Protocol-Aided Channel Equalization in Wireless ATM

Jeffrey Q. Bao, *Student Member, IEEE* and Lang Tong, *Member, IEEE*

***Abstract*—In this paper, we study the equalization problem in time-division-multiple-access wireless asynchronous transfer mode (ATM) systems. Aiming at minimizing the overhead associated with equalization, we propose a protocol-aided channel equalization (PACE) approach for wireless ATM. Specifically, the medium access control (MAC) and data link control (DLC) protocols are exploited to provide known ATM cell headers to the receiver at the base station. A blind channel estimation-decision feedback equalizer (BCE-DFE) algorithm is developed for uplink data transmissions. There are two advantages of the BCE-DFE algorithm: the elimination of training symbols for uplink data bursts and the removal of channel estimation error propagation suffered by conventional block equalization schemes. Simulation results show BCE-DFE has robust performance for wireless ATM uplink data transmissions over fast time-varying channels.**

***Index Terms*—Adaptive equalizers, asynchronous transfer mode, blind equalizers, wireless ATM, wireless LAN.**

I. Introduction

WIRELESS asynchronous transfer mode (ATM) is a promising candidate [23] for broadband integrated services wireless networks. Major advantages of wireless ATM include 1) Quality of Services (QoS)-based multimedia services for mobile users and 2) seamless connectivity with wireline ATM networks. However, the design of wireless ATM networks faces many technical challenges. One important issue of the physical layer design in wireless ATM is how to overcome time-varying multipath impairments of radio channels. In this paper, we consider time-division-multiple-access (TDMA) systems with linear modulation techniques such as QAM and PSK, and address the equalization problem. Discussions about systems using spread spectrum or orthogonal frequency division multiplexing (OFDM) can be found in [4], [6], [23].

With possible maximum channel dispersion of 20 μs for outdoor environment, and with a high data rate (effective data rate of 25 Mb/s, gross data rate can be higher depending on physical overhead and coding), wireless ATM networks will have severe inter-symbol-interference (ISI) on radio links [10], [15], [25]. Therefore, equalization is necessary. In addition, wireless ATM is a multiuser packet system; different users send their data in short bursts over radio links. Thus, equalizers at the base stations will have to deal with short data bursts from different sources.

Referred to as training-based methods, conventional equalization techniques usually require training to learn the channel during which known symbols are transmitted. In wireless ATM, especially for the uplink, every burst goes through a different channel. Thus, if training-based equalization methods are used, every burst in principle has to carry training symbols. The number of training symbols required depends on the channel impulse response, the equalizer structure, and the equalization algorithm. For example, according to [6], if LMS algorithm is used in training-based methods, a one-half to five ATM cell long preamble needs to be transmitted for one training process. If we consider fast time-varying channels, more than one training may be necessary within a single data burst. Given a fixed system bandwidth, these training symbols greatly affect the bandwidth efficiency. The challenge for training-based equalizations in wireless ATM is to minimize the number of training symbols. There are ongoing research activities in this direction [5], [14]. An alternative to training-based methods is blind equalization which eliminates training. Studies of applying the Constant Modulus Algorithm (CMA) to short packet wireless systems have been reported in the literature [17].

Aiming at minimizing the overhead associated with equalization, we propose a protocol-aided channel equalization (PACE) approach for wireless ATM. Besides utilizing unknown data in blind channel estimation, we exploit wireless ATM protocol information to improve the equalization performance. Specifically, medium access control (MAC) and data link control (DLC) protocols are exploited to provide known ATM cell headers in uplink data bursts. A new algorithm based on blind channel estimation and the knowledge of ATM cell header is developed to provide robust performance for data bursts transmission over fast fading channels without training.

The paper is organized as follows: Section II presents the protocol-aided equalization approach; Section III develops a new algorithm based on the approach; Section IV gives simulation results and discussions.

Manuscript received August 13, 1998; August 16, 1999. This work was supported by the National Science Foundation under Contract CCR-9804019 and by the Office of Naval Research under Contract N00014-96-1-0895.

