P_7	P_5	P_{10}	?	?	?
step 1:	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.39 Example for a file with 10 segments: schedule of segments after first step

In order to decide what to schedule at the other infostations it is necessary to consider each infostation separately. Let us consider infostation 5. Starting at the user position (infostation 4) there are several partial paths from 4 through 5. The set

$$\theta(4,5) = \phi(4,5) = \{3,4,5,6,7\} \tag{1.92}$$

includes all the infostations in all partial paths. This set creates a box, as shown in Figure 1.40.

	?	P_7	$[P_5$	P_{10}	?	?	?]
step 1:	P_8	P_6	$[P_4$	X	P_9	P_1	$P_3]$
position:	1	2	[3	4	5	6	7]
				M			

Figure 1.40 Box created after calculating $\theta(4,5)$: all segments that are in the box should not be scheduled to position 5.

All segments that are already scheduled to infostations in the box, or were delivered to infostations in the box in previous steps, should not be scheduled to infostation 5. The set with these segments is given by

$$\nu(4,5) = \mu(4,5) = \{P_1, P_3, P_4, P_5, P_9\} \tag{1.93}$$

Thus, we could schedule segment P_8 to infostation 5, as shown in Figure 1.41.

	?	P_7	P_5	P_{10}	P_8	?	?
step 1:	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.41 Example for a file with 10 segments: schedule of segments after first step, segment P_8 copied in two infostations

It is clear from the figure that the segment P_8 would never be delivered twice to the mobile, since the file delivery would end before the mobile passes through both infostations, 5 and 1, no matter which path is taken.

The configuration shown in Figure 1.42 would also be possible, where segment P_6 is copied in two infostations. Note that in this case segment P_6 could be delivered twice if path $\{4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2\}$ is taken. But note that this would not increase the delivery delay in terms of number of infostations, since at infostation 2 only one segment need to be delivered (assuming that infostations 3, 4 and 5 always deliver new segments).

4.4 THE ALGORITHM

Let N be the total number of file segments. Let i be the infostation where the mobile is located at a given step. At every step the algorithm will do:

	$?$	P_7	P_5	P_{10}	P_6	$?$	$?$
step 1:	P_8	P_6	P_4	X	P_9	P_1	P_3
position:	1	2	3	4	5	6	7
				M			

Figure 1.42 Example for a file with 10 segments: a different configuration, segment P_6 copied in two infostations

1. Calculate boundaries I_l and I_r;
2. Obtain $\lambda(I_l, I_r)$
3. For each infostation $j \in [I_l, I_r]$ do:

- Find $\theta(i, j)$ and $\nu(i, j)$
- If $|\lambda(I_l, I_r)| < N$ then
  - schedule a segment $\{P_i \mid P_i \notin \lambda(I_l, I_r)\}$ to infostation j;
  - add the segment P_i to $\lambda(I_l, I_r)$
- Else
  - If $|\nu(i, j)| < N$ then
    - schedule a segment $\{P_i \mid P_i \notin \nu(i, j)\}$ to infostation j;
    - add the segment P_i to $\eta(j)$
  - If $|\nu(i, j)| = N$ then
    - schedule a segment $\{P_i \mid P_i \notin \eta(j)\}$ and:
      - If $j > i$ then schedule from the range $[j + 1, I_r]$, starting from the set $\eta(I_r)$.
      - If $j < i$ then schedule from the range $[I_l, j - 1]$, starting from the set $\eta(I_l)$.
      - If $j = i$ then schedule from the set $\eta(I_r)$ or $\eta(I_l)$.
      - add the segment to $\eta(j)$.

The algorithm loses its optimality if, at a given step, $|\lambda(I_l, I_r)| = N$ and, for some j, $|\nu(i, j)| = N$. At this point the choice of what to send to j is heuristic. Using the idea that the boundaries may decrease, we bring something from the boundaries since that infostation may be removed from the range considered. This choice is not proven to improve the performance, but it is a good heuristic choice. Other methods could be used.

