Portender: Improving Mobile Applications Through Cellular Handoff Prediction

Anand Iyer, Vishnu Navda, Ramachandran Ramjee, Venkata N. Padmanabhan
Microsoft Research India

ABSTRACT

Handoffs in cellular data networks are often disruptive. The buffering of packets in the network at the time of handoff results in a delay spike. Furthermore, handoffs between base stations attached to different radio network controllers is often "hard", resulting in packet loss. The delay spike and packet loss degrades the performance of networked applications running on mobile devices.

In this paper, we argue that *predicting* impending handoffs is the key to mitigating the impact of handoffs. We start with a characterization of the macroscopic and microscopic properties of handoffs based on extensive measurements in multiple cellular networks across 3 cities in 2 countries. Based on the findings, we present *Portender*, a system for accurately predicting handoffs. Portender operates at two levels: it anticipates impending handoffs at a fine time granularity (a few seconds) based on signal information and it also estimates the frequency of handoffs over longer periods (e.g., a 30-minute drive along a particular route) based on history. In both cases, the prediction is performed by the mobile device without requiring cooperation of the network.

We have implemented Portender on Windows Phone 7. We demonstrate the accuracy of its predictions based on data from 950 km of drives encompassing 3300 handoffs in aggregate. We evaluate the utility of Portender's short-term predictions in improving TCP throughput and improving aggregate cell throughput, and discuss the use of Portender's long-term predictions as an advisory indication to set user expectations.

1. INTRODUCTION

Handoffs are intrinsic to cellular networks, wherein mobile users often cross cell boundaries. While much research and engineering has gone in to making handoffs as seamless as possible, handoffs in cellular networks remain disruptive for data services. For example, 3G networks are based on wideband CDMA (WCDMA), which supports soft handoffs, a make-before-break mechanism that should, in principle, eliminate any disruption during handoffs. However, soft handoffs entail the overhead of duplicate, concurrent transmissions by multiple base stations, so the use of this mechanism is confined to the performance-sensitive voice traffic and does not extend to data traffic such as that over HSDPA connections.

Given the above, the first issue we turn our attention to is a measurement-based characterization of handoffs and their impact on data traffic. Our measurement data is from 950 km of drives across 3 cities in 2 countries, and encompasses 3300 handoffs in aggregate. We characterize handoffs at two levels — macroscopic and microscopic. At the macroscopic level, we consider the spatial and temporal patterns of handoffs. We find that the handoff density along a route is stable but that there are significant differences across routes. At the microscopic level, we consider the low-level signal information that triggers handoffs. We find that handoffs happen over a wide range of signal strength values, *not* just when the signal is weak. *Our first contribution is this measurement-based study which, to our knowledge, is the first such characterization of handoffs in cellular networks.* Our study complements the many analytical and simulation-based studies of handoff in the literature.

The findings of our measurement study lead us to the central question: how well can we predict handoffs? *To this end, we present our second contribution,* Portender*, a system for accurately predicting handoffs based only on information available on mobile devices and without requiring the cooperation of the network.* Portender operates at two levels. First, it operates over long timescales to predict the frequency of handoffs based on past observations. For example, Portender might predict, based on past measurement data, that handoffs are likely to be much more frequent on a route that cuts through the core of a city than one heading towards the outskirts. Second, Portender operates over short timescales to predict impending handoff events, just seconds in advance of their occurence.

Being able to anticipate handoffs opens up the possibility of mitigating their impact by taking suitable steps in advance. We start with an experimental study of the impact of handoffs on TCP performance. We show the potential for and the limitations of employing the short-term predictions provided by Portender for mitigating the impact of handoffs on the performance of *individual* TCP connections. Based on this finding, we evaluate how the prediction provided by Portender could be used to regulate individual TCP connections for the larger good: to enable others nodes that are not undergoing handoff, and the cell in aggregate, to experience higher throughput. *We view the experimental evaluation of these alterative strategies for leveraging handoff predictions as the third contribution of our work.* Separately, we also discuss how both short-term and long-term handoff predictions could be used to alert users and to set user expectations in the context of VoIP and gaming applications.

The rest of the paper is organized as follows. In Section 2, we present a detailed measurement study, highlighting the different characteristics of handoffs. In Section 3, we present Portender, a system for predicting handoffs, over both the short-term (few seconds) as well as over the long-term (few minutes). In Section 4, we present several applications that benefit from Portender. In Section 5, we present related work before presenting our conclusions in Section 6.

2. MEASUREMENT-BASED CHARACTERIZATION OF HANDOFFS

We present a brief overview of the handoff procedures in 3G networks, describe our measurement methodology, and present various macroscopic and microscopic properties of handoffs.

2.1 Handoff Procedures in 3G

Handoffs in cellular networks are quite different from that of WiFi where the client mostly controls which AP to handoff to, and when to handoff. Moreover, handoffs in WiFi are hard, i.e., the old link is broken completely before the new one is established, which typically causes disruptions. In addition, handoffs in WiFi are quite limited to settings such as enterprises or campuses where large blanketed WiFi coverage may be available. In cellular networks, handoffs are controlled by the network operator. In addition, handoffs are a common occurance for mobile users in wide-area cellular networks.

The handoff procedure in WCDMA-based 3G networks varies depending on the state of the mobile client the type of networks available in the vicinity of the mobile client. When it is in idle mode, the client, typically referred to as user equipment (UE), can decide independently of the network which cell to camp on since it only needs to be able to receive broadcast messages and network-wide paging requests. However, when there is sufficient data traffic for an UE, the network allocates a dedicated channel (DCH) to the UE. In DCH mode, the network instructs the UE on which cell(s) the UE should monitor at any time. The UE is then required to measure the signal strength of the pilot transmission also referred to as received signal code power (RSCP) value from the specified cells. In addition it also measures the noise level in the carrier and computes the signal to noise ratio or ECN0 value. The UE reports these values back to the network. The UE can passively perform these measurements of cells in its neighborhood (i.e., the *monitored set*), by listening to the pilot signals sent on a common channel. These measurements are reported back to the network either periodically or in a triggered fashion whenever certain conditions are met (e.g., when a neighbouring cell is significantly stronger than the current cell of attachment).