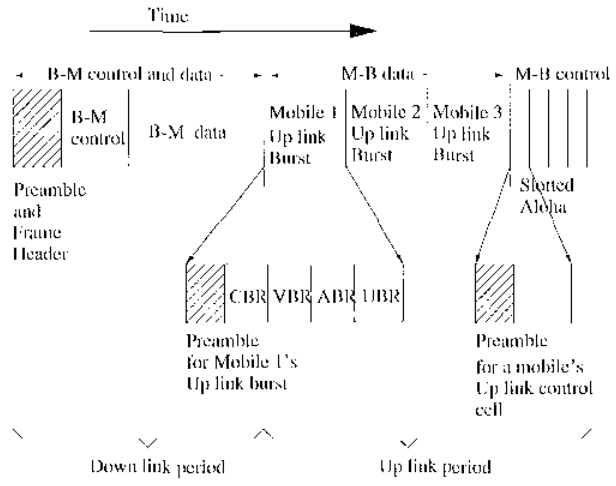
J. Q. Bao was with the Department of Electrical Engineering, University of Connecticut, Storrs, CT 06269 USA. He is now with Motorola Inc., Mansfield, MA 02048 USA.

L. Tong is with School of Electrical Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: ltong@ee.cornell.edu).

Publisher Item Identifier S 0733-8716(00)01289-0.

II. Protocol-Aided Channel Equalization (PACE) Approach

In this section, we present the PACE approach. First, we describe briefly the frame structure and the protocol which will be taken advantage of in the PACE approach. Then we explain how to exploit the protocol information for equalization.



B-M: Base station to Mobile
M-B: Mobile to base station

Fig. 1. WATMnet's MAC frame structure.

A. Frame Structure and MAC Protocol in Wireless ATM

Currently there is no standard for wireless ATM. We describe the frame structure and MAC protocol based on NEC's WATMnet prototype [22]. Other prototypes [8], [11], [20], [21], may have small variations.

As a TDMA/TDD (time division duplexing) system, WATMnet transmits ATM cells in time slots over radio links with each time slot carrying one ATM cell (ATM data and control cells have different size, thus time slots for data and control cells also have different size). Time slots are formed into frames according to the MAC protocol. Fig. 1 shows a MAC frame format of WATMnet. We can see each MAC frame has two periods: downlink and uplink. During the downlink period, control and data cells from the Base Station (BS) are broadcasted in a single burst. During the uplink period, different mobile users transmit their data and control bursts in the Mobile-to-Base station (M-B) data and control subframes respectively and there is guard time between different user' bursts.

The MAC protocol in wireless ATM is responsible for assigning time slots to different virtual connections (VC) so that their QoS requirements are satisfied and the bandwidth efficiency is maximized. Since ATM supports traffics with different characteristics, such as constant bit rate (CBR), variable bit rate (VBR), unspecified bit rate (UBR), and available bit rate (ABR), demand-assignment type of MAC's are more efficient. In fact most proposed MAC's for wireless ATM in the literature are demand-assignment based [8], [16], [18], [20], [22]. In such MAC's, the BS plays the role of assigning time slots. The most important criterion of assigning time slots is to satisfy every active VC's QoS requirements. Therefore, time slots are assigned per VC basis. Usually every active VC's QoS requirements have been obtained by the BS from the call setup process because ATM is a connection-oriented network. In WATMnet's example [18], for the downlink transmission, the BS can arrange time slots so that each downlink VC's QoS is met. For the uplink transmission, each mobile which wants to transmit cells for its uplink VC's has to reserve time slots by either sending a control cell to the BS during the current uplink M–B control subframe in a contention mode, or piggybacking the reservation in current uplink data cell transmissions. The BS will schedule the uplink time slot for all requesting mobiles, taking into account their VCs' QoS requirements and the available bandwidth. Then, during the downlink Base station-to-Mobile (B–M) control subframe of the next frame, the BS broadcasts time slot assignments for the uplink period of that frame. Mobiles will transmit data cells in designated uplink time slots. Time slots for the same mobile's VC's are often grouped into a burst as shown in Fig. 1. However, the uplink M–B control subframe slots are not controlled by the BS, they are randomly accessed by all mobiles in a contention mode (e.g., Slotted Aloha).

Note in Fig. 1, the shaded areas are preambles that can be used by equalization. In fact, it is suggested in [22] that a training-based decision feedback equalizer (DFE) was used with 18–27 bytes long preambles.

B. Exploitation of Wireless ATM Protocol Information for Equalization

Because the similarity between the downlink equalization problem and the traditional single user transmission system equalization problem, we shall not consider the downlink equalization in this paper. Again we use WATMnet prototype [22] as an example for the following discussion.

