

Using Predictive Triggers to Improve Handover Performance in Mixed Networks

Huaiyu Liu, Christian Maciocco, and Vijay Kesavan

Communication Technology Lab, Corporate Technology Group, Intel, Hillsboro, OR 97124
{Huaiyu.Liu, Christian.Maciocco, Vijay.S.Kesavan}@intel.com

Abstract. End-users of multi-radio devices expect ubiquitous connectivity anytime and anywhere across heterogeneous networks. A minimal service disruption time while roaming across networks is key for successful deployments. Today's roaming decisions are reactive and traditionally based on thresholds for signal strength, missed beacons, or other properties, leading to undesirable handover delays. In this paper we propose a novel and energy efficient approach to predict when a device needs to take proactive actions to perform either a horizontal or vertical handover. The significant improvements our solution achieved in reducing service discontinuity time are demonstrated by applying our algorithms on WiFi and WiMax networks, in both enterprise and city-wide wireless environments.

1 Introduction

With the introduction of multi-radio devices and the deployment of multiple access networks (WiFi, WiMax, GSM, CDMA, etc), referred as Mixed Networks, end-users are expecting ubiquitous connectivity for their mobile devices and expect these devices to roam seamlessly across networks with no service disruptions while using demanding applications. Supporting seamless mobility across networks is a challenging task for both the clients and the networks.

On the client side, the handover process can be classified into three steps, i) “when and why” should the device transition to a new network, ii) “where” should the device transition to, and iii) “how” should the device transition between networks? The “when and why” step corresponds to what we describe as the triggering process when the mobile device receives an indication that it should look to operate on another network, for instance, due to signal strength degradation. The “where” step decides which network to operate on. The “how” step defines the execution of the handover and how the device performs the transition, e.g. either doing a horizontal handover to the same network or a vertical handover to other networks. This paper addresses the “when and why” step. A companion work [4] addresses the “where” step, and a SIP-based solution for the “how” step is discussed in [3].

There are various efforts underway in standard organizations and industry forums to standardize vertical handovers, including IEEE 802.21[14] and 3GPP SA WG2[15]. For instance, the IEEE 802.21 Working Group is standardizing a set of media-independent handover (MIH) methods and procedures that facilitate mobility management. This allows the higher layer services or applications to treat all access

technologies in a generic manner. 802.21 specifies three media-independent services: event service, command service, and information service. Events may indicate changes in state and transmission behavior of the physical, data and logical link layers, or predict state changes on these layers. The event service maps to the “when and why” step mentioned above. Events defined in 802.21 include Link-Parameters-Change, Link-Up, Link-Down, Link-Going-Down, etc. Link layer triggers provide a mechanism for upper layer entities, such as Multi-radio Connection Manager, to monitor and detect changes in link quality and link parameters. For example, Link-Going-Down can be used by Connection Manager to search for a new point of attachment before the current point of attachment cease to carry frames, hence reduce handover delay in vertical handovers.

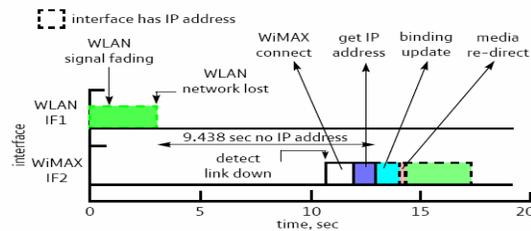


Fig. 1. Network switching without link triggers

Fig. 1 shows a baseline measurement on a generic operating system without any link layer triggers for a vertical handover, based on our prototype of a WiFi/WiMax multi-radio system [3]. It also shows the breakdown of the overall delay when a SIP audio session is transferred from a WiFi network to a WiMax network after WiFi signal degrades. (Note, here we do not consider authentication to the network). Transferring a SIP session across networks involves radio initialization, network address acquisition on the new interface, sending SIP de-registration and re-registration messages for old and new interfaces respectively (binding update), and media redirection [3]. Our measurements show that it took more than 9 seconds from the time WiFi signal was lost to the time the SIP client re-established the audio session on a new interface. This includes the time it took to detect loss of connectivity at the application level (since there is no link-layer triggers) and re-establish connection on a new interface, assuming that the SIP application does not include any intelligence to proactively detect loss of connectivity. These measurements demonstrate that an early indication of Link-Going-Down (LGD) will significantly help reduce the handover delay.