4.5 AN EXAMPLE

In this section we will present an example of the algorithm application to a file that is divided in 10 segments. The number of infostations that have to be considered is nine (four on each side of the mobile). We assume that the mobile starts at position 5, and the parts $P_1, \ldots, P_{10}$ are scheduled at every step. Below is the output of the algorithm:

step 0: mobile starts at position 5:

step 0:	P_8	P_6	P_4	P_2	P_9	P_1	P_3	P_5	P_7
mobile:					M				
position:	1	2	3	4	5	6	7	8	9

step 1: mobile goes to position 4:
new boundaries are $I_l = 1, I_r = 7$;
P_1, P_3 and P_8 are repeated out of partial paths:

step 1:	P_1	P_3	P_5	P_8	P_{10}	P_7	P_8		
step 0:	P_8	P_6	P_4	X	P_9	P_1	P_3		
position:	1	2	3	4	5	6	7	8	9
mobile:				M					

step 2: mobile goes to position 3:
new boundaries are $I_l = 1, I_r = 6$;
nothing is scheduled to position 6 since all parts belong to $\nu(3, 6)$:

step 2:	P_{10}	P_7	P_{10}	P_3	P_6	0			
step 1:	P_1	P_3	X	P_8	P_{10}	P_7			
step 0:	P_8	P_6	X	X	P_9	P_1			
position:	1	2	3	4	5	6	7	8	9
mobile:			M						

step 3: mobile goes to position 4:
new boundaries are $I_l = 2, I_r = 6$;
nothing is scheduled to position 6 since all parts belong to $\nu(4, 6)$;
μ_4^5 contains all the segments and therefore P_1 is brought from position 6:

step 3:		P_1	P_9	P_7	P_1	0			
step 2:		P_7	P_{10}	X	P_6	0			
step 1:		P_3	X	X	P_{10}	P_7			
step 0:		P_6	X	X	P_9	0			
position:	1	2	3	4	5	6	7	8	9
mobile:				M					

step 4: mobile goes to position 3:
new boundaries are $I_l = 2, I_r = 5$;
nothing is scheduled to position 2 since all parts belong to $\nu(3, 2)$;
nothing is scheduled to position 5 since all parts belong to $\nu(3, 5)$;
μ_3^4 contains all the segments and therefore P_6 is brought from position 5;
P_1 is repeated in position 3 since it is not in $\nu(3, 3)$:

step 4:		0	P_1	P_6	0				
step 3:		P_1	X	P_7	P_1				
step 2:		P_7	X	X	0				
step 1:		P_3	X	X	P_{10}				
step 0:		P_6	X	X	P_9				
position:	1	2	3	4	5	6	7	8	9
mobile:			M						

step 5: mobile goes to position 2;
file transfer is complete.

4.6 PERFORMANCE OF THE ALGORITHM

The average number of file segments picked up by step s, $\bar{P}(s)$, was obtained through simulations. We performed 900 trials for each value of s, meaning that we have 900 independent random walks for each value of s. Assuming a large enough number of file segments that it is impossible for them all to be picked up by step s, we take s steps and count the total number of segments that are picked up by the mobile. This result is compared with the theoretical result given by Equation 1.31 and a simulation assuming an optimum algorithm. The results are shown in Figure 1.43 with confidence intervals about each point smaller than the symbol size. As can be seen, the three curves are hard to differentiate.

Because the number of segments picked even with the optimal algorithm depends on the path chosen, there is considerable variance in the number of segments picked up at any given step. Shown in Figure 1.44 is the corresponding standard deviation for our algorithm and for the optimum which again suggests the near-optimality of our algorithm.