On the uplink, there are two kinds of bursts: control and data. One data burst can have varying size, from one ATM data cell (standard ATM data cell is 53 bytes long) to several data cells. Control bursts have the same size: one ATM control cell (in WATMnet, one control cell is 8 bytes long). Since control cells are short and they are critical to network operation, training-based equalization techniques are more appropriate. Next we will focus on equalization problems in the uplink.

Our objective is to achieve good equalization performance without transmitting training symbols for the uplink data bursts. For fast time-varying channels, one challenge for equalizer design is channel tracking. An effective way of data transmission is to use the so-called block structure in which unknown data and known symbol blocks are transmitted alternatively. Essentially, the unknown data blocks are designed small enough so that the channel can be assumed unchanged during one data block. The known symbol blocks inserted between unknown data blocks can be used as training symbols for channel tracking and unknown data block detection. For example, block transmission structure has been successfully applied in high frequency (HF) modem design [1]. But the block structure requires training symbols between adjacent data blocks. If we want to eliminate training symbols but keep the benefit of the block structure, we need to exploit other source of information so that the receiver can know some symbols in a data burst. Note that for cells from the same VC in ATM, the cell headers remain the same or change in a predetermined way. If the receiver can know cell headers of a data burst, the

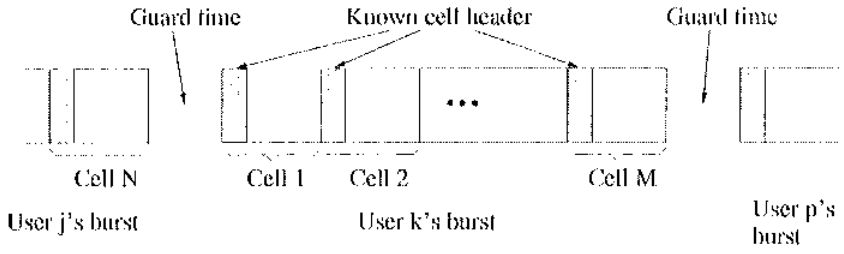


Fig. 2. Up link cell block structure.

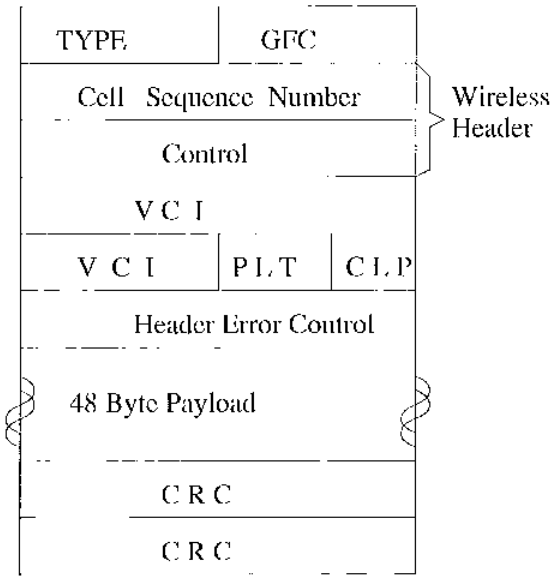


Fig. 3. WATMnet's data cell format.

data burst can be converted into a block structure with each cell's header as the known symbol block and the rest of the cell as the unknown data block. Fig. 2 illustrates such a converted block structure.

Can cell headers be known to BS's receiver so that they can be taken advantage of in equalization? To answer this question, we first show that it is necessary to include a properly designed header in a wireless ATM cell. Then we explain how to exploit the MAC and DLC protocols to make cell headers known to the BS's receiver.

A WATMnet's data cell format is shown in Fig. 3. It has two main components: header and payload. The payload part is the actual information the cell carries. The header part in wireless ATM can be broken into wireless header and original ATM cell header (possibly compressed). In WATMnet's example, the wireless header is two bytes long and contains a sequence number field and mobility field [23]. The original ATM cell header is compressed into a 12-bits-long virtual connection identifier (VCI) field and some control fields such as pay load type (PLT), cell loss priority (CLP), and general flow control (GFC). There is a header error control (HEC) field in the header which is a cyclic redundancy check (CRC) coding for the whole header.