The standard organizations, however, do not define the algorithms for event services, for instance, the “intelligence” to decide when to generate a link layer trigger. In this paper, we address the problem of providing *accurate* and *predictive* link layer triggers to enable seamless handovers, horizontal or vertical, and minimize service interruption, and present our algorithms for trigger generation and trigger prediction. Our algorithms are of low complexity. Evaluation of the algorithms is based on real WiFi and WiMax signal traces. Results show that our trigger prediction algorithm is able to predict accurately and early in time. For instance, the WiFi results show that our algorithm accurately predicted 96% of LGD triggers and on average, the prediction was made 1.13 seconds before LGD is detected. Note that a LGD trigger would

be generated before loss of connection, hence would remove the gap between “network lost” and “detect link down” in Fig.1. The benefit of trigger prediction is to generate warnings of LGD even earlier in time. Accurate and early prediction would enable Connection Manager to take proactive actions and establish alternative connections before the quality of the current connection degrades to an intolerable level, thus minimize or eliminate service interruption.

Although there has been plenty of work in signal strength prediction [5, 7, 8, 9, 10], to the best of our knowledge, there are few works on link trigger predictions. In [6], an analytical model is proposed to set appropriate thresholds for generating link triggers, however, no evaluation on real signal strength traces were presented.

The remainder of the paper is structured as follows. In Section 2 we present the trigger generation process. In Section 3, we address trigger prediction and evaluate our trigger generation and prediction algorithms on WiFi and WiMax networks in Section 4. We conclude in Section 5.

2 Trigger generation

In this section, we discuss how to generate link triggers such as a LGD trigger, or in other words, when an event such as LGD is deemed to happen. In the next section, we discuss how to predict such an event. Typically, signal strength is used to reflect link quality [7]. For example, Received Signal Strength Indication (RSSI) is used in WiFi networks and Carrier to Interference-plus-Noise Ratio (CINR) in WiMax networks. In this paper, we focus our discussion on RSSI measurements. The same algorithm is applicable to other networks as well, as shown by results from WiMax networks in Section 4.

2.1 Trigger Generation Method

In WiFi networks, handovers are typically triggered when RSSI is lower than a pre-configured threshold. We follow this common practice to generate link layer triggers. Table 1 lists the thresholds for trigger generation used in this paper. (According to [1], a RSSI value of -80 corresponds to no effective communication. And -76 is a commonly used threshold to trigger handovers in WiFi networks.) They could be set to other values and the algorithms proposed in this paper are independent of the values.

Table 1. Thresholds for trigger generation

| | |
|------------------------------------|-----|
| Link Up threshold (LU_TH) | -60 |
| Link Coming Up threshold (LCU_TH) | -70 |
| Link Going Down threshold (LGD_TH) | -76 |
| Link Down threshold (LD_TH) | -80 |

Due to shadowing and fading effects, the radio environment is highly time varying and RSSI measurements (**raw RSSI**) fluctuate severely over time. Typically, a smoothing method is applied to smooth out raw RSSI. In addition, we apply a simple

optimization to reduce duplicated triggers, the *Link-status-update* method (shown in Table 2). The basic idea is to maintain the state of the link status in a conservative way, and only generate triggers when link status changes (e.g. when link status changes to LGD, a LGD trigger is generated). In Table 2, the top row specifies the value of the current smoothed RSSI, and the first column specifies the previous link status. The other items define the next link status given the previous status and the current smoothed RSSI value. For instance, if the link status is LGD, then unless the smoothed RSSI goes above LCU_TH, the link status will not change even if it goes above LGD_TH (as shown by the item on the 4th row and the 4th column). Thus, even though smoothed RSSI may go above and below LGD_TH multiple times, no additional triggers would be generated unless it goes up high enough.

