

FIT: On-the-Fly, In-Situ Training with Sensor Data for SNR-Based Rate Selection

Hui Liu, Jialin He, Onur Altintas, Rama Vuyyuru, Joseph Camp and Dinesh Rajan

Abstract—Existing rate adaptation protocols have advocated training to establish the relationship between channel conditions and the optimal modulation and coding scheme. However, wireless devices for outdoor and vehicular communications frequently enter environments they have not yet encountered and therefore, have insufficient training for rate adaptation decisions. In addition, protocols are often optimally tuned for indoor environments but, when taken outdoors, perform poorly. In both cases, the decision structure formed offline lacks the ability to acclimate to a new situation *on the fly*. The diverse and ever-changing environments of increasingly mobile wireless devices call for a rate adaption scheme that can quickly adjust accordingly to form a unique environment set established by the user. In this paper, we propose an on-the-fly, in-situ training (FIT) mechanism which addresses the challenges of making rate decisions with unpredictable fluctuation and lack of repeatability of real wireless channels. We design and conduct extensive experiments on emulated and in-field wireless channels to evaluate the in-situ training process, showing that the rate decision structure can be updated as channel conditions change using existing traffic flows.

I. INTRODUCTION

Environmental variations make it challenging to achieve high throughput over wireless networks, especially with increasingly mobile and vehicular devices. To increase the spectral efficiency when the channel quality improves and maintain a robust connection when channel quality degrades, wireless devices must detect changing channel conditions and adapt their transmission rate accordingly.

One type of rate adaptation commonly used in practice is based on packet losses due to its ease of implementation. However, loss-based rate adaptation suffers from poor rate decisions due to a confusion between collision-based and channel-based losses [1]–[3]. A second type of rate adaptation uses channel quality directly (i.e., SNR-based protocols) and can distinguish between such losses. However, to perform optimally, SNR-based schemes require a particular type of wireless device to train in each environment in which it is used [4]. If the training is done offline, even excessive levels of training may not capture the relationship between SNR and the optimal transmission rate in different environments. Further, if a comprehensive decision structure were to be achieved for all contexts, the training still will likely be hardware dependent (i.e., the decision structure formed would be based on the

device type). The notion of channel quality differs across devices due to various factors from the RF front end, antennas, and gain control mechanisms to the type of channel quality metric and averaging thereof. All of these factors lead to a distortion of the theoretical channel quality to transmission rate relationship, upon which prior SNR-based protocols have relied [5], [6].

Nonetheless, since mobile and vehicular nodes are now equipped with a plethora of sensors, they can establish a clearer notion of the environmental context with collected sensor information [7], [8]. Context-awareness can leverage the collected context information to facilitate the convergence of optimal transmission parameters when the user is transitioning across different environments. Context-aware rate adaptation protocols are just beginning to form with the consideration of vehicular or pedestrian speed, the direction of motion, and/or channel type information [7], [9], [10].

However, most rate adaptation mechanisms are based on a set of predefined decision structures, which map wireless measurements, e.g., SNR, to the performance of transmission rates for a finite set of environment types. New environments would need a different SNR to rate mapping. These predefined decision structures are not adaptable to new wireless environments and not effective when new environments are encountered, resulting in poor rate decisions, low throughput, and lack of robustness. For example, despite an exhaustive level of training on an SNR-based protocol in one environment, the performance of the rate decisions and resulting throughput could still be very poor if the environment of training is different than the environment of operation. To collect enough training data in the new environment to cover all possible situations, there is a delay between training data collection and deployment of the new decision structure with a stable performance. In response to the aforementioned rate adaptation challenges, a key question that we seek to address in this work is the following: *Since users have been shown to have a unique set of environments [8], can the training for each user's rate decision structure be efficiently customized on the fly in the environment of operation?*

In particular, we investigate two challenges for in-situ training based rate adaptation: 1) How can training data be collected and leveraged across highly dynamic and diverse environments? 2) How can the rate adaptation decision structure be modified according to newly-collected training data? In an attempt to solve these challenges, we design, implement, and evaluate an on-the-fly, in-situ training (FIT) framework. The main contributions of this paper are summarized as follows:

- We propose FIT, an on-the-fly, in-situ training based rate adaptation framework where the decision structure is

Hui Liu, Jialin He, Joseph Camp and Dinesh Rajan, Department of Electrical Engineering, Southern Methodist University, Dallas, TX, {huil, jhe, camp, rajand}@smu.edu; Onur Altintas, Toyota InfoTechnology Center, Co. Ltd., Tokyo, Japan, onur@jp.toyota-itc.com; Rama Vuyyuru, Toyota InfoTechnology Center, U.S.A. Inc., Mountain View, CA, rama@us.toyota-itc.com.