In order to use the cell headers for equalization, they must be transmitted in data bursts. This is indeed the case in most wireless ATM proposals. Although transmitting cell headers can decrease bandwidth efficiency, there are several reasons that make their transmission necessary. First, wireless header part is necessary for network operation. The sequence number field is necessary for the data link control (DLC) to track the erroneous cells and perform selective automatic repeat request (ARQ) and the mobility fields are needed for network mobility management. Second, keeping the original ATM cell header can simplify the cell format conversion at both the BS and mobiles. Third, the VC identification field (e.g., VCI field in WATMnet's format) in a header can be used to identify a BS in mobile initiated handoffs in Virtual Connection Tree architecture [2]. Last, keeping the VC identification field in the header can improve network robustness against cell loss and cell misinsertion. For example, if the base station is expecting a cell of VC 5 from a mobile, but due to the control cell exchange error between the mobile and the base station, the mobile actually transmits a cell of VC 6. Without the VC identification field in the header, the base station will assume the cell is for VC 5. The result will be VC 6 loses one cell and VC 5 receives a misinserted cell. Therefore, it is necessary to transmit wireless header part and at least a VC identification field in wireless ATM.

Now we explain what part of cell headers can be known to the BS's receiver and why. Recall that for the MAC protocol described in Section II-A, the BS assigns time slots on the per VC basis for uplinks. In other words, the BS knows which uplink time slot carries which VC's cell, e.g., VCI fields of each uplink data cell is known to the BS. Once the VCI field of a cell is identified, the Type field can be known immediately because the MAC at the base station has the QoS information including Type of each VC. The PLT field is always the same (000) for data cells [27]. The GFL field has not been defined yet, we can assume it is known. The sequence number field can also be obtained by exploiting the DLC protocol. In wireless ATM, there is a DLC process running for each active VC to keep track of erroneous cells, the receiver at the base station can consult a particular VC's DLC process to obtain the current cell sequence number for that VC's cells. The HEC field is not known if the rest of the header is not completely known.

Therefore, if the BS's MAC and DLC can pass necessary information to the receiver, the header or at least major part of the header of uplink data cells can be known to the BS's receiver and the uplink data bursts can be converted into the block structure shown in Fig. 2.

III. Algorithm Development

In this section, we shall develop an algorithm based on the converted block structure of uplink data bursts with the payload part of a cell as the unknown data block and the known header part of a cell as the known symbol block. We consider time-varying channels that can change during a data burst and there is no training symbols for each data burst. We first examine an existing block channel equalization algorithm and point out problems if this algorithm is applied to wireless ATM. Then we develop a new algorithm by circumventing these problems.

A. Algorithm Schematic

1) A Conventional Block Equalization Algorithm: Consider applying a widely used block equalization algorithm [13] to wireless ATM uplink data bursts. The algorithm schematic is shown in the left side of Fig. 4. With an initial channel estimate $\tilde{h}$ which is obtained either from the previous cell or from training, the tentative detection $\tilde{s}$ can be obtained by an equalization scheme like decision feedback equalizer (DFE). The detected symbols $\hat{s}$ is then used to reestimate the channel to obtain the updated channel estimate $\hat{h}$, which is used to produce the final detection $\hat{s}$. At last, the channel is estimated again for the next cell by using the final decision $\hat{s}$. Known symbols are used in different detection schemes and the reestimation of the channel. A typical example of this algorithm schematic is nonlinear data-directed estimator (NDDE) [12] which is used in HF modems. Usually known symbol blocks are as long as the channel so that detection for different cells can be separated, especially if decision feed back is used. When known symbol blocks are shorter than the channel, they are enlarged to the length of the channel by including adjacent detected symbols from the previous unknown data block [9]. Since detected symbols are utilized in channel estimation, we refer this algorithm as decision directed channel estimation (DDCE)-DFE. Although DFE is shown in the flow diagram, block detection techniques like NDDE and those studied in [13] are also applicable.

There are two problems associated with DDCE-DFE. One problem is the channel estimation error propagation which is caused by the passing of channel estimation across cells. The reason is that previous cell's channel estimation may not be accurate due to either channel variation or detection error in the previous cell. An inaccurate channel estimate will lead to more detection error in the current cell, and a worse channel estimate for the next cell, thus even more detection errors for the next cell. This cycle of channel estimation error and symbol detection error can last a large number of cells. Another problem is how to provide the initial channel estimation for the first cell in a burst since there is no training symbols before each burst and it is not possible to pass channel estimation across bursts.