Table 2. Link status transition in the Link-Status-Update method

| | $RSSI \geq LU_TH$ | $LU_TH > RSSI \geq LCU_TH$ | $LCU_TH > RSSI \geq LGD_TH$ | $LGD_TH > RSSI \geq LD_TH$ | $LD_TH > RSSI$ |
|------------------------------|--------------------|------------------------------|-------------------------------|------------------------------|-----------------|
| <i>LINK_UP (LU)</i> | LU | LU | LU | LGD | LD |
| <i>LINK_COMING_UP (LCU)</i> | LU | LCU | LCU | LGD | LD |
| <i>LINK_GOING_DOWN (LGD)</i> | LU | LCU | LGD | LGD | LD |
| <i>LINK_DOWN (LD)</i> | LU | LCU | LD | LD | LD |

2.2 Signal strength smoothing

As mentioned before, it is necessary to apply a smoothing method to smooth out raw RSSI. We next evaluate several smoothing methods to choose one that can both respond fast to short term changes and catch long term trend properly. The evaluation is based on the total number of triggers generated, unnecessary triggers, and triggering delays. Intuitively, unnecessary trigger indicates false alarms, and triggering delay indicates the level of responsiveness. The larger the triggering delay, the less responsive a method is to the changing trend of RSSI.

- A LGD/LD (or LCU/CU) trigger generated after obtaining the i th raw RSSI is a **unnecessary trigger** if 60% of next m raw RSSI values are no lower (or no higher) than LGD_TH/LD_TH (or LCU_TH/LU_TH).
- After obtaining the i th raw RSSI value, a LGD/LD (or LCU/CU) trigger is said to be delayed for one step, or **triggering delay** is increased by one step, if 60% of next m raw RSSI values are below (or above) LGD_TH/LD_TH (or LCU_TH/LU_TH). In this paper, we set $m=30$.

The smoothing methods we considered are listed below, where $x(i)$ is the i th smoothed RSSI, $r(i)$ is the i th raw RSSI measurement, and $N=50$. In practice, RSSI values are reported as integers. Hence, all smoothed RSSI values were also converted to integer values in our study.¹

- Average: average of the past N raw RSSI values.
- Exponential average: $x(i) = \alpha x(i-1) + (1-\alpha)r(i)$. We set α to 0.9 [1].

¹ For exponential average, we keep $x(i)$ in real number for calculation accuracy, and return the integer part as the value for the current exponential-average RSSI.

- Olympic average: Remove the highest n values and lowest n values of the past N values, and take the average of the remaining $N-2n$ values, $n=3$.
- Median: the median value of $r(i-N+1)$ to $r(i)$.
- Mode: the most frequent value of $r(i-N+1)$ to $r(i)$.²

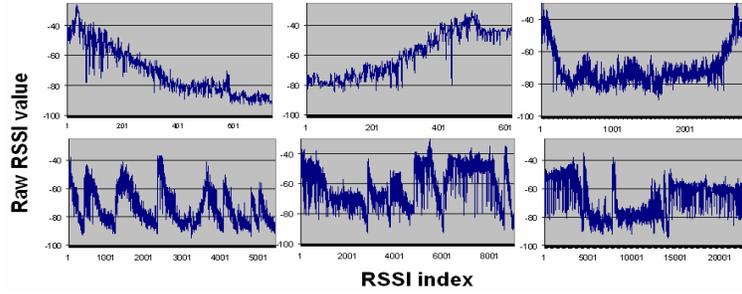


Fig. 2. Traces of raw RSSI, part I.

To evaluate the performance of these smoothing methods in trigger generation, we collected multiple RSSI traces by moving inside Intel’s office buildings. We used an IBM T60 laptop equipped with an Intel 3945 wireless card. Altogether, more than 42K raw RSSI values were collected. Fig.2 shows the traces, where x axis is the index of each RSSI measurement (100ms apart), and y axis is the value of each measurement. Each sharp increase of RSSI corresponds to switching to another AP. Roaming aggressiveness was set to the lowest, so that for each AP the laptop associated, we could collect a complete trace of RSSI from the start of the association to the time the connection is almost lost due to bad signal strength.