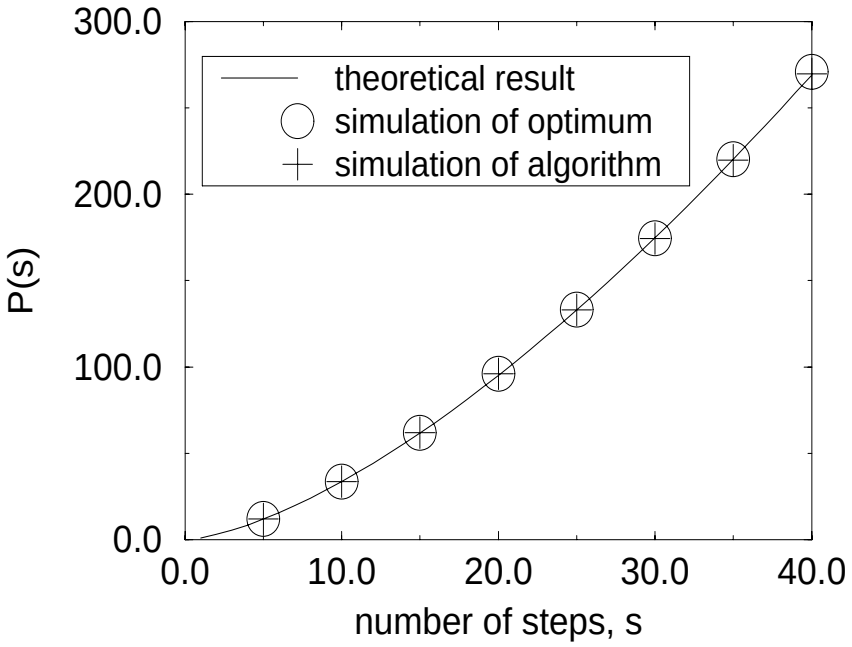

Figure 1.43 Comparison for the average number of file segments picked up in s steps, $\bar{P}(s)$, file size = 820 segments, 900 trials, 95% confidence interval, confidence intervals are smaller than a symbol size

In Figure 1.45 we provide the complementary CDF of the difference between the delay for an optimal algorithm applied to a file of size 200 pieces and the delay associated with our algorithm for the same random mobile user path. We did 1000 trials, and it can be seen that the heuristic algorithm fails by more than 3% only 16% of the time and by more than 10% only 0.4% of the time.

The excellent performance of the algorithm is in part due to the fact that we are transmitting in *all* infostations that will possibly be in the path. In the single user case this is not a problem since all the links can be used for one user. If we increase the number of users it may not be possible to coordinate all transmissions to all users if the boundaries overlap.

Therefore, to evaluate the algorithm efficiency we change the number of infostations to where the system transmits to. We consider the case where the system transmits to C infostations to the right and to the left of the mobile ($2C + 1$ total number of infostations).

Figure 1.46 shows the total number of file pieces transmitted before completion of file delivery, as a function of the file size, for different values of C.