2) Blind Channel Estimation-DFE Algorithm: Motivated by DDCE-DFE algorithm, we propose the Blind Channel Estimation (BCE)-DFE algorithm for wireless ATM. There are several reasons BCE is necessary. Even if the BS's receiver can know cell headers, these known symbols alone are not enough as training. For example, suppose the whole header of 6 bytes is known, if a 16QAM modulation is used, the known header is only 12 symbols long. It is impossible to identify channels longer than 12 symbols. (Detailed results about channel identification conditions with known headers in wireless ATM can be found in [25].) Another reason is that by applying BCE to every cell in a burst, the initial channel estimation only relies on the current cell, thus the two problems associated with DDCE-DFE can be avoided.

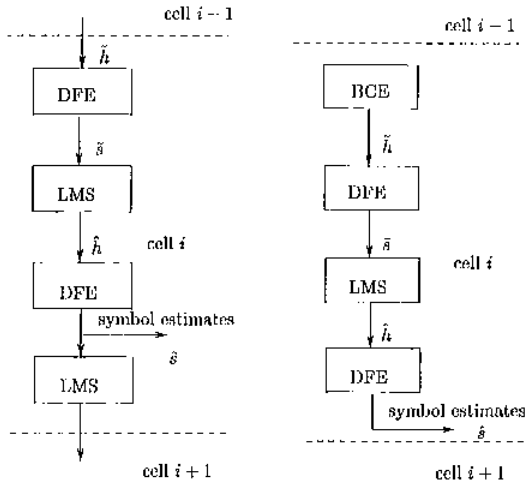


Fig. 4. Left: DDCE-DFE. Right: BCE-DFE.

The BCE-DFE algorithm flow diagram is shown in the right side of Fig. 4. The most important difference between DDCE-DFE and BCE-DFE is how the initial channel estimation is obtained. In DDCE-DFE, the initial channel estimate is obtained from previous cell's channel estimation or from the training symbols before the cell. In BCE-DFE, the initial channel estimate is obtained blindly from the current cell. With the blind channel estimate $\tilde{h}$, we can apply the same iterative process as in DDCE-DFE to get the final detection $\hat{s}$. However, because of the introduction of blind channel estimator, BCE-DFE will have higher computation cost than DDCE-DFE.

Next, we will explain the blind channel estimator used in BCE-DFE. Other techniques involved in the algorithm such as DFE are fairly standard and will not be discussed. Then we will discuss how header information is utilized in the BCE-DFE algorithm and some implementation issues in the context of wireless ATM.

B. Blind Channel Estimation

As discussed before, the smallest possible data burst is a single ATM cell, therefore, the blind channel estimator should have finite sample convergence property, i.e., the channel can be perfectly estimated from finite sample of received data without noise. A survey of blind channel estimation algorithms can be found in [26]. In this paper, a least square based blind channel estimator [29] is used in BCE-DFE.

1) The Model: Within the period of time of a cell, we assume the channel is time-invariant. The continuous time baseband model is given by

$$y_c(t) = \sum_k s_k h_c(t - kT) + n_c(t) \tag{1}$$

where

$y_c(t)$ received baseband signal;
s_k information symbol sequence;
T symbol period;
$h_c(t)$ (continuous-time) channel impulse response that includes the shaping filter at the transmitter, the impulse response of the propagation channel, and the receiver front-end filter;
$n_c(t)$ additive noise.

By sampling $y_c(t)$ at a rate of $2/T$, we obtain a discrete-time equivalent two-subchannel model shown in Fig. 5 where $h_k^{(1)}$ and $h_k^{(2)}$ are even and odd samples of $h_c(t)$, respectively [26]. Similarly, $y_k^{(1)}$ and $y_k^{(2)}$, n_k^1 and n_k^2 are even and odd samples of $y_c(t)$ and $n_c(t)$. The discrete time baseband model for a subchannel is given by

$$y_k^{(j)} = \sum_{i=0}^{L} h_i^{(j)} s_{k-i} + n_k^{(j)}, \quad j = 1, 2 \tag{2}$$

where $y_k^{(j)}$ and $n_k^{(j)}$ are the received (noisy) signal and noise of jth subchannel, and L is the channel order.

2) Least Square Blind Channel Estimator: The basic idea of lease square blind channel estimator is to exploit the multichannel structure. Ignore the noise first. The output of the two subchannels $x_k^{(1)}$ and $x_k^{(2)}$ are actually driven by the same input s_k, hence they are correlated. Considering the model shown in Fig. 6, following [29], we have

$$\begin{aligned} x_k^{(1)} &= s_k * h_k^{(1)} \\ x_k^{(2)} &= s_k * h_k^{(2)} \end{aligned} \tag{3}$$

where $*$ denotes convolution.