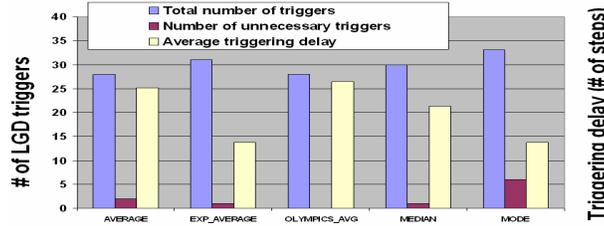


Fig. 3. Smoothing methods’ performance comparison, for LGD triggers

Using each smoothing method, triggers were generated for each trace in Fig.2 using the link-status-update method in Table 2. We then compare the total number of triggers, unnecessary triggers, and trigger delays for all smoothing methods. Fig.3 presents results for LGD triggers. (We see similar trends for other triggers.) Overall,

² Actually, to obtain the Mode, we divide the RSSI value space into buckets of 3, e.g., bucket[0]={-40, -41, -42}. Mode is defined to be the middle value of the bucket that contains the most raw RSSI values. The reason for not using the most frequent value directly is that the frequency is usually low and may not be the right representative of $r(i-N+1)$ to $r(i)$.

all methods generated similar number of triggers, Mode tends to generate more unnecessary triggers than others, and both exponential average and mode methods have much shorter triggering delay than other methods.³ We conclude that exponential average fits our needs the best.

In summary, for trigger generation, we use the Link-Status-Update method to avoid duplicated triggers, and use exponential average as the smoothing method to react faster to recent trend of signals strength and minimize service interruption.

3 Trigger Prediction

In this section, we discuss how to predict a link trigger, for instance, how to predict that LGD is going to happen. Such a prediction serves as a warning and Connection Manager could start preparation for handover. Once the predicted LGD trigger is really generated, the necessity for a network handover is confirmed and the on-going session (e.g. VoIP) could be transferred to the new connection. Note that although our discussion focus on prediction of LGD triggers, the same algorithm is applicable to other link layer triggers as well. For simplicity, in what follows, we use “RSSI” to refer to the exponential-average of RSSI, and explicitly use “raw RSSI” otherwise.

The objectives for trigger prediction are:

- A large “proactive window”, which is the period from the time a prediction is made to the time the predicted LGD trigger is generated.
- Accurate prediction. (1) No miss: when a LGD trigger is generated, a prediction has been made some time in advance. (2) No false-alarm: when a prediction is made, some time later a LGD is indeed generated; and in between the signal strength does not go high enough such that link quality is considered coming up instead of going down.

Note that if we miss predicting a trigger, we are doing no worse than trigger generation without prediction. However, with false alarms, we may trigger unnecessary preparation to network handovers or even unnecessary handovers.

As shown in Table 2, a LGD trigger is generated when the previous link status is Link-Up or Link-Coming-Up and the smoothed RSSI goes below LGD_TH. Hence, to predict the trigger, we first discuss how to predict future RSSI values, and then discuss how to predict triggers based on predicted RSSI values.

3.1 Signal strength prediction

The RSSI prediction problem is defined as below, where $x[i]$ denotes the exponential average of RSSI at step i , and $x'[i]$ the predicted value for $x[i]$.

- **RSSI prediction:** Given $x[i-N+1]$ to $x[i]$, predict $x[i+j]$, $j>0$.

We call $x[i-N+1]$ to $x[i]$ the data in the **history window**, and j the **prediction step**. Typically, RSSI measurements are collected every 100ms. Hence, predicting one step ahead corresponds to predicting 100ms into the future. Note that since smoothed RSSI values are reported as integers, predicted RSSI values are also returned as integers.

³ Though in this figure, the triggering delays of exponential average and mode are similar, mode has a much higher triggering delays in generating Link-Down triggers.

There has been a lot of work in predicting fading signals [5, 7, 8, 9, 10]. Most are based on complex computation, such as autoregressive methods, and are applied to predict raw RSSI. In contrast, for trigger prediction, we only need to predict smoothed RSSI. Moreover, we need a method that provides reasonable accuracy but does not require complex computation. The reason is the prediction algorithm would be executed each time a new RSSI measurement is obtained, and most likely it would be implemented in the wireless card driver, where simple methods are always preferred. As such, we investigated the following non-complex methods:

- **Straight-line:** $x'[i+j]=k*j+x[i]$, where $k=(x[i]-x[i-N+1])/N$ (the slope of the line formed by the oldest and newest RSSI values in the history window).
- **Step-by-step:** First predict $x[i+1]$ by applying the straight-line method to $x[i-N+1]$ to $x[i]$. Then slide the history window by 1 to include $x[i-N+2]$ to $x'[i+1]$ and predict $x[i+2]$, and so on.
- **Least square estimator (LSE)** [16].