The work was done when Rama Vuyyuru was a member of the staff at Toyota ITC.

adjusted according to the dynamic wireless conditions and environments when new context information and resulting transmission rate performance is observed.

- In contrast to offline training based adaptation methods, the proposed method requires no additional overhead. In this work, we directly use wireless performance and context information from existing data transmissions (i.e., there is no need to generate additional overhead traffic to train for rate decisions).
- We implement the proposed FIT framework on an off-the-shelf platform and test its performance in the field to demonstrate the significant impact of in-situ training updates in rate selection performance.

The remainder of the paper is organized as follows. We introduce our solutions to challenges for in-situ training and the proposed FIT framework in Section II. In Section III, we compare the performance of FIT with a purely SNR-based method, the offline training based algorithm, and Minstrel with emulated and in-field data and in-field experimental evaluation. Finally, we conclude the paper in Section IV.

II. ADAPTATION FRAMEWORK

We first introduce the notation used in this work. Let set $\mathcal{A} = \{A_1, A_2, \dots, A_L\}$ represent the various attributes or context information that are available in a particular system, where L is the number of available context attributes. Let $\mathcal{M} = \{m_1, m_2, \dots, m_M\}$ represent the set of transmission modes (i.e., modulation and coding schemes) available in a given system, where M is the number of modes.

In this paper, the optimization metric of interest is the measured throughput G . We use the notation $G(m_i)$ to denote the throughput of the i -th transmission mode. The throughput is calculated as:

$$G(m_i) = (1 - PER(m_i)) \times R(m_i) \times \frac{l_{payload}(m_i)}{l_{packet}(m_i)} \quad (1)$$

where $PER(m_i)$ is the measured packet error rate, $R(m_i)$ is the physical layer data rate, and $l_{payload}(m_i)$ and $l_{packet}(m_i)$ are the size of the payload and the packet, respectively.

We consider a wireless node that has knowledge of the following context information: *channel type* and vehicular velocity, in addition to the measured SNR. We choose the context information set \mathcal{A} as:

$$\mathcal{A} = \{A_1 = \text{channel type}, A_2 = \text{velocity}, A_3 = \text{SNR}\}. \quad (2)$$

Two different channel conditions are said to belong to the same *channel type* if the performance (as measured by the throughput) of the various modes exhibits similar behavior for all values of the various attributes (the recognition of channel types is discussed in [11]). The velocity refers to the relative velocity between the transmitter and receiver nodes. In this adaptation system, the transmitter needs to select the desired mode based on the available context information. The objective is to select a mode from \mathcal{M} to maximize the throughput given

the context information \mathcal{A} . Formally, the problem is defined as follows:

$$\begin{aligned} & \max_{m_i \in \mathcal{M}} G_{c_\ell, v_j, s_k}(m_i) \\ & \text{given } A_1 = c_\ell, A_2 = v_j, A_3 = s_k \end{aligned} \quad (3)$$

Let $\mathcal{C} = \{c_1, c_2, \dots, c_C\}$, $\mathcal{V} = \{v_1, v_2, \dots, v_V\}$, and $\mathcal{S} = \{s_1, s_2, \dots, s_S\}$ represent, respectively, the sets of channel types (A_1), velocity (A_2), and SNR (A_3) values in the training set. We use $G_{c_\ell, v_j, s_k}(m_i)$ to denote the throughput of the i -th mode m_i with the context tuple $\{c_\ell, v_j, s_k\}$.