The measurement reports are received by the basestation controller (BSC), or radio network controller (RNC) in WCDMA parlance. The RNC instructs the UE on which cells it should include in its *active set*, i.e., the set of cells from which the UE is actively receiving data. The RNC effects a handoff by having the UE change its active set, either by adding or by dropping a cell.

Note that the active set could include more than one cell, each of which the UE is attached to concurrently and receiving transmissions from. This typically happens during a handoff, when the UE is connected to both the old and the new cell for a while before attaching solely to the new cell. Such a handoff is referred to as a soft handoff and is facilitated by the reuse-1 pattern of frequency usage in CDMA networks. However, in cells that support HSDPA, which is a newer revision of WCDMA, only one cell is designated as a serving cell at any given time, even though multiple active cells may exist. Handoffs here imply a change in the serving cell, which results in a disruption during handoff.

Although, soft handoffs are expected to have no service disruptions, our experimental results indicate that handoffs are disruptive. Several channel parameters are often reconfigured at the physical and transport layers, after a radio link to a new cell is established. In addition, the connection between the UE and the Radio Resource Controller (RRC) node within the network (responsible for connection management), can sometimes get broken and needs to be reestablished after a handoff. Such disruptions often result in packet loss. Sometimes the "before" and "after" cells are on different frequencies, so such inter-frequency handoffs are necessarily hard handoffs in contrast to intra-frequency soft handoffs. Finally, because of the overhead entailed in duplicate, concurrent transmissions from multiple towers, the use of soft handoffs in WCDMA networks is typically confined to voice traffic and does *not* extended to a data service such as HSDPA.

2.2 Measuring Handoffs

To better understand handoff characteristics in cellular 3G networks, we carried out extensive measurements on multiple carrier networks during 950 km of

Operator	Location	Drive length	# Handoffs
Operator-1	City-A	218 kms	1002
Operator-2	City-A	500 kms	1470
Operator-2	City-B	78 kms	308
Operator-3	City-A	88 kms	416
Operator-4	City-C	64 kms	131

Table 1: Measurement dataset

drives carried out over a period of 4 months in four cities across two countries. Our datasets included 3300 instances of handoff. Table 1 smmarizes our datasets.

All our measurements were perfrormed solely on the client side, comprising an LG GW910 handset equipped with a 3G HSDPA radio and a 1GHz processor from Qualcomm, and running the Windows Phone 7 OS. A photo of our measurement setup is shown in Figure 1. We are able to pull out the pilot signal measurements that are recorded in the phone in real time using a proprietary Qualcomm diagnostic tool [14], which records signals samples every 20 ms and, in addition, provides a cummulative report of the measurements every second. The tool runs on a Windows 7 laptop, which communicates to the phone over a USB interface. Furthermore, we use field tools running on the phone itself to record the measurement reports sent by the phone to the network.

We can distinguish the pilots signals from different basestations through a unique code, referred to as the Primary Scrambling code (PSC) [6], that is embedded in each pilot. A handoff is detected whenever the PSC corresponding to the current attached cell changes. Each measurement is also tagged with the corresponding GPS coordinate.

During our experiments, we always had some data traffic on the cellular link ensuring that the UE is in DCH mode. We employed two workloads from a traffic source in the Internet to the mobile UE – TCP bulk transfer and UDP CBR traffic.

2.3 Macroscopic Analysis of Handoffs

We consider various macroscopic properties of 3G celular handoffs pertaining to the locations where handoffs occur, the spatial and temporal stability of handoff patterns, and the impact of direction of movement on the locations where handoffs occur.

Figure 2 shows the sequence of cells that a mobile is attached to during 11 drives in each direction along a 14 km route in City-A. These drives were performed during different times of the day and different days of the week. Each cell is shown with a unique colour. The figure shows that the location at which handoffs occur between a pair of cells, say A and B, is quite consistent when only the drives in the same direction of travel are considered. The point of handoff between two cells

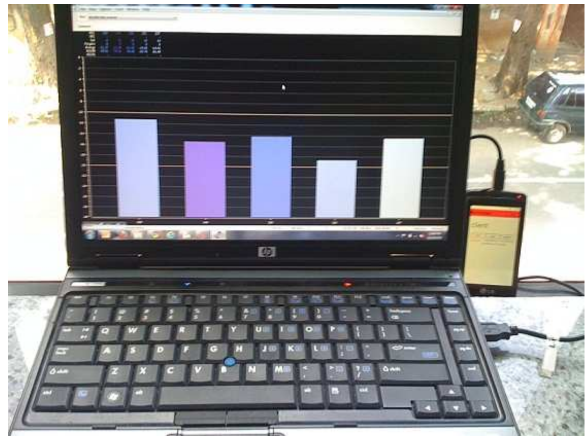

Figure 1: Measurement setup: instantaneous pilot signals measured on the phone is sent to the laptop for analysis via USB.

across multiple drives is spread over a distance of only 200 meters in 75% of the drives. However, the point of handoff between cells A and B along a drive in one direction is offset relative to that between cells B and A in the opposite direction. In prior work [15] it was hypothesized that this offset is due to the hysterisis built into the handoff process, which causes a mobile node to stick with its current cell until a neighbouring cell becomes sufficiently stronger. We elaborate on this in our detailed discussion of the dynamics of the handoff process in Section 3.

The spatial stability of the handoff location (modulo the the drive direction issue noted above) leads us to the next question of how the spatial distribution of handoff locations varies along different routes. Figure 3 shows the CDF of the inter-handoff distance along three routes in City-A and one route in City-C. The handoff locations are much more finely spaced along route 1, which cuts through the heart of the city, than along route 4, which leads to the airport on the outskirts of the city. The median inter-handoff distance for routes 1, 2, 3, and 4 is 50 m, 400 m, 170 m, and 500 m, respectively. This variation in the spatial density of handoffs arises from the corresponding variation in cell size to accomodate the differences in user density across different parts of the city.