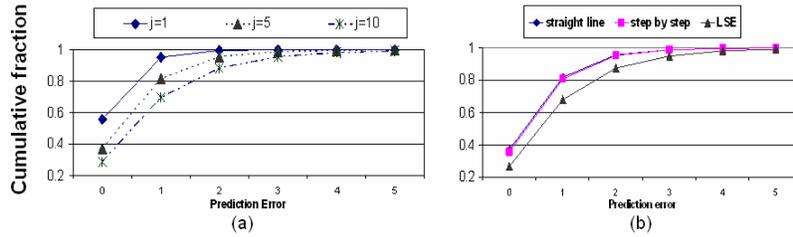


Fig. 4. RSSI prediction errors

To evaluate the prediction accuracy of the above three methods, we apply them to the traces in Fig. 2 and compare **prediction errors** (the difference between $x[i]$ and $x'[i]$.) Fig.4 presents some evaluation results, where the cumulative percentage is calculated based on the prediction errors from all traces. The closer a curve is to the top line (100%), the more accurate the prediction is. First, Fig.4(a) shows that the larger the prediction steps (larger j), the bigger the prediction errors. Apparently, there is a tradeoff between prediction accuracy and large prediction steps. In Section 4, we will discuss choosing an appropriate value for the prediction step in trigger prediction. Second, Fig.4(b) shows prediction errors for the three prediction methods over all traces for $j=5$. Interestingly, the straight-line and step-by-step methods (which have similar performance) perform better than LSE.⁴

In summary, due to its simplicity and higher accuracy, we choose the straight-line method for RSSI prediction.

3.2 Trigger prediction

Next, we discuss how to predict a LGD trigger by predicting RSSI. For simplicity, when at step i it is predicted that a LGD trigger will happen, we say that a **PreTrigger** is generated at step i , and we will use *trigger prediction* and *PreTrigger generation* interchangeably. A basic algorithm could be to apply the trigger generation algorithm

⁴ The reason could be that integer values (low precision) are used to represent RSSI values, and also these RSSI values are smoothed values and are not totally independent of each other.

to predicted RSSI values. However, since RSSI prediction has errors, and the further we predict, the larger the errors, this basic algorithm could result in many false alarms and missed predictions.

To improve the basic algorithm, we have a few observations. First, the algorithm should consider both the long term trend and short term changes, in order to avoid false alarms and respond to sharp changes in time. Second, in predicting RSSI values, the size of the history window (parameter N) is important. A large window is good for following the long term trend but less sensitive to sharp changes. On the other hand, a small window reacts faster to sharp changes but may result in more false alarms due to fluctuation of signal strength. We propose to leverage both window sizes in the prediction. To react fast to recent sharp decrease of RSSI values, we predict RSSI values by applying both large and small history windows and take the minimum of the two predicted values to be the predicted RSSI. On the other hand, to catch the long term trend to avoid false alarms, we apply a trend analysis algorithm to RSSI values in the large history window. Only when the predicted RSSI is below LGD_TH and the long term trend of RSSI is also going downward, will we make a prediction (i.e. generate a PreTrigger).

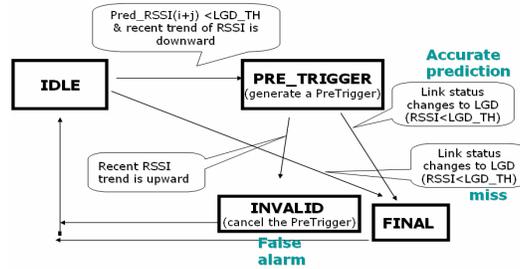


Fig. 5. State machine diagram of Trigger Prediction (PreTrigger) algorithm

The state machine diagram of PreTrigger generation is specified in Fig. 5, where $\text{Pred_RSSI}(i+j) = \text{Min}(x'[i+j]_{N_1}, x'[i+j]_{N_2})$ and $x'[i]_N$ denote the predicted value of exponential average RSSI at step i , by using window size N and the straight-line prediction method. We use N_1 to denote the large window size and N_2 the small window size, and set $N_1=50$ and $N_2=10$. Parameter j is tunable and we discuss how to choose an appropriate value for j in Section 4.