We make the following remarks regarding the problem formulated in (3):

- Although we use throughput as the indicator of link performance, other metrics such as BER, PER, delay, or jitter could also be used.
- To stay in focus, we consider only SNR, velocity, and channel type as the context information in this paper. However, different context information from the transmitter side or the receiver side can also be used in the proposed framework depending on the users' need.
- The 802.11a/b/g and 802.11p systems use a single 20 MHz or 10 MHz channel. The 802.11n doubles the channel width from 20 MHz to 40 MHz and offers a wide range of rate options in addition to two MIMO modes, the diversity-oriented single-stream (SS) mode and the spatial multiplexing driven, double-stream (DS) mode. FIT can equally treat each pair of rate and MIMO mode combination as one rate adaptation option and be easily integrated into advanced protocols.

In the following subsections, we first briefly review the offline training based framework and then present our solutions to the aforementioned two challenges for in-situ training in rate adaptation. Based on the solutions, we propose an in-situ training based rate adaptation framework, FIT.

A. Prior work

In prior work, we developed an offline training based rate adaptation framework [10], [12]. In the offline training phase, we consider that the training set includes the knowledge of the optimal transmission mode to use for each combination of context attributes. We use offline training to generate a mapping from the set of context attributes to the optimal mode.

The established decision structure for rate adaptation can suggest the potential optimal settings for the transmitter/receiver pair according to the channel conditions. Among classical learning methods, the decision tree is widely used due to its low complexity and competitive performance on the accuracy of prediction [13], [14]. Such algorithms typically use training data to derive an empirical relationship between the inputs (attributes) and the desired output (class). An example of the resulting relationship is represented using a tree-like classifier as shown in Fig. 1. During the rate adaptation phase, this tree serves as a look-up table to decide the optimal transmission mode for each context tuple $\{c_\ell, v_j, s_k\}$. For instance, when the received SNR is in the range (15 dB, 32 dB] and the velocity is higher than 45 kmph, mode 3 is suggested as the optimal mode based on the decision structure in Fig. 1.

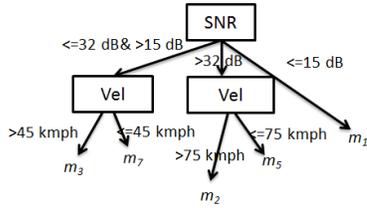


Fig. 1. An example decision tree for rate adaptation.

Once the tree-based classifier is generated based on classical tree building algorithms, e.g., C4.5, adapting its structure requires repeating the entire training process with a new or enlarged training set, which is time-consuming and difficult to implement in practice. Even with high levels of training (costly), new environments are still likely to be encountered, motivating us to investigate in-situ training based framework that can quickly find its context and adjust accordingly. In the following subsections, we first present our solutions to two main challenges for implementing in-situ training based rate adaptation, and then we propose the on-the-fly, in-situ training framework for rate adaptation (FIT).

B. Challenge 1: No Knowledge of Optimal Mode

The decision structure with the offline method models the relationship between the context information and the optimal mode. To derive the decision structure, an offline training based method requires the knowledge of the best mode m^* in the training data which can be obtained as:

$$m^* = \arg \max_{m_i} G_{c_\ell, v_j, s_k}(m_i) \quad \text{given } A_1 \equiv c_\ell, A_2 \equiv v_j, A_3 \equiv s_k. \quad (4)$$

The training data for the offline method is represented as

$$\{c_\ell, v_j, s_k, m^*\}. \quad (5)$$

Due to the fluctuating wireless channel, we only know the performance of a single mode or a random subset of modes for each combination of the attributes values. In other words, there is not enough time to sequentially try each rate for a given channel condition before that channel condition changes. Thus, a major challenge for in-situ training is that the optimal mode, m^* , is not available on the fly. Instead of using best mode in the offline training based framework, we have to alter the input and output of the decision structure for rate adaptation. In this section, we introduce our approach to prepare training data for in-situ training based rate adaptation, solving the first challenge.