To the extent handoffs are disruptive, a small inter-handoff distance implies a *potentially* high degree of disruption. However, the actual degree of disruption depends on the distribution of inter-handoff *time*. Figure 4 shows the corresponding CDFs for the three routes in City-A and one route in City-C. The median inter-handoff time is 10 s, 38 s, 17 s, and 24 s across the four routes. There is less variation in the distribution of inter-handoff time across the three routes than there

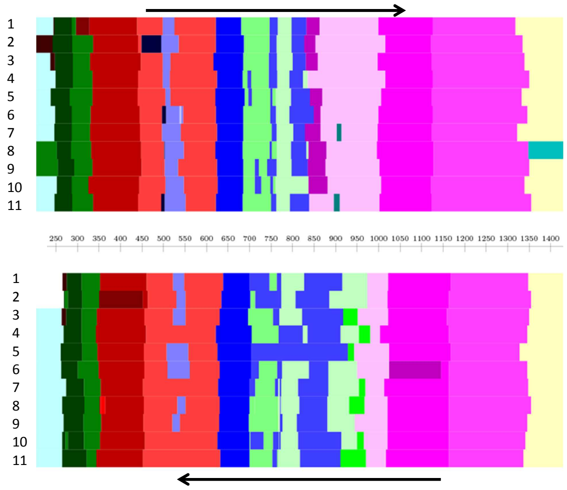

Figure 2: Handoffs during 11 drives in each direction along a 14 km route. The routes are shown stacked one on top of each other. Each color represents a unique basestation the handset is attached to at that location.

is in the distribution of inter-handoff distances. This is so because in our measurements, the drives through the dense parts of town, with a small inter-handoff distance, tended to have a relatively low speed, thus dilating the inter-handoff time. In other words, the inverse relationship between speed and cell density mitigates the frequency of disruption caused by handoffs. In some cases there is even an inversion in the ordering; for example, the median inter-handoff distance for routes 4 is larger than that for route 2 (500 m vs. 400 m), but the median inter-handoff time is smaller (24 s vs. 38 s). This is because route 4 is in city C, which is in a developed country with a lower population density (and hence lower tower density and larger cell size) and faster moving traffic than route 2 in city A, which is in a developing country with a higher population density and more congested traffic.

However, it is pertinent to note that unlike the inter-handoff distance, which is a function of the relatively static spatial layout of cells in the network[1], the inter-handoff time can vary sharply depending on user's mode of travel. For instance, a user riding a metro train is likely to be travelling at roughly the same speed across different parts of town, so the frequency of handoffs (and the attendant disruption) will tend to vary according to the spatial density of cells.

In summary, our macroscopic analysis of handoffs indicates that spatial distribution of handoffs along a route remains stable over time and that the distrubution can vary significantly across different routes even in the same city. In Section 3, we leverage these findings to perform coarse-grained prediction of handoff frequency in aid of applications.

[1]Even with techniques such as cell breathing, the spatial distribution typically only changes relatively slowly, on the timescale of hours.

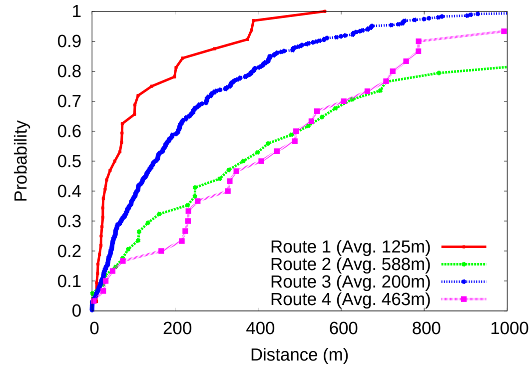

Figure 3: CDF plot of distance traveled between consecutive handoffs.

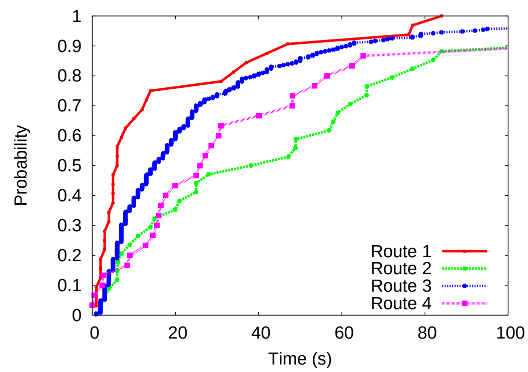

Figure 4: CDF plot of time between consecutive handoffs.

2.4 Microscopic Analysis of Handoffs

Next, we turn to a microscopic analysis of handoffs. Using the timeline of an actual handoff event as a reference, we show how the strength of the signal from the current and neighbouring cells varies before and after the handoff event. We also consider the impact of the handoff event on packet delay and loss experienced by ongoing traffic flows.

Figure 5 shows the timeline of a handoff event recorded in our experiments. We plot the strength of the pilot signal from the current cell and from the strongest neighbour. Note that we use a consistent line-style for the current cell and for the strongest neighbour. So when a handoff occurs, the line-style that depicted the signal strength for the previous base station is used for the new basetation, and likewise for the strongest neighbours before and after the handoff. We do likewise for the signal-to-noise ratio (ECNO) curves for the current base station and the strongest neighbour.

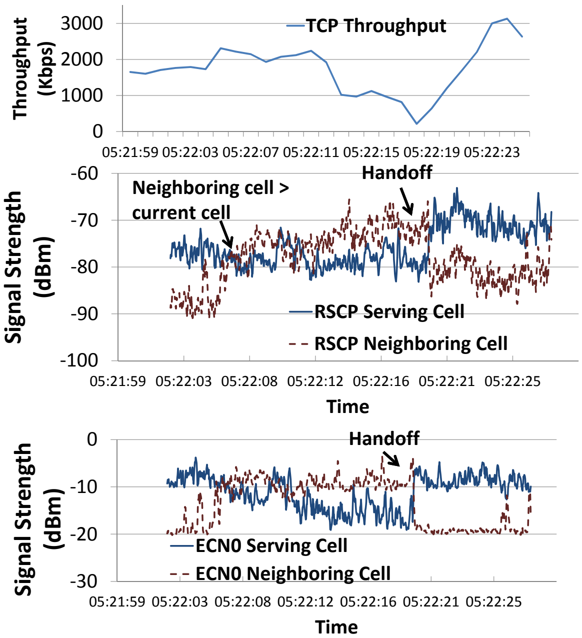

Figure 5: Timeline of signal strength and throughput values during a handoff.

In Figure 5, we also show the throughput of a TCP bulk download being received by the mobile node during the handoff event. We observe a significant drop in TCP throughput leading up to the handoff event. This is explained by the drop in the ECNO of the serving cell because of increasing interference from the neighbouring cells as the point of handoff is approached.