When the state transitions from IDLE to PRE_TRIGGER, a PreTrigger is generated. Later, if a LGD trigger is indeed generated, the state changes to FINAL (i.e. the prediction is accurate). The transition from PRE_TRIGGER to INVALID indicates that after the prediction is made, RSSI actually goes up and the prediction is considered a false alarm. The transition from IDLE to FINAL indicates that a LGD trigger is generated before a prediction could be made, hence a missed prediction.

To analyze the recent trend of signal strength, we apply the FFT method. (Refer to [2] for details of the FFT method.) Given a set of RSSI, $x(i-N_j+1)$ to $x(i)$, the trend analysis algorithm will return the trend as *downward*, *upward*, or *undefined* (meaning recent RSSIs do not show a strong trend of either going up or down).

Finally, Algorithm 2 presents the overall procedure of trigger generation and prediction. It is executed each time a new RSSI measurement is obtained.

| |
|--|
| Algorithm 1: Trend Analysis Algorithm |
|--|

| |
|--|
| <p><i>Step 1:</i> Apply the FFT method to get the trend of RSSI in the following history windows, which will return the trend as UP, DOWN, or UNDEFINED ($N_1=50$ and $N_2=10$):</p> |
|--|

- | |
|--|
| <ul style="list-style-type: none">• Long term trend: the trend of $[x(i-N_1+1), x(i)]$.• Half long term trend: the trend of $[x(i-N_1/2), x(i)]$.• Short term trend: the trend of $[x(i-N_2+1), x(i)]$. |
|--|

| |
|--|
| <p><i>Step 2:</i> The recent trend is downward if one of the following is true: long term trend is DOWN; long term trend is UNDEFINED and half long term trend is DOWN; or long term trend is UNDEFINED and short term trend is DOWN.</p> |
|--|

| |
|---|
| <p>Similarly, the recent trend of RSSI is upward if the following is true: long term trend is UP; long term trend is UNDEFINED and half long term trend is UP; or Long term trend is UNDEFINED and short term trend is UP.</p> |
|---|

| |
|--|
| <p>Otherwise, the trend is undefined.</p> |
|--|

| |
|---|
| Algorithm 2: Overall Algorithm for Trigger Generation and Prediction |
|---|

- | |
|---|
| <ol style="list-style-type: none">1. Initialize data structures when associated with AP and set $i=0$.2. When a new RSSI measurement is obtained, calculate exponential average, $x(i)$.3. If $i < N_2-1$, wait for more RSSI measurements. Otherwise, continue.4. Apply the trigger generation algorithm (Section 2) and generate triggers if link status changes.5. Apply the trigger prediction algorithm (Fig. 5), and generate PreTrigger when entering the PRE_TRIGGER state.6. $i=i+1$. Repeat steps 2 to 6. |
|---|

4 Evaluation of Trigger Prediction

To evaluate the performance of trigger prediction, we apply Algorithm 2 to real traces of signal strength measurements. For each trace, at each step a measured RSSI value (or CINR for WiMax) is read in from the trace and the algorithm is applied. For all traces, we count the number of accurate predictions, missed predictions, and false alarms. We also evaluate the length of proactive window of each accurate prediction.

4.1 Results from WiFi traces

We used eight WiFi traces for evaluation. Two traces were from those in Fig.2 (the first two in the bottom). Six more were collected and are shown in Fig.6. The top three were collected by walking from inside of an office building to the outside. The bottom three were collected by moving from inside to outside of a house, at different directions and with different speeds. Note that these traces include both sharp decay and slow degradation of signal strength.