Then,

$$\begin{aligned} x_k^{(2)} * h_k^{(1)} &= (s_k * h_k^{(2)}) * h_k^{(1)} \\ &= (s_k * h_k^{(1)}) * h_k^{(2)} \\ &= x_k^{(1)} * h_k^{(2)}. \end{aligned} \tag{4}$$

Therefore, we have

$$x_k^{(2)} * h_k^{(1)} - x_k^{(1)} * h_k^{(2)} = 0. \tag{5}$$

After we collect K samples, we can write (5) into matrix form as

$$[\mathbf{X}_1 \quad -\mathbf{X}_2] \begin{bmatrix} \mathbf{h}^2 \\ \mathbf{h}^1 \end{bmatrix} = \phi(\mathbf{X})\mathbf{h} = 0 \tag{6}$$

where $\mathbf{X}_i$, $\mathbf{h}^i$ and $\phi(\mathbf{X})$ are defined as the following:

$$\mathbf{X}_i = \begin{pmatrix} x_L^{(i)} & x_{L-1}^{(i)} & \cdots & x_0^{(i)} \\ x_{L+1}^{(i)} & x_L^{(i)} & \cdots & x_1^{(i)} \\ \vdots & \vdots & \cdots & \vdots \\ x_{K-1}^{(i)} & x_{K-2}^{(i)} & \cdots & x_{K-L-1}^{(i)} \end{pmatrix}$$

$$\mathbf{h}^i = \begin{pmatrix} h_0^{(i)} \\ h_1^{(i)} \\ \vdots \\ h_L^{(i)} \end{pmatrix}, \quad i = 1, 2$$

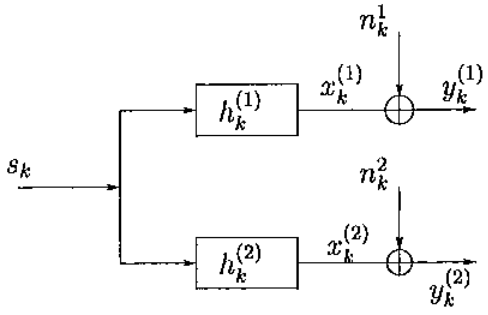


Fig. 5. Multichannel model.

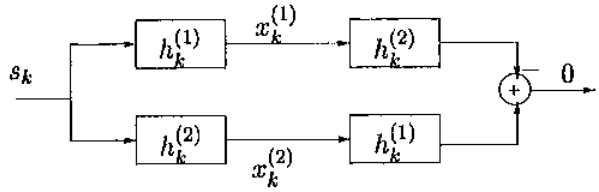


Fig. 6. LS blind channel estimation.

$$\phi(\mathbf{X}) = [\mathbf{X_1} \quad -\mathbf{X_2}], \quad \mathbf{h} = \begin{pmatrix} \mathbf{h}^2 \\ \mathbf{h}^1 \end{pmatrix}. \tag{7}$$

It has been shown in [29] that when two subchannels are coprime, i.e., $\mathbf{h}^1(z)$ and $\mathbf{h}^2(z)$ share no common roots, and the information source s_k satisfies some constraints on linear complexity, $\phi(\mathbf{X})$ has a one-dimention null space. Thus the channel $\mathbf{h}$ can be identified up to a scalar by finding a vector in the null space of $\phi(\mathbf{X})$.

When there is noise, the received data are $y_k^{(i)}$. Similarly, we can construct matrices $\mathbf{Y}_1$, $\mathbf{Y}_2$, and $\phi(\mathbf{Y})$ from $y_k^{(i)}$ as in the noiseless case. A simple way to estimate the channel is by minimizing the following least square cost

$$\hat{\mathbf{h}} = \arg \min_{\|\mathbf{h}\|=1} \|\phi(\mathbf{Y})\mathbf{h}\|^2 = \arg \min_{\|\mathbf{h}\|=1} \mathbf{h}^H \mathbf{Q} \mathbf{h} \tag{8}$$

where $\mathbf{Q} = \phi^H(\mathbf{Y})\phi(\mathbf{Y})$ and $\mathbf{h}$ is subject to the nontrivial constraint $\|\mathbf{h}\| = 1$. In fact, the solution of this minimization is just the eigenvector corresponding to the smallest eigenvalue of $\mathbf{Q}$.