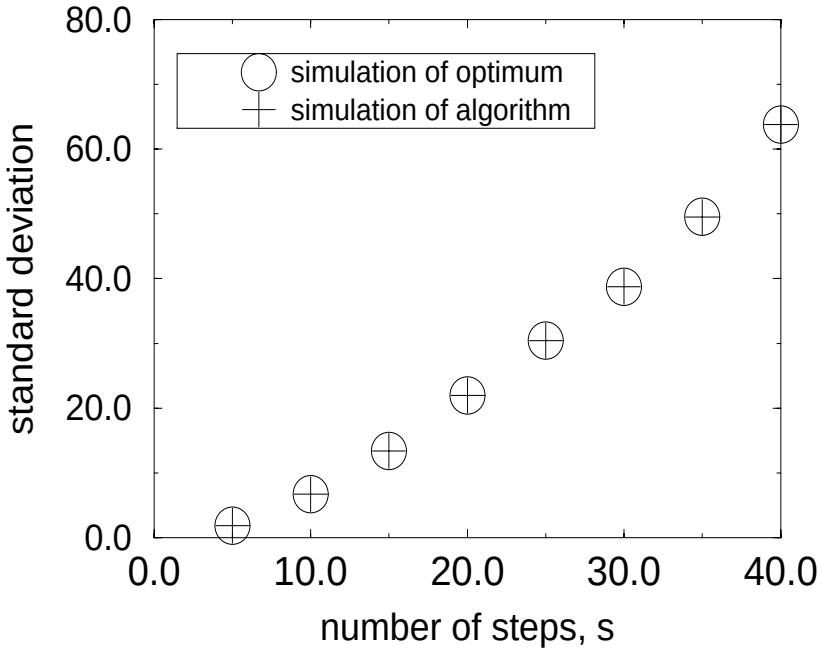

Figure 1.44 Standard Deviation of the number of file segments picked up in s steps for 900 trials

The total transmitted minus the file size represents the wastage of the system. Figure 1.47 shows the average delivery delay. As can be seen, reducing the value of C to 3, for example, increases considerably the efficiency and does not affect as much the delay performance. That shows that it may be possible to accommodate more users without affecting very much the system performance.

5. CONCLUSIONS AND WORK IN PROGRESS

In this work we considered the problem of delivering a file in system which features high rate discontinuous coverage. The collection of access points and the algorithms which support file delivery we call an Infostation System. Assuming that there are several infostations in the mobile path, the file is divided into segments and different segments can be transmitted to different infostations along the path.

The constant velocity case was studied and the most interesting result is that higher mobile velocity reduces delay if different data can be delivered to multiple infostations in parallel over the fixed network. Then a random walk mobility model was introduced with constant velocity but with randomly

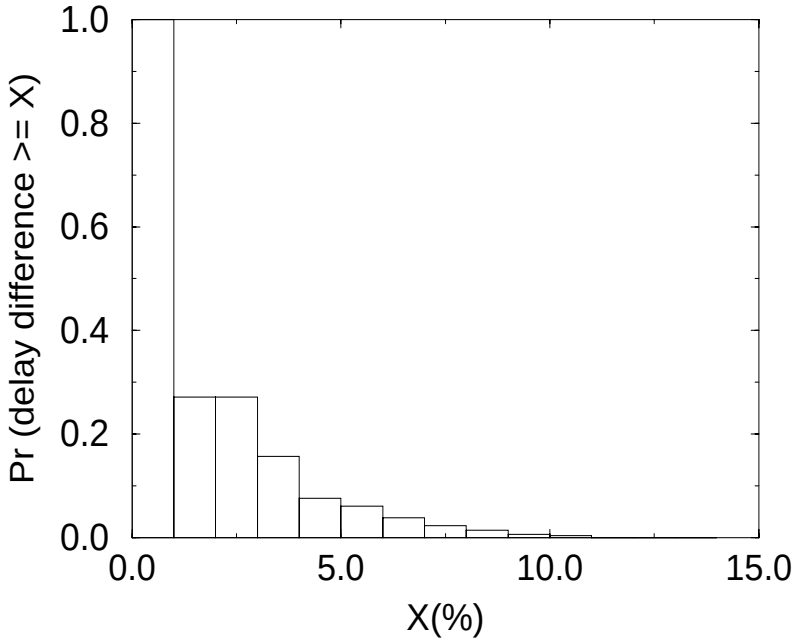

Figure 1.45 Complementary CDF of the difference between the delay for an optimal algorithm and our algorithm, file size = 200 segments, 1000 trials

chosen travel direction at each step. Results for bounds on the average number of file segments picked after a given number of steps for a general infostations topology were obtained. It was shown that the fewer infostations are revisited in a path, the larger the average number of segments obtained at each step.

An algorithm for the one-dimensional case was proposed. The algorithm simply tries to avoid repetitions of segments in places where the mobile is likely to visit along a path. The algorithm is not optimum in the sense that it does not achieve the absolute bound associated with foreknowledge of the user path, and it fails when it cannot avoid the repetitions. However, it is an open question whether this algorithm is indeed optimal among all algorithms which do not know the user path beforehand. Regardless, simulation results showed that its delay performance is extremely close to the known-path optimum.

Currently the authors are working on the scheduling problem for the multiple user case. This problem is much more complex than the single user case since each infostation has to decide not only which segments to transmit but also which user to serve.

Some other suggestions for future work are:

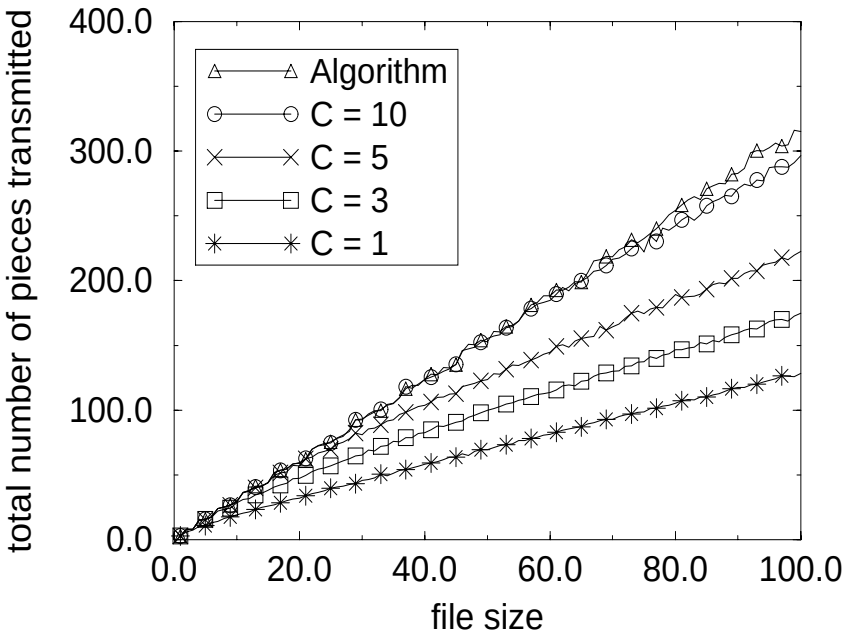

Figure 1.46 Total number of pieces transmitted, 100 trials

- We have assumed a constant velocity between infostations. It would be interesting to consider the situation where the time spent traveling between two infostations is a random variable. Yet another important factor is that in our analysis we assumed a *grid* scenario. When considering discontinuous coverage area, though, the system can be modeled as a complete graph, where every node represents the infostations and the weight of every edge is the transition probability between the two infostations.
- We assumed that the bottleneck of the system was the wired backbone and that the radio was always capable of transmitting all the segments to the mobile during the time the mobile is in the coverage area. It is necessary to consider imperfections over the wireless channel, such as errors and retransmissions, which may cause some file segments to be probabilistically missed at any given infostation.

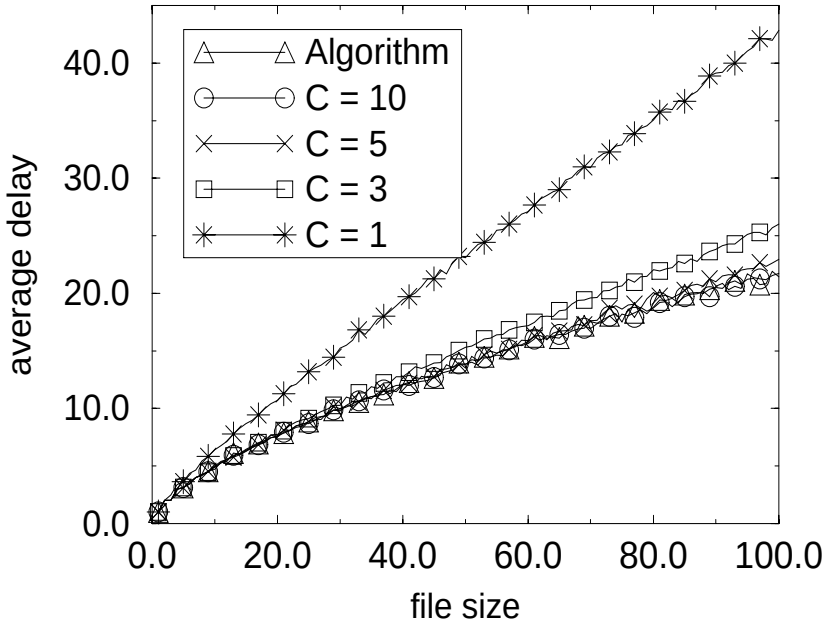

Figure 1.47 Average Delivery Delay, 100 trials

References

[1] R. D. Yates and N. B. Mandayam. Issues in wireless data. In *IEEE Signal Processing Magazine*, May 2000. To appear.

[2] R. S. Cheng and S. Verdu. Gaussian Multiacess Channels with ISI: Capacity Region and M ultiuser Water-Filling. *IEEE Transactions on Information Theory*, 39(3), May 1993.

[3] D. J. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates. INFOSTATIONS : A New System Model for Data and Messaging Services . In *Proceedings of IEEE VTC'97*, volume 2, pages 969–973, May 1997. Phoenix, AZ.

[4] J. Borras. *Capacity of an Infostation System*. PhD thesis, Rutgers University, January 2000.

[5] A. Goldsmith and P. Varaiya. Capacity of fading channels with channel side information. *IEEE Trans. Inform. Theory*, pages 1218–1230, Oct 1997.

[6] Apple, 1999. Apple Computer Inc., URL = http://www.apple.com/airport.

[7] P. H. Lewis. Not born to be wired. The New York Times, Circuits Section, November 25 1999.

[8] J. Borras and R. D. Yates. Infostations overlays in cellular systems. In *Proceedings of the Wireless Communications and Networking Conference, WCNC*, volume 1, pages 495–499, 1999.

[9] R. H. Frenkiel and T. Imielinski. Infostations: The joy of "many-time, many-where" communications. Technical Report TR-119, WINLAB, Rutgers, The State University of New Jersey, April 1996.

[10] G. Wu, C.-W. Chu, K. Wine, J. Evans, and R. Frenkiel. Winmac: A novel transmission protocol for infostations. *Vehicular Technology Conference*, 1999.

[11] H. Mao, G. Wu, C.-W. Chu, J. Evans, and M. Carggiano. Performance evaluation of radio link protocol for infostations. *Vehicular Technology Conference*, 1999.

[12] J. Irvine, D. Pesh, D. Robertson, and D. Girma. Efficient umts data service provision using infostations. *Vehicular Technology Conference*, 3:2119–2123, 1998.

[13] J. G. Evans. A low cost asymmetric radio for infostations. Technical Report TR-130, WINLAB, Rutgers, The State University of New Jersey, September 1996.

[14] T. Ye, H.-A. Jacobsen, and R. Katz. Mobile awareness in a wide area wireless network of info-stations. *ACM Mobicom*, pages 109–102, 1998. Dallas.

[15] J. Irvine and D. Pesh. Potential of dect terminal technology for providing low-cost wireless internet access through infostations. In *IEE Colloquium on UMTS Terminal and Software Radio*, pages 12/1–6, 1999.

[16] A. L. Iacono and C. Rose. Minimizing file delivery delay in an infostations system. Technical Report TR-167, WINLAB, Rutgers, The State University of New Jersey, August 1998.

[17] A.L. Iacono. *File Delivery Delay in and Infostations System*. PhD thesis, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, June 2000.

[18] A. L. Iacono and C. Rose. Bounds on file delivery delay in an infostations system. In *Proceedings of the IEEE Vehicular Technology Conference*, 2000. To appear.

[19] R. H. Frenkiel, B.R. Badrinath, J. Borras, and R. D. Yates. The infostations challenge: Balancing cost and ubiquity in delivering wireless data. Submitted to IEEE Personal Communications, 1999.

[20] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume I, Chapters IV & XIV*. Wiley, third edition, 1968.