To address this challenge, we directly learn the relationship between the context information and the throughput of each mode. The training data includes collected context information, the mode index, and the measured throughput which is represented as:

$$\{c_\ell, v_j, s_k, m_i, G_{c_\ell, v_j, s_k}(m_i)\}. \quad (6)$$

In the training phase, we measure the throughput $G_{c_\ell, v_j, s_k}(m_i)$ that the system can achieve using mode m_i in channel type c_ℓ with SNR s_k and velocity v_j . With a trained decision structure, the throughput for a given mode m_i with a certain context tuple $\{c_\ell, v_j, s_k\}$ can be predicted.

The decision tree can learn from continuous attributes; however, it does not accept continuous values (throughput) as the predicted labels, indicating we can not directly use $\{c_\ell, v_j, s_k, m_i, G_{c_\ell, v_j, s_k}(m_i)\}$ in the training process. To tackle this issue, we introduce a quantization technique, which is widely used in decision tree learning process when the attribute is continuous [15]. With this strategy, the training data will be,

$$\{c_\ell, v_j, s_k, m_i, Q[G_{c_\ell, v_j, s_k}(m_i)]\}. \quad (7)$$

Comparing (7) to (5), the new format of the training set does not need the best mode, making in-situ training possible.

C. Challenge 2: Classifier's Inability to Incrementally Retrain

The previously mentioned training methodology derives the decision structure as a look-up table based on the training data prepared offline and the decision structure will be embedded in the rate adaptation framework. However, the look-up table is static, and the entire training process must be restarted from the beginning for a newly encountered environment. In contrast, we develop an online version of the decision tree based learning algorithm to retrain the classifier incrementally according to the new set of encountered environment, thereby solving the second challenge. Our algorithm is based on the incremental learning algorithm in [16] and is able to efficiently update the existing tree with new training data.

The decision structure resulting from our algorithm is represented as a binary tree. The general procedure is based on Quinlan's information gain ratio [17]. If the classifier can predict the class label of new training data correctly, the data will be added to the set of retained data related to the leaf node. Further, the testing condition as well as the optimality of the sub-tree involved in the classification of the new data, will be modified based on the changed dataset using the attribute selection metric. The order for checking the optimality of a sub-tree is from the root node to its sub-trees. If the value for one of the attributes is missing, the data will be passed down the right branch that does not satisfy the testing condition.

The decision tree derived by the in-situ training process can be adapted to newly-collected data efficiently by updating a sub-tree. A sub-tree is usually much smaller than the whole decision structure, making it possible to implement incremental in-situ training for rate adaptation.

D. FIT Framework

Based on the solutions to the two challenges, we introduce our in-situ training framework for SNR-based rate selection. Instead of directly deriving the mapping function for rate adaptation by the offline training based algorithm as:

$$f_{\text{offline}} : \{c_\ell, v_j, s_k\} \rightarrow m^*, \quad (8)$$

first, we derive the function below using online learning with the proposed format of the training data in (7),

$$f_{\text{FIT}} : \{c_\ell, v_j, s_k, m_i\} \rightarrow Q[G_{c_\ell, v_j, s_k}(m_i)]. \quad (9)$$

Then, given the collected context information, (9) is able to predict the throughput for all the modes. Subsequently, the best

mode m^* is selected by comparing the predicted throughput values of all modes as:

$$m^* = \max_{m_i} f_{\text{FIT}}(c_\ell, v_j, s_k, m_i). \quad (10)$$

The framework of FIT is depicted in Fig. 2. With the collected context information $\{A_1, A_2, \dots, A_L\}$, FIT performs rate adaptation as follows:

- **Step 1:** Initialize the decision tree T ;
- **Step 2:** For new incoming data $\{A_1, A_2, \dots, A_L\}$ and the mode index i in \mathcal{M} , i.e., $m_i \in \{m_1, m_2, \dots, m_M\}$, the decision tree T predicts the throughput values of m_i as $\hat{G}(m_i)$;
- **Step 3:** Choose the mode m^* with the highest predicted throughput (10) to configure the transmitter;
- **Step 4:** Measure the throughput performance with mode m^* and prepare a new training data of $\{A_1, A_2, \dots, A_L, m^*, Q[G(m^*)]\}$;
- **Step 5:** Update the decision structure T with the new training data $\{A_1, A_2, \dots, A_L, m^*, Q[G(m^*)]\}$;
- **Step 6:** Go to **Step 2**.

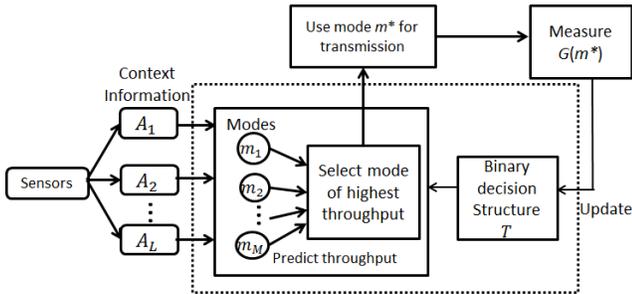


Fig. 2. FIT framework.

III. EXPERIMENTAL EVALUATION

In this section, after introducing the experimental setup and its implementation, we first investigate the effects of the proposed training data structure and quantization technique. We then evaluate the FIT framework with emulated data and test it with in-field mobility. We compare FIT with the static offline training based method, a purely SNR-based method and Minstrel.

A. Experimental Setup

To collect data for analyzing and comparing the performance of rate adaptation frameworks, we first use repeatable and controllable wireless channels, using the Azimuth ACE-MX channel emulator [18]. We compare the link adaptation frameworks in four standardized channel models specified by International Telecommunication Union (ITU). We use Channel 1, 2, 3 and 4 to represent the ITU Pedestrian A and B and ITU Vehicular A and B channel model, respectively [19]. The emulator allows us to test the performance of all available modes under each channel, SNR, and velocity combination to determine the actual best mode via exhaustive search. For an upper bound on rate adaptation performance, we use prediction accuracy, the throughput improvement over a rate adaptation method based only on SNR, and the throughput

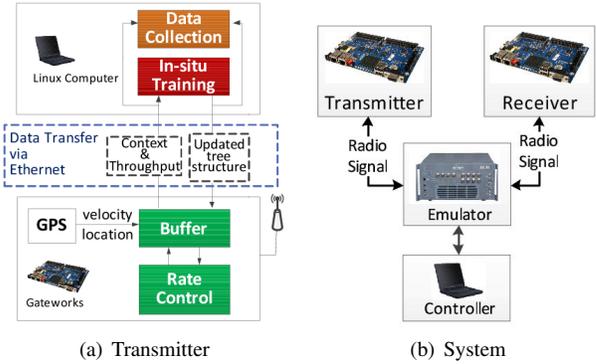


Fig. 3. Experiment Setup for the Hardware Implementation.

gap from maximum achievable found via exhaustive search as performance metrics.

All experiments are conducted by using an off-the-shelf, 802.11-based Gateworks 2358 platform with Ubiquiti XR-2 radios. Our hardware supports the following eight modes for link adaptation: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps. In each channel type, we collect the training data under 40 different scenarios, which are combinations of eight attenuation values: 0, 6, 12, 18, 24, 30, 36, 42 dB and five velocities: 0, 30, 60, 90, 120 kmph. The testing data (i.e., the data used to test the efficacy of the training) is collected with random velocities and attenuation values that are distinct from the training data.

To implement FIT in hardware, we use a Linux laptop, which runs the in-situ training based on the received training data. We build a special version of the Atheros Linux wireless driver - *ath5k*, in which the rate control module has been redesigned to act as an agent between the training laptop and the transmitting module. The driver is designed to pass the performance data to the laptop and to update the tree structure. We install the modified driver into the Gateworks 2358 board as the transmitter node, essentially changing the functionality of the kernel space. By comparison, the applications running in the user space are impervious to this change. Fig. 3(a) shows the above structure of the transmitter. Our system setup is shown in Fig. 3(b). As the velocity is not a common parameter for the wireless driver, we use code running in the user space to feed the velocity from the GPS module to the kernel space. This information is updated every second. Note that the computer in Fig. 3(b) only functions as a controller of the channel emulator to ensure the consistency of each experiment. It does not exchange any information with the wireless nodes.

B. Effects of Training Data Structure and Quantization

To solve challenges for FIT-based rate adaptation, we proposed a new training data structure and a quantization technique. In this subsection, we evaluate the effects of the new training data structure and the quantization technique.

In each context, we record the channel type, SNR, velocity, and throughput for all eight modes. To investigate the effect of the granularity of the quantization process on the performance of the proposed framework, we quantize the throughput value into N_Q discrete values and vary N_Q from 32 to 1000 with the following values 32, 63, 125, 250, 500, 1000. The performance of FIT with respect to N_Q is demonstrated in

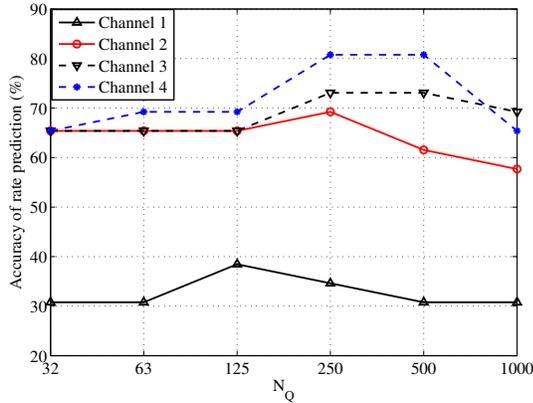


Fig. 4. Accuracy of rate prediction with various N_Q .

TABLE I
THROUGHPUT PERFORMANCE COMPARED TO SNR-ONLY METHOD AND THE MAXIMUM ACHIEVABLE THROUGHPUT WITH VARIOUS N_Q . “I” REPRESENTS IMPROVEMENT OVER SNR-ONLY METHOD, AND “G” DENOTES GAP FROM MAXIMUM ACHIEVABLE THROUGHPUT.

N_Q		32	63	125	250	500	1000
Channel 1	I (%)	-17	-18	-21	-14	-13	-17
	G (%)	26	27	30	23	22	26
Channel 2	I (%)	433	430	461	434	407	415
	G (%)	22	23	17	19	28	28
Channel 3	I (%)	172	140	168	161	157	165
	G (%)	22	28	24	26	30	23
Channel 4	I (%)	171	177	164	168	161	160
	G (%)	28	25	33	32	35	33

Fig. 4. As N_Q increases, the accuracy of rate prediction first increases and then decreases. Increasing the granularity of the quantization can help the decision tree distinguish between roughly equivalent throughput values, but too many class labels would result in a complex structure of the binary decision tree. To achieve the best performance, we should find a trade-off between the granularity of the quantization and the complexity of the decision tree. We also compare FIT with an SNR-only method, and the comparison with respect to N_Q are shown in Table I. We observe that the training based rate adaptation framework with the proposed training data structure consistently outperforms the SNR-only method with varying N_Q . On average, the training based framework with format 2 obtains 148% relative improvement in throughput. Since channel 1 is a slow-fading channel with a small delay spread, the SNR-only method performs effectively in this situation.

C. FIT Evaluation with Channel Emulator

In this subsection, we test the performance of FIT with emulated data with two kinds of trials: (1) training and testing in the same channel type, and (2) training and testing in different channel types. We collect training and testing sets in four different channel types.

The training data set contains 40 data points with velocities no higher than 60 kmph. The testing data set contains 26 data points with velocities higher than 60 kmph, new scenarios never encountered in the training set. The results are shown in Table II. With the offline-based method, the link adaptation framework is embedded in the classifier derived from the training set and tested with the testing set. In contrast, FIT retrains the link adaptation decision structure during the testing phase after each testing point collected in one context. Comparing the results between the offline-based method and FIT, we note that with the adaptive decision structure (FIT) improves the rate selection accuracy by 71% when training and testing in the channel 3 and by up to 333% when training in channel 4 and testing in channel 3. Similarly, FIT improves the throughput performance by 39% when training and testing in the channel 3 and by up to 345% when training in channel 1 and testing in channel 3. These results demonstrate the necessity and effectiveness of in-situ training, which is performed on the fly.

D. FIT Real-Time Performance Comparison

1) *Robustness to Channel Transitions*: The training sets for different channel types represent the various characteristics of the channel types. In other words, if the system is only trained with the training set from one channel type, without FIT training, the accuracy of the rate prediction will degrade when the system encounters a different channel type. In contrast, if FIT training is employed in such a case, the system will adapt to the new environment and achieve better performance.

To show the performance improvement of the FIT training over the offline training, we have created several scenarios on the channel emulator which involve channel type transitions. By using the channel emulator as the experimental setup shown in Fig 3(b), we expose both online training and offline training in the same channel-type transitions to show that the former has a higher adaptability to the environment variation than the latter. In this particular experiment, for each channel transition, the system is only trained with the training set of the first channel type so that it initially knows nothing about its performance in the second channel type.

TABLE II
PERFORMANCE OF FIT AND OFFLINE-BASED METHOD ON EMULATED DATA. “A” REPRESENTS THE ACCURACY OF RATE PREDICTION, “I” REPRESENTS IMPROVEMENT OVER THE SNR-ONLY METHOD, AND “G” DENOTES GAP FROM MAXIMUM ACHIEVABLE THROUGHPUT.

Channel		Training: 1			Training: 2			Training: 3			Training: 4		
		A(%)	I(%)	G(%)									
Testing: 1	Offline	0	-81	83	31	-46	53	12	-31	37	0	-55	58
	FIT	46	-24	29	62	-7	15	46	-28	33	65	-8	15
Testing: 2	Offline	8	149	67	73	549	15	19	313	46	50	524	19
	FIT	62	384	37	73	585	11	46	429	31	77	594	9
Testing: 3	Offline	0	171	74	27	648	29	38	593	34	15	721	22
	FIT	62	762	18	69	842	10	65	822	12	65	769	17
Testing: 4	Offline	0	210	68	58	681	19	16	538	34	58	692	18
	FIT	26	389	49	53	656	22	47	668	20	58	716	15

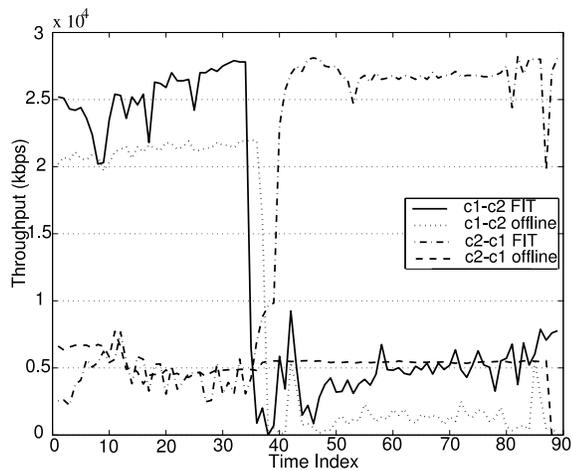


Fig. 5. Performance with channel transitions.

Fig. 5 plots the per-second throughput obtained in one transition using both training methods. The transition happens at approximately second 35. It can be seen that the offline training can (at most) achieve similar performance as the online training before the transition. However, its performance falls below that of the scheme with online training. Table III shows that FIT training tends to provide throughput gains over offline training when new environments are encountered.

TABLE III

THROUGHPUT COMPARISON FOR DIFFERENT CHANNEL TRANSITIONS

Transition	Offline	In-situ	Improvement
Channel 1 to Channel 2	8245	9452	15%
Channel 2 to Channel 1	8992	13664	52%
Channel 3 to Channel 4	7736	7220	-7%
Channel 4 to Channel 3	6274	7932	26%

2) *FIT Evaluation with In-Field Mobility*: This set of experiments evaluates the effectiveness of FIT by performing in-field trials. For mobility, the experimental setup includes two nodes located in different vehicles, which act as the transmitter and the receiver, respectively. The receiver's location is fixed, and the transmitter equipped with FIT is driving along the street. The antenna of each node is mounted on a car (2 meters in height). The built-in GPS module can inform the transmitter node of its velocity. As the receiver node is fixed, the velocity is the relative speed between the transmitter and the receiver.

TABLE IV

THROUGHPUT COMPARISON FOR IN-FIELD EXPERIMENTS (MBPS)

Location	Offline	Minstrel	FIT
1	0.72	2.09	2.71
2	1.01	2.22	3.36

Location 1 is along the main street of the SMU campus, and the street is surrounded by trees and buildings. *Location 2* is a park area. The park playground and trees can cause dramatic degradation of signal strength at certain locations. For the offline training based framework, we train four decision structures with training sets collected from the four broadly used channel types defined by the International Telecommunication Union (ITU). We test their performance and select the one with the highest measured throughput. Table IV shows the average throughput comparison for different algorithms. The channel types of the in-field propagation environments could

belong to another type that is beyond the four channel types defined by ITU. Therefore, the offline training based framework with predefined decision structures has worse throughput. FIT updates the decision structures with in-situ training on the fly to track the changing environment and always obtains the best performance. At *Location 1*, FIT obtains 30% and 276% relative throughput improvement over Minstrel and the offline training based framework, respectively. At *Location 2*, gains of 51% and 233% relative improvement are similarly achieved.

IV. CONCLUSION

In this paper, we discussed the two challenges posed when training rate adaptation on the fly in the field when encountering new environments. We presented our solutions to these challenges by introducing a new data structure for training and applying an online adaptive decision structure. In the future, we will explore the importance of other environment-related context information and how users can share FIT-based training to more effectively converge to optimality in unencountered scenarios.

ACKNOWLEDGEMENTS

The SMU authors were supported on this work by NSF grants CNS-0958436 and CNS-1150215 and by Toyota InfoTechnology Center, U.S.A. Inc.

REFERENCES

- [1] J. Choi et. al., "Adaptive optimization of rate adaptation algorithms in multi-rate wlangs," in *Proc. of ICNP*, 2007, pp. 144–153.
- [2] S. Wong et. al., "Robust rate adaptation for 802.11 wireless networks," in *Proc. of ACM MobiCom*, 2006.
- [3] J. C. Bicket et. al., "Bit-rate selection in wireless networks," M.S. Thesis, MIT, February 2005.
- [4] J. Camp et. al., "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1949–1962, Dec. 2010.
- [5] G. Holland et. al., "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. of ACM MobiCom*, Jul. 2001.
- [6] B. Sadeghi et. al., "Opportunistic media access for multirate ad hoc networks," in *Proc. of ACM MobiCom*, Sep. 2002.
- [7] L. Ravindranath et. al., "Improving wireless network performance using sensor hints," in *Proc. of USENIX NSDI*, San Jose, California, Apr. 2010.
- [8] R. Meikle et. al., "A global measurement study of context-based propagation and user mobility," in *Proceedings of the 4th ACM international workshop on Hot topics in planet-scale measurement*. ACM, 2012, pp. 21–26.
- [9] P. Shankar et. al., "CARS: Context-aware rate selection for vehicular networks," in *Proc. of IEEE ICNP*, Breckenridge, CO, Oct. 2008.
- [10] J. He et. al., "Design and experimentation of context-aware link-level adaptation," in *Proc. of IEEE INFOCOM*, 2012.
- [11] J. Camp et. al., "Context-aware analysis and adaptation," Patent Pending U.S. Patent Application 13/617, 367, Sep. 2012.
- [12] H. Liu et. al., "Astra: Application of sequential training to rate adaptation," in *Proc. of IEEE SECON*. IEEE, 2012, pp. 443–451.
- [13] S. R. Safavian et. al., "A survey of decision tree classifier methodology," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 21, pp. 660–674, 1991.
- [14] C. Yang et. al., "Application of decision tree technology for image classification using remote sensing data," *Agricultural Systems*, vol. 76, no. 3, pp. 1101–1117, 2003.
- [15] J. Quinlan, "Improved use of continuous attributes in c4. 5," *Artificial Intelligence Research*, 1996.
- [16] P. Utgoff, et. al., "Decision tree induction based on efficient tree restructuring," *Machine Learning*, vol. 29, no. 1, pp. 5–44, 1997.
- [17] J. R. Quinlan et. al., "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, March 1986.
- [18] "Aximuth ACE - MIMO channel emulator," <http://www.azimuthsystems.com>, Mar. 2011.
- [19] T. Sorensen et. al., "Extension of the ITU channel models for wideband (OFDM) systems," in *Vehicular Technology Conference*, 2005.