C. Utilization of Cell Header Information in BCE-DFE

Known cell header information is utilized in detection and channel reestimation in BCE-DFE. Since the blind channel estimator does not require the knowledge of input signal, header information is not used. However, header information plays an important role of separating detection of a cell from adjacent cells. In other words, the (known) header blocks the detection error propagation across cells. Because DFE is used, the detection of the first symbol of the current cell involves the feedback of detected symbols before this symbol. With a known cell header before the first symbol in a cell, the header can be used as the feedback for the first symbol. There are two cases: 1) header length is equal to or longer than the feedback filter length and 2) header length is shorter than the feedback filter length. For the first case, since the header is known, the feedback for the first symbol is always correct, therefore detected symbols of the

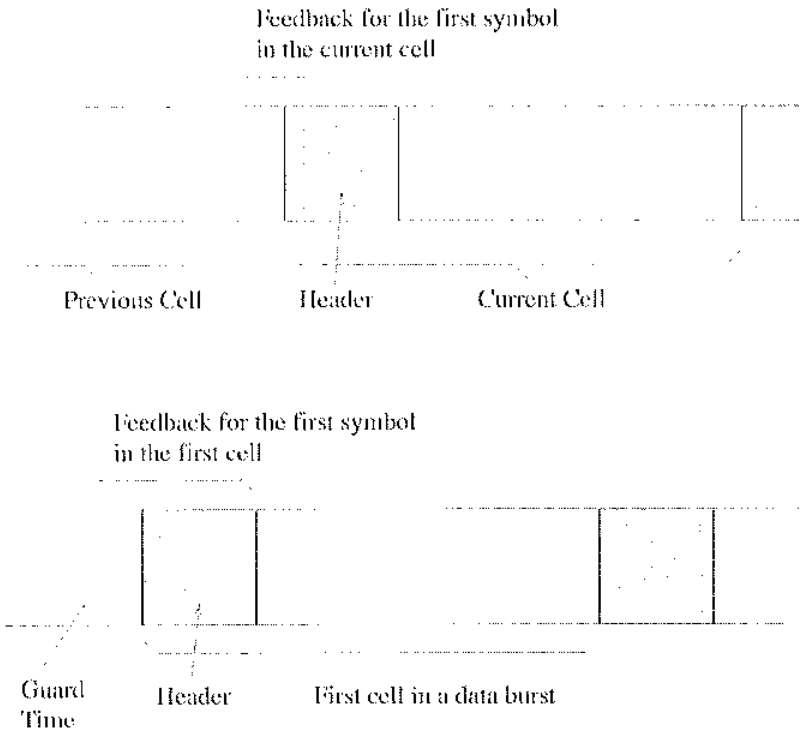


Fig. 7. Header utilization in detection.

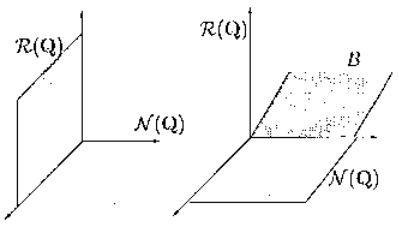


Fig. 8. Channel basis.

previous cell will not be included in the feedback for the detection of symbols in the current cell, detection errors are totally blocked across cells. For the second case, when the cell is not the first cell in a data burst, some detected symbols from the previous cell are used together with the header as the feedback for the first symbol in the current cell; when the cell is the first cell in a data burst, guard time (the period of time between two different mobile user's data bursts when no mobile transmits data) is used together with the header as the feedback (equivalent to padding zeros before the header). Therefore, detection errors are not totally blocked across cells. Fig. 7 illustrates the header utilization in detection. Of course, headers are used in the channel reestimation phase of the BCE-DFE algorithm as part of the whole cell.

D. Implementation Issues

Now we discuss some implementation issues of the BCE-DFE algorithm.

1) The Utilization of Channel Basis in BCE: The optimization in (8) requires the matrix $\mathbf{Q}$ have one-dimensional null space. However, it is often not the case in practical communication problems. In order to achieve channel identifiability, more information about the channel is needed. We apply the idea of "channel basis," i.e., we find a subspace in which the channel lies and identify the channel from the intersection between the channel basis subspace and the null space of $\mathbf{Q}$. Fig. 8 illustrates the geometric interpretation of this idea. $\mathcal{R}(\mathbf{Q})$ and $\mathcal{N}(\mathbf{Q})$ are the range space and null space of $\mathbf{Q}$ respectively, $\mathcal{B}$ is the channel basis subspace. The left side of Fig. 8 corresponds to the scenario that $\mathcal{N}(\mathbf{Q})$ is one dimension and the channel can be identified. On the right side is the scenario where $\mathcal{N}(\mathbf{Q})$ has two dimensions, with the help of basis subspace $\mathcal{B}$, the intersection of $\mathcal{N}(\mathbf{Q})$ and $\mathcal{B}$ can still identify the channel. Detail discussions about channel basis can be found in [19] and [30].. In Appendix A, we describe how to construct such a channel basis for multipath channels in wireless ATM networks.