Altogether, there are 25 LGD triggers generated in the eight traces. Table 3 presents the results for using different prediction steps, $j=5$ and $j=10$. With a larger prediction step, we could make earlier predictions (i.e. larger proactive window). As shown in Table 3, with $j=10$, for accurate predictions, on average our algorithm was able to predict LGD triggers 1.50 seconds ahead of time. On the other hand, the larger the prediction step, the more missed predictions and false alarms. In Table 3, with

$j=10$, there were 10 canceled triggers (false alarms) and 4 missed predictions, where both numbers are larger than those of $j=5$. Hence, we recommend setting $j=5$ to trade-off between prediction accuracy and large proactive window. On the other hand, Table 3 demonstrates that with $j=5$, our algorithm was able to successfully predict 24 out of the 25 triggers, and on average made predictions 1.13 seconds ahead of time (for reference, around half predictions were made more than 1 second ahead of time). Hence, by applying our algorithm, an warning that link would be going down could be sent out 1.13 seconds before LGD is detected, thus giving Connection Manager plenty of time to monitor alternative connections, select one with good quality, and initiate a handover.

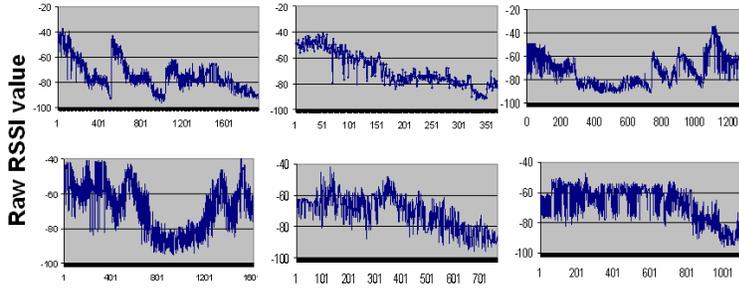


Fig. 6. Traces of raw RSSI, part II.

Table 3. Summary of trigger prediction results from WiFi traces

| | $j = 5$ | $j = 10$ |
|--|---------|----------|
| Total # of LGD triggers from all traces | 25 | 25 |
| # of PreTriggers generated by PreTrigger algorithm | 24 | 31 |
| # of Accurate PreTriggers | 24 | 21 |
| # of canceled PreTriggers | 0 | 10 |
| # of missed PreTriggers | 1 | 4 |
| Average length of proactive windows (seconds) | 1.13 | 1.5 |

4.2 Results from WiMax traces

We obtained a set of WiMax traces that were collected by driving around in the fields with real deployment of WiMax base stations. In each trace, measurements of downlink signal strength (CINR) were recorded every second.⁵ We extracted 5 traces that altogether span for 4 hours and 27 minutes and applied our algorithms. Figure 1 shows two of the traces. From private conversation, we learned that in WiMax networks, connection failures are likely to occur when CINR is below 0, and 9 was used to trigger handover in collecting these traces. Hence, for the WiMax traces, we set LGD_TH=9 and LD_TH=0.

⁵ Due to implementation limitations, measurements could not be taken at a higher frequency.

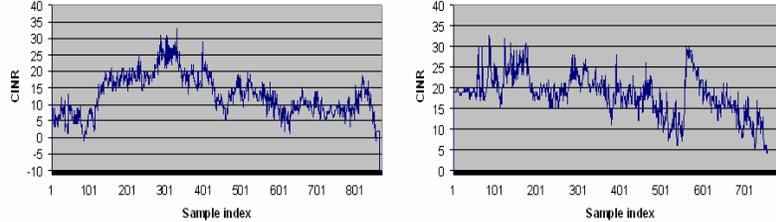


Figure 1 Sample WiMax traces

Results are presented in Table 4. Altogether there were 30 LGD triggers, and our algorithm successfully predicted 24 of them with an average proactive window of 8.7 steps.⁶ On the other hand, the results show one canceled PreTrigger and missed 6 predictions. We conjecture that the higher percentage of missed prediction in WiMax traces, compared to WiFi results, is due to various driving speeds in collecting WiMax traces. In the current design of our algorithms, we used fixed sizes of history windows. One enhancement is to apply adaptive window sizes to adapt to different moving speeds, which would be part of our future work.