Conventional wisdom and also the example timeline in Figure 5 above suggests that a handoff event is triggered by a combination of (a) a weak signal from the current cell, and (b) a strong signal from a neighbouring cell. To evaluate this hypothesis, we plot in Figure 6 the signal strengh corresponding to the tower of attachment before and after a handoff event.

Contrary to our hypothesis that handoff is trigged by a weak signal, we find that there is a wide range in the absolute strength of the "before" signal at the time that a handoff is triggered. A handoff is just as likely when the signal is as strong as -60 dBm as when it is as weak as -110 dBm. Even when we consider the signal-to-noise ratio (not shown), there is a wide range.

However, as hypothesized, the distribution of "after" signal strength in Figure 6 is shifted to the right (i.e., towards stronger signals) compared to the distribution of the "before" signal strength. For instance, the median "after" signal strength is 6 dB higher than the median "before" signal strength. Nevertheless, when we compare the "before" and "after" signal strengths for individual handoff events (Figure 7), we find that the "after" signal strength is lower in about 25% of the cases.

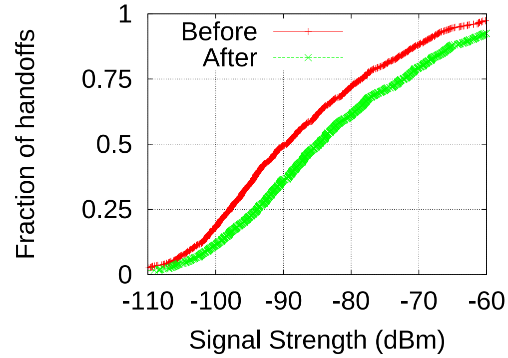

Figure 6: CDF plot of average RSCP value seen before and after a handoff.

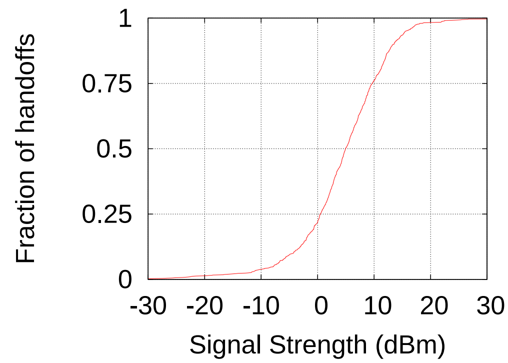

Figure 7: CDF plot of difference of average RSCP value seen after a handoff and before a handoff. Significant fraction of the handoffs have negative differences.

Our study of handoff prediction in Section 3 sheds more light on the relationship between signal strength and the handoff event.

Next, we turn to the impact of a handoff event on application traffic. At its core, a handoff event could include a period when no packets are delivered to the mobile node because there is not a working radio link. In addition, packets in flight can get lost when the radio links are being reconfigured. Figure 5 depicts how the TCP throughput starts to drop close to a handoff.

3. HANDOFF PREDICTION

From the previous section it is clear that disruptions to network traffic due to hard handovers in cellular networks could degrade application performance. In this section, we present Portender, a system to accurately predict whether a handoff is likely to occur in the next few seconds. In addition, Portender can also provide long-term predictions for handoffs that occur minutes into the future, although with a lower accuracy. We will first briefly describe why handoff prediction is useful, and why is it a challenging problem. Next, we evaluate different features that can be used for handoff prediction. Last, we will look at the design of Portender, which combines the different features to provide both short-term and long-term prediction of handoffs.

3.1 Why Handoff Prediction?

The prediction hints provided by Portender can be used by mobile applications to take necessary steps to handle one or more impending disruptions in the future. For example, Portender's short-term prediction can be used by a mobile application which may choose not to communicate during a handoff in order to avoid packet losses and unnecessary retransmissions. Such adaptations need not be just limited to the client-side application, but any host or server in the network that is communicating with the mobile could benefit from such an advance notice. As an example application, in Section 4, we describe how the throughput of a TCP bulk transfer can be improved using the short-term predictions provided by Portender. The long-term prediction of Portender can be useful in other applications. For example, a VoIP application, upon discovering that frequent handoffs are likely (for example, because the device is shortly expected to travel through a dense region of small cells), we could notify the user of the impending disruption. The VoIP user may then choose to either postpone an important call.

3.2 Why Handoff Prediction is hard?

Unlike in WiFi networks, predicting handoffs in cellular networks is challenging mainly because the handoff decisions are controlled by the network rather than by the client. When a user is moving while a data session is active, the handset is continuously monitoring the received power of the pilot signals that are periodically broadcast by the basestations. If signal strength of one of the neighboring basestation is found to be better than that of the currently attached one for a certain period of of time, a measurement report is sent to the Radio-Network controller (RNC), a node in the operator's network that is responsible for taking handoff decisions. If the handoff criteria set by the operator are met, the RNC issues a reconfiguration command to the mobile node to handover to the new basestation. Packets buffered in the old basestation of attachment may be flushed out during reconfiguration. Although, it is possible to get access to the pilot signal measurements on the handset, the criteria used to determine whether or not a handoff is needed is known only to the operator. Moreover, our measurements reveal that the exact criteria could vary across operators. Thus, one needs to reverse engineer the decision process to be able to effectively predict handoffs from the client side.

3.3 Features for Handoff Prediction

In Portender we infer the network's handoff decision-making process at the client side. Based on the handoff evaluation criteria described in the 3G network standards, we focus on three specific features of relevance to predicting a handoff: the SNR value (also referred to as EC/N0 value) of the current attached basestation, the SNR difference, δ, between the current base station and the strongest neighbor, and past information of handoff locations. Intutively, when the signal strength of the current attached basestation is low, one would expect a client to handoff to another nearby tower, if any. However, as indicated in Figure 6, handoffs are not confined to periods of weak signal. Therefore, in addition, we consider the signal strength difference, with the expectation that the signal strength of the neighbor is expected to be higher than the current basestation for a handoff to be triggered. However, as indicated in Figure 7, this is not always the case either. So we also factor in the history of handoff locations. Previous work [15] has shown that signal strength at any location along a path is fairly stable with less than 5 dB variation over time. Based on this observation, one would expect handoffs to also also be triggered at around the same locations along a path.

Given multiple drives corresponding to the same route, the question we ask is do handoffs occur at the same locations along the route? Are there patterns that could be leveraged to predict handoffs for the same route at a later point in time? Figure 2 shows a plot of the locations where the client handovers to different basestations along a 14 km drive. We notice that across the 11 runs that were carried out on different days and at different times of the day, for the majority of the basestations, the location of handoffs are within 250 meters of each other. This granularity of predictions is suitable for long-term course grained predictions to estimate the overall frequency of predictions along a certain route. However, location alone is not suitable for predicting when the next handoff is likely to occur to within a few seconds into the future.

Next, we look at how ECN0 of the current basestation and the presence of a stronger neighbor can impact handoffs. The UE sends measurement reports to the network with values of RSCP and ECN0 for the current cell and the neighboring cells in the monitored set. In our measurement setup, we are able to intercept these reports that are sent to the cellular network by the mobile radio. In Figure 9, we plot the ECN0 value for the current cell, and the difference δ with the strongest neighbor, both of which are obtained from the measurement reports. We classify the points into two groups – "handoff" and "no-handoff" depending on whether there was any handoff within few seconds of the transmission of a measurement report. Clearly, there exists

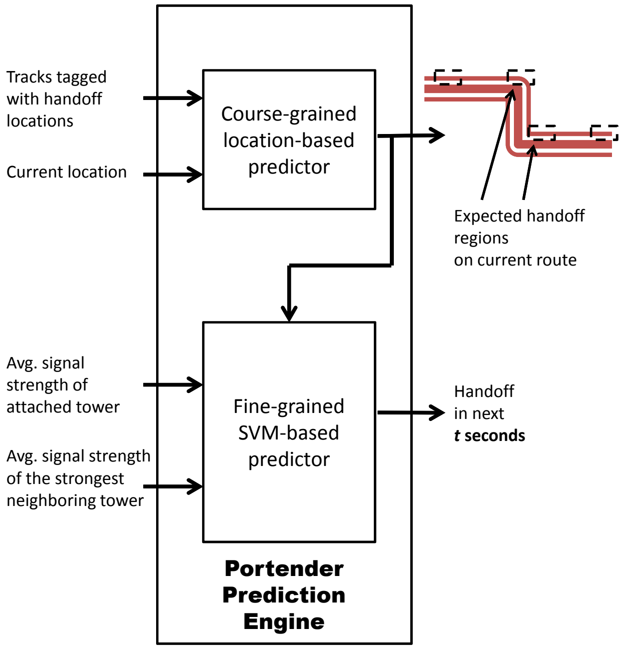

Figure 8: Portender prediction engine

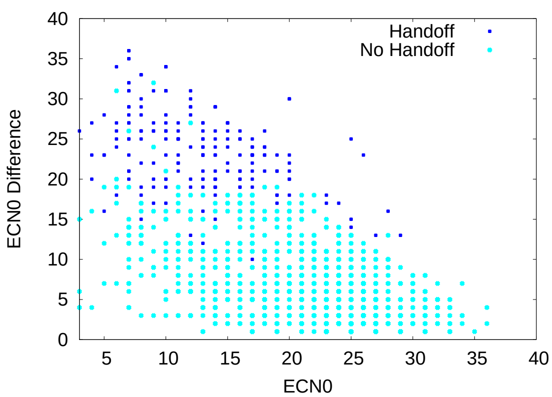

Figure 9: Signal strength of the attached basestation vs. δ value (difference between the attached basestation and the strongest neighbor).

a strong correlation between these two parameters and occurance of handoffs. We train a simple linear kernel SVM based classifier that computes thresholds using the two features – ECN0 and ECN0 difference to distinguish the two classes with the largest separation for datasets from different operator networks.

Thus, the Portender system uses the three features, namely location, ECN0, and δ, in combination to achieve high prediction accuracy, while keeping mis-predictions low. Figure 8 shows a block diagram of the prediction engine. We employ a fine-grained predictor, which intercepts measurement reports sent by the mobile to the RNC at run-time and uses the classifier to predict whether a handoff is expected in the next few seconds. In addition, we use a simple course-grained location-based predictor which uses prior tracks of handoffs to

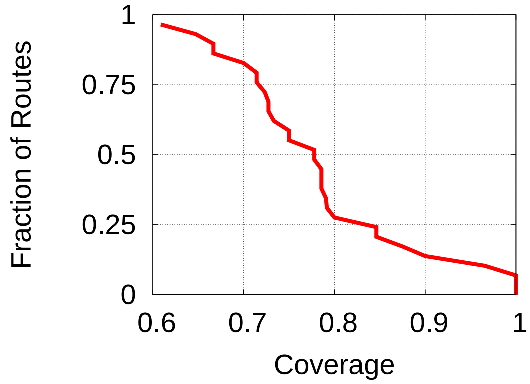

Figure 10: CDF of prediction coverage of Portender on different routes

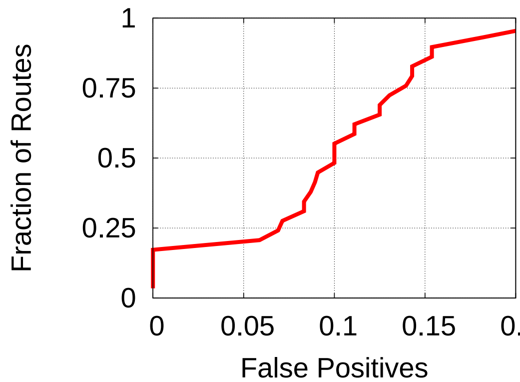

Figure 11: CDF of false positive ratio on different routes

fortell the regions where a cluster of handoffs can be expected for a given route. Such clustered handoffs could occur, say, at the edge of multiple cells, and result in severe impact on applications.

3.4 Handoff Prediction Evaluation

We now present results from our evaluation of the Portender prediction algorithm. We use data collected from multiple routes for evaluation, which include a total of 1577 handoffs over a distance of 320 km and comprises data services from multiple different operators in two countries. The training data set was based on a different and small subset of routes from one operator. Overall, Portender provides a coverage (i.e., the complement of the false negative rate) of 81% and a false positive rate of 10%.

Since handoff prediction accuracy varies significantly over different routes, we compute prediction coverage and false positives rate separately for each route. Figure 10 shows the CDF of Portender's prediction coverage and Figure 11 shows the CDF of false positives rate for the evaluated routes. From the results, we see

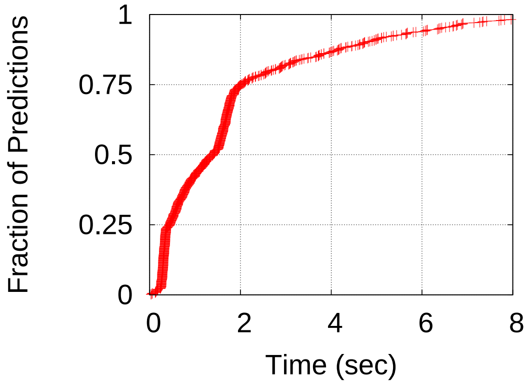

Figure 12: CDF of time between prediction and actual handoff

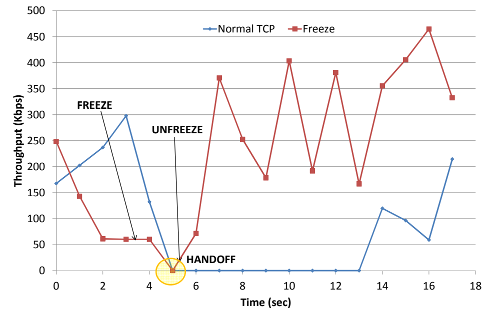

Figure 13: TCP throughput with and without Freeze-Unfreeze.

that Portender has a coverage of 78% for the median route and that the minimum coverage achieved by Portender (i.e., coverage corresponding to the worst route) is still over 60%. At the same time, the false positive rate ranges between 0-20%, with half the routes having a false positive rate of 10% or lower.

It turns out that there is a delay between when a particular measurement report is sent, and when the network actually instructs the mobile to handoff to a new cell, and this delay is not constant across different handoffs. We believe that this delay is both due to hysterisis imposed by the radio network controller (RNC) in the network and the processing delays in the RNC, which is the ultimate arbiter of the handoff decision. Figure 12 depicts shows the CDF of time elapsed between Portender's predictions and the actual handoffs. It is important to see that the mean delay is 1.8 seconds and variance is 1.9 seconds. As we shall see in Section 4, this variable lag of up to a few seconds between the handoff event and Portender's prediction has important implications for leveraging Portender's predictions, for instance, to improve TCP performance during handoffs.

4. APPLICATIONS

In this section, we examine the benefits of using Portender in three situations, namely, improving an individual connection's TCP throughput, improving a cell's aggregate TCP throughput and providing advance notifications to VoIP users.

4.1 Improving TCP throughput

In this section, we look at how we can improve an individual node's TCP throughput using Portender's handoff prediction engine. As discussed earlier (Figure 5), TCP throughtput often drops sharply due to packet losses during a handoff. If a significant number of packets is lost, the TCP sender might time out, resulting in an interruption of a few seconds before the sender initiates retransmission of the lost packets. This problem has been well-studied in the literature [4, 7] and a simple client-side solution is TCP Freeze [7]. In TCP Freeze, the receiver sends an ack with the receiver-advertised window set to zero. The TCP sender then freezes its congestion window. The sender resumes transmission at the same congestion window when the receiver unfreezes, i.e., starts advertizing a non-zero window. While TCP Freeze has been evaluated in simulation in the literature where it is assumed that handoffs can be somehow predicted, we are not aware of any prior work that has evaluated this mechanism *experimentally*, in conjunction with an actual handoff prediction scheme and has identified its benefits and limitations, as we do here.

We used Portender's handoff prediction to trigger the TCP receiver on the mobile node to send out an ack with zero receiver-advertised window in advance of an impending handoff. As soon as the handoff is completed (or after more than a threshold duration has elapsed without a handoff occuring, say, because Portender's prediction was incorrect), the receiver unfreezes by sending out an ack restoring the receiver-advertised window, and then the flow of data on the connection resumes. Fig 13 shows the throughput of a TCP connection during a handoff with the TCP Freeze mechanism in use and one without it in use. From the figure it is clear that the freeze prior to the handoff[2] helps avoid the timeout that would have otherwise been experienced by the TCP connection and enables the connection to quickly ramp back up to the throughput it was enjoying prior to the handoff event.

While the above example shows the effectiveness of TCP Freeze in a particular instance of handoff, we found that applying TCP Freeze over a long drive with multiple handoffs did not improve an individual TCP connec-

[2]Note that the throughput does not immediately go to zero after a freeze since packets that are buffered in the network are being drained.

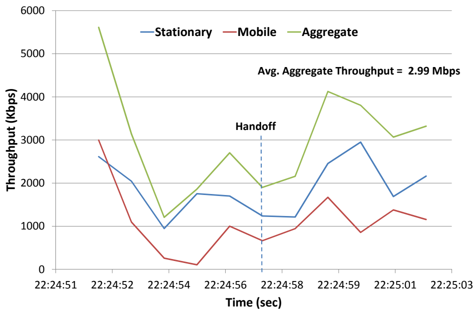

Figure 14: Aggregate network throughput of two tcp flows during handoff.

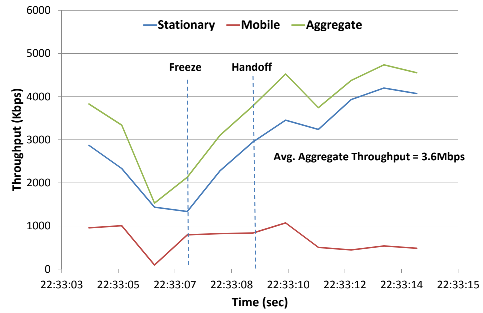

Figure 15: Aggregate network capacity with freeze-unfreeze.

tion's throughput significantly. In fact, in many cases, the connection's throughout with TCP Freeze was lower than it would have otherwise been. This was because whereas handoffs resulting in severe packet loss and hence TCP connection timeouts (as in the above example) are the ones that TCP Freeze can provide the maximum gains for, not all handoffs are so severe. Often, a handoff results in only a single packet loss or no loss at all, meaning there is no TCP timeout. In such instances, freezing the TCP connection a few seconds in advance of the handoff results in loss of throughput compared to the unmodified operation of the TCP connection. Thus, while prior studies based on *simulations* have reported significant gains from employing TCP Freeze [7], our *experimental* findings show that these gains are limited to the specific handoff events where there is severe packet loss. Absent a method to predict the severity of a handoff event, TCP throughout of an *individual* connection is likely to suffer rather than benefit from the application of TCP Freeze.

Although we are unable to consistently improve the throughput of an *individual* TCP connection, applying TCP Freeze or otherwise regulating the traffic to a node with an impending handoff can still be beneficial at a *global* level, i.e., to the nodes in the network in aggregate. We examine this opportunity next.

4.2 Improving aggregate TCP throughput

While the use of TCP Freeze a few seconds prior to handoff may not always help an individual connections throughput, it turns out that TCP Freeze during a handoff can have a positive impact on other ongoing connections in the cell and, thus, help improve aggregate cell capacity. Let us see how this is possible through an example.

The EVDO Rev A 3G standard [3] supports a range of discrete downlink data rates, from 38.4 Kbps to 3072 Kbps, based on the signal strength of the pilot signal at the receiver. Consider the case of two users — user A at the cell's periphery, capable of receiving data at 38.4 Kbps, and user B at the center, capable of receiving at 3072 Kbps. Assume that the users are continuously receiving data. The 3G base station employs proportional fair scheduling that provides equal time access to the two users and, thus, results in an aggregate throughput of 1728Kbps with user A and B receiving 19.2Kbps each and user C receiving 1536Kbps. Now, suppose user A's TCP connection freezes for a few seconds just before the handoff. The base station can then fully serve user B at 3072 Kbps, resulting in a doubling of aggregate capacity. Similarly, user A may benefit another time when a different device is ready to handoff. Thus, if all devices freeze their TCP communication for a few seconds prior to handoff, aggregate cell capacity may improve, providing benefit to the user population as a whole.

It is important to make a distinction between the application of TCP Freeze, as outlined above, and the normal operation of proportional fair scheduling. With TCP Freeze employed as outlined above, a mobile node is making the conscious decision to give up airtime for a short period coinciding with the well-defined handoff event, a period during which the node would likely receive little or no throughput anyway. In contrast, by itself, proportional fair scheduling would still devote (and waste) airtime to the node in question.

While the above hypothetical example illustrates the potential aggregate benefit of TCP Freeze prior to a handoff, we were also able to experimentally validate these gains in our experiments. We placed one device performing a large file download in a static location where the base station's received signal strength was high. We then took a second device that was also attached to the same base station and had it also perform a large file download while on a mobile trajectory until the device handed off to another base station. The mobile device was running Portender and each time Portender predicted a handoff, it would freeze its TCP connection and then unfreeze it as soon as the handoff was complete. We performed this experiment late in the night, when we expected that the network would be lightly loaded, with little or no background activ-

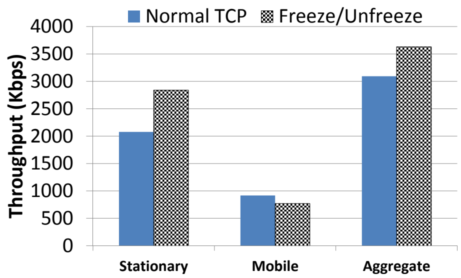

Figure 16: Throughput of a stationary node and mobile node for TCP and TCP freeze-unfreeze.

ity that might interfere with our controlled experiment. We repeated the experiment 25 times.

Figure 14 shows a timeline of TCP throughput during one particular handoff for two devices, one static and one mobile, and the aggregate throughput when TCP Freeze was not employed, while Figure 15 shows the same metrics when TCP Freeze was executed based on the prediction by Portender. The average aggregate throughput in the former case is 2.99 Mbps, and it increases to 3.6 Mbps when TCP Freeze is used based on Portender's prediction, for a gain of 20%.

Figure 16 shows the average throughput over 5 seconds before a handoff over the entire experiment that included 25 handoffs in total. From the figure, we see that while the mobile node suffers a small drop in throughput when using TCP Freeze, this drop is more than compensated by the increase in throughput for the stationary node. Thus, the aggregate throughput of the network is improved on average by around 20% using TCP Freeze.

Thus, given that signal quality at the edge of a cell prior to handoff is poor and data rates while transmitting to a device at the cell-edge is significantly lower compared to data rates when transmitting to a device with higher signal quality, the consistently help increase the aggregate cell throughput.

4.3 User Notifications

Finally, we discuss applications of Portender's functionality of handoff prediction over long timescales. As a device moves quickly through a dense region of small cells or cuts through the "edge" of several cells, it will likely experience frequent handoffs and the consequent disruption to applications and user experience.

Based on prior handoff measurements (either by the same user or crowd-sourced measurements from other users), Portender could provide a simple notification service that alerts the user approaching such a region of the impending quality and duration of disruptions; the user may then choose to postpone or cut-short a VoIP call or gaming session. Figure 17 shows the Mean

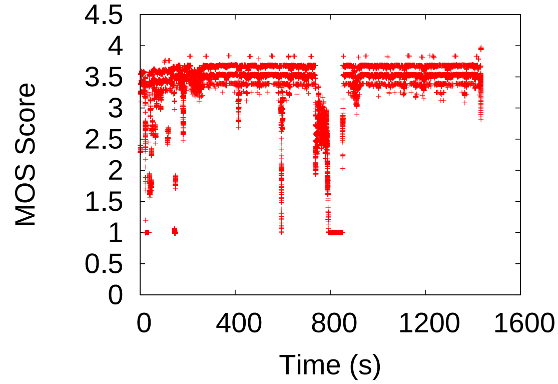

Figure 17: VoIP Mean Opinion Score (MOS) over time.

Opinion Score (MOS) over time for a VoIP call along a particular drive. From the figure, one can see that disruption due to handoffs can result in sharp dips in the MOS scores. Some of these handoffs can be especially detrimental to a VoIP call (e.g., the severe disruption seen around time 800s) and alerting users of such impending disruptions is one possible use of Portender's long-term handoff prediction.

Furthermore, if handoff locations and regions of poor network connectivity are integrated with a mapping service, users will then be able to choose routes that have better network coverage and fewer disruptions due to handoffs. We plan to further explore this application of Portender in future work.

5. RELATED WORK

Related work can be divided into two categories, work on mobility/handoff prediction and work on mitigating the impact of handoffs. We discuss each of these below.

5.1 Mobility/Handoff prediction

Mobility prediction is a well-studied area in the literature [1, 2, 8, 10, 13, 16, 19]. Many of these papers are targeted towards a centralized model, such as that of a cellular operator, tracking movement of users in order to provision resources appropriately [1, 2, 8, 13]. One of the early papers on motion prediction for anticipating handoffs was [10] where a user's movement history is used to anticipate the change of location of a user. Thus, services can be pre-configured at the new location, thereby eliminating any handover delay.

Authors in [19] derive a mobility prediction model from compression techniques and uses this to predict the time/location (cell) to which a mobile will handoff in order to reserve bandwidth and reduce handoff call dropping probabilities. Using traces from a WiFi network, authors in [16] show that simple low-order Markov predictors with a fall-back mechanism work as well as complex predictors such as those based on compression techniques.

Unlike the above papers, Portender is aimed at using current and neighboring tower signaling information of cellular networks to be able to predict handoffs at a fine-grain timescale (few seconds) and uses a profile of past history of handoffs at various locations to predict handoffs at coarse-grain timescales (few minutes).

5.2 Mitigating impact of handoff

Many papers have looked at the problem of reserving resources at the destination in order to provide seamless handover experience for applications [9, 17, 19]. In [17], the authors use handoff prediction so that network can reserve resources only in select neighboring cells, thereby avoiding unnecessary reservation. In [9], the authors show that roughly accurate handoff-time and location predictions, even with modest prediction accuracy, can improve bandwidth reservation schemes for VOIP apps

Negative impact due to disruption/handoff on TCP is a well-known problem [5] and several solutions for mitigating the negative impact of handoff on TCP such as M-TCP [4], TCP-Freeze [7], TCP-HO [18], ACK Pacing [11] have been proposed. However, all these proposals assume that handoff can be predicted at the mobile and are evaluated through simulations/emulations of the handoff or disruption event. In this paper, we implement TCP-Freeze in conjunction with a handoff prediction system on a mobile smartphone and show through real experiments in the field that the performance of TCP can be improved during handoffs using these techniques.

Breadcrumbs [12] is a system that couples a second-order mobility model with a WiFi AP performance database to predict future connectivity. Applications can use the predicted connectivity to appropriately schedule communication, thereby improving performance and saving energy. Bartendr [15] is another system that uses predicted mobility and future signal strength values to preferentially schedule communication when signal strength is high, thereby reducing energy consumption due to mobile communication. Complementary to these systems, Portender focuses on handoff prediction over short and long timescales in order to support better adaptation of applications to disruption.

6. CONCLUSION

The paper presents a detailed measurements study of cellular handoff characteristics from an extensive dataset collected in 3 different cities on 4 different network operators. In addition, we show how handoffs impact application performance. We have implemented and evaluated a prediction algorithm that can operate in both short timescales as well as in long timescales. We finally show how applications can make use of these predictions.

7. REFERENCES

[1] I. F. Akyildiz and W. Wang. The predictive user mobility profile framework for wireless multimedia networks. *IEEE/ACM Transactions on Networking (TON)*, 12(6):1021–1035, 2004.

[2] A. Bhattacharya and S. K. Das. Lezi-update: an information-theoretic approach to track mobile users in pcs networks. In *Mobicom*, 1999.

[3] N. Bhushan, C. Lott, P. Black, R. Attar, Y.-C. Jou, M. Fan, D. Ghosh, and J. Au. Cdma2000 1ev-do revision a: a physical layer and mac layer overview. *Communications Magazine, IEEE*, 2006.

[4] K. Brown and S. Singh. M-tcp: Tcp for mobile cellular networks. *ACM CCR*, 1997.

[5] R. Caceres and L. Iftode. Improving the performance of reliable transport protocols in mobile computing environments. *IEEE JSAC Special Issue on Mobile Computing*, 1994.

[6] Common Pilot Channel - Wikipedia. http://en.wikipedia.org/wiki/Common_pilot_channel.

[7] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta. Freeze-tcp: A true end-to-end tcp enhancement mechanism for mobile environments. In *Infocom*, 2000.

[8] B. Liang and Z. J. Haas. Predictive distance-based mobility management for multidimensional pcs networks. *IEEE/ACM Transactions on Networking (TON)*, 11(5):718–732, 2003.

[9] Libo Song and Udayan Deshpande and Ulas C. Kozat and David Kotz and Ravi Jain. Predictability of wlan mobility and its effects on bandwidth provisioning. In *Infocom*, 2006.

[10] G. Liu and G. Maguire. A class of mobile motion prediction algorithms for wireless mobile computing and communications. *ACM/Baltzer Mobile Networks and Applications (MONET)*, 1(2):113–121, 1996.

[11] Y. Matsushita, T. Matsuda, and M. Yamamoto. Tcp congestion control with ack-pacing for vertical handoff. In *WCNC*, 2005.

[12] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom*, 2008.

[13] P. N. Pathirana, A. V. Savkin, and S. Jha. Mobility modelling and trajectory prediction for cellular networks with mobile base stations. In *MobiHoc*, 2003.

[14] Qualcomm QxDM Professional tool. http://www.qualcomm.com/products_services/testing_diagnostics/qxdm_pro.html.

[15] A. Schulman, V. Navda, R. Ramjee, N. Spring,

P. Deshpande, C. Grunewald, K. Jain, and V. N. Padmanabhan. Bartendr: A practical approach to energy-aware cellular data scheduling. In *Mobicom*, 2010.

[16] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *IEEE Transactions on Mobile Computing (TMC)*, 5(12):1633–1649, 2006.

[17] W. Su and M. Gerla. Bandwidth allocation strategies for wireless atm networks using predictive reservation. In *Globecom*, 1998.

[18] X. Wu, M. C. Chan, and A. L. Ananda. Tcp handoff: A practical tcp enhancement for heterogeneous mobile environments. In *ICC*, 2007.

[19] F. Yu and V. C. M. Leung. Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks. *Computer Networks*, 38(5):577–589, 2002.