Therefore, with a proper channel basis subspace $\mathcal{B}$, we can represent the channel as $\mathbf{h} = \mathcal{B}\mathbf{g}$ where $\mathbf{g}$ is the coefficient vector for $\mathbf{h}$ in the basis subspace $\mathcal{B}$. As a result, we can estimate the channel by estimating the coefficient vector $\mathbf{g}$

$$\hat{\mathbf{g}} = \arg \min_{\|\mathbf{g}\|=1} \mathbf{g}^H \mathcal{B}^H \mathbf{Q} \mathcal{B} \mathbf{g} \tag{9}$$

$$\hat{\mathbf{h}} = \mathcal{B}\hat{\mathbf{g}}. \tag{10}$$

Another advantage of doing this optimization is that we can reduce complexity since the dimension of $\mathbf{g}$ is usually smaller than the dimension of channel $\mathbf{h}$. For example, for two-ray channel

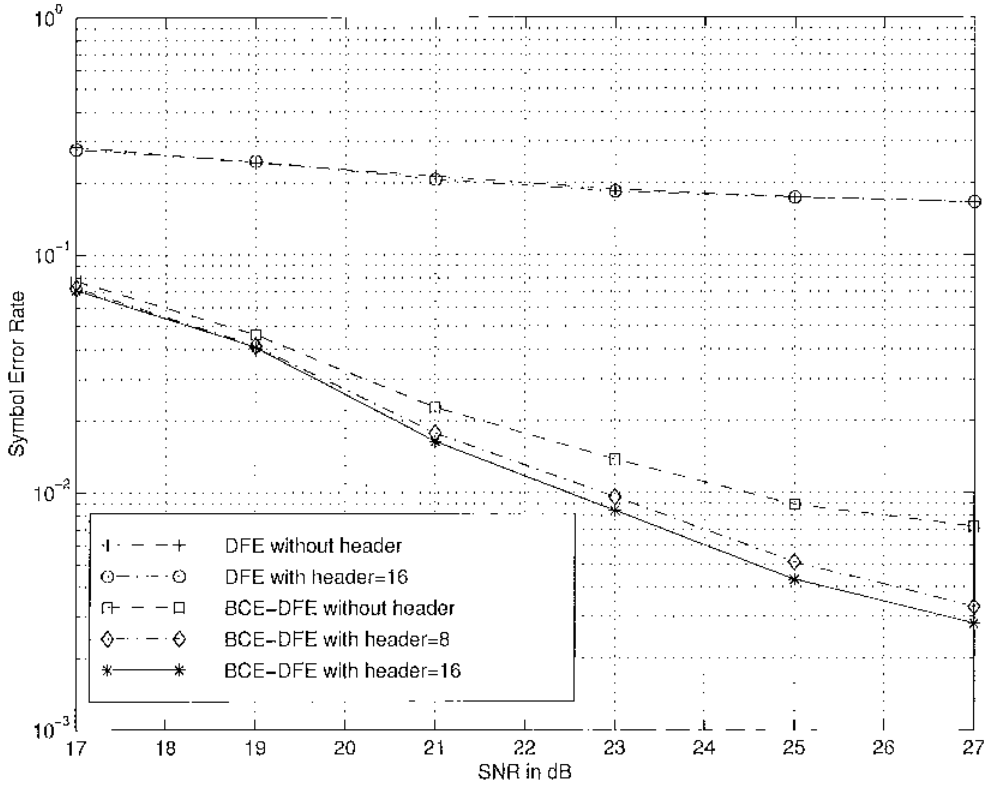


Fig. 9. SER versus SNR for BCE-DFE with different header length and DDCE-DFE.

models, the above optimization is reduced to solving the eigenvectors of a 2×2 matrix.

2) DFE Versus Block DFE and Linear Equalizers: In BCE-DFE, we chose a standard DFE as a detection scheme for a cell. Other block detection