Table 4 Summary of results of trigger prediction for WiMax traces

| | |
|---|-----|
| Total # of LGD triggers from all WiMax traces | 30 |
| # of Pre-Triggers generated by our algorithm | 25 |
| # of Accurate Pre-Triggers | 24 |
| # of canceled Pre-Triggers | 1 |
| # of missed Pre-Triggers | 6 |
| Avg. length of proactive window (# of steps) | 8.7 |

5 Conclusion

For multi-radio devices that expect ubiquitous connectivity across networks, vertical handovers could be lengthy (in terms of multiple seconds), since it typically involves a combination of link layer, IP layer, and application layer executions. In this paper, we proposed a novel approach to predict when a device needs to take proactive actions and prepare for handovers. With our link trigger predication algorithm, events such as Link-Going-Down could be predicted more than 1 second ahead of time. As indicated by the measurement in Fig.1, such predictions not only enables a Connection Manger to detect Link-Going-Down before connection is lost (instead of waiting for OS messages for confirmation), but also provides it extra time to turn on another radio, obtain IP address, and redirect a session. More specifically, with link triggers and trigger predictions, the service interruption, if any, is mainly due to media redi-

⁶ For WiMax results, we do not report the length of the proactive window as in seconds, because the CINR values were collected every 1 second, and this frequency may not reflect future implementations. With more frequent samples, we conjecture that the absolute time span of a proactive window may be smaller, but the its length in terms of average number of steps would remain similar.

rection and should be in milliseconds. We are now integrating our algorithms into our WiFi/WiMax multi-radio prototype system for further validation.

Another potential benefit of trigger prediction is that it could be applied to enable energy efficient network monitoring and selection for handovers. In general, to make appropriate handover decisions, it is desired to monitor the quality of other available networks (or available access points in other channels) [1][2]. However, leaving all the radios on all the time for monitoring network quality consumes a lot of power. By applying our work, one can turn on other radios only when a prediction of LGD is made. The early prediction enables a reasonably long time window for monitoring and comparing networks, and also enables radios not-in-use to be turned off to save power most of the time. We plan to further investigate in this direction.

Acknowledgments. The authors thank Brent Elliott, Intel Corporation, for acquiring and sharing the WiMax data.

Reference

- [1] V. Mhatre and K. Papagiannaki. "Using smart triggers for improved user performance in 802.11 wireless networks". In *Proceedings of the 4th international conference on Mobile systems, applications and services (MobiSys)*, 2006.
- [2] C. Guo, Z. Guo, Q. Zhang & W. Zhu, "A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks", *IEEE Journal on Selected Areas in Communications*, June 2004.
- [3] K N Choong, V S Kesavan, S L Ng, F de Carvalho, A L Y Low and C Maciocco, "SIP-based IEEE802.21 media independent handover — a BT Intel collaboration", *BT Technology Journal*, Vol 25 No 2, April 2007.
- [4] K. Mitsuya et al., "IEEE802.21 assisted Network Selection for Seamless Handover", in submission.
- [5] S. F. Chien, Huaiyu Liu, Andy L. Y. Low, Christian Maciocco, and Y.L. Ho, "Smart Predictive Trigger for Effective Handover in Wireless Networks", in submission.
- [6] S. Woon, N. Golmie, and Y. A. S,ekercioglu, "Effective Link trigger to Improve Handover Performance", In *Proceedings of The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2006.
- [7] Gregory P. Pollhi, "Trends in Handover Design". *IEEE Communications Magazine*, March 1996.
- [8] T. Eyceoz, S. Hu and A. Duel-Hallen, "Performance Analysis of Long range Prediction for Fast fading Channels,"*Proceedings of the 33rd CISS*. Baltimore, MD, March 1999.
- [9] L.Piroddi, W. Spinelli, , "Long-range Nonlinear Prediction: A Case Study," In *Proceedings of IEEE Decision and Control 42nd Conference*, June 2006.
- [10] S. Semmelrodt and R. Kattenbach, "Investigation of Different Fading Forecast Schemes for Flat Fading Radio Channels," In *Proceedings of IEEE VTC*, , Orlando, FL, Oct. 2003.
- [11] BT 21 Century Network, <http://www.btplc.com/>.
- [12] International Télécommunication Union, <http://www.itu.int/home/index.html>.
- [13] Telecoms and Internet converged Services and Protocols for Advanced Networks, <http://www.etsi.org/tispan/>.
- [14] IEEE P802.21/D01.00, March 2006
- [15] 3GPP Architecture Enhancements for non-3GPP accesses (3GPP TS 23.402), 2007.
- [16] NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/pmd/section4/pmd431.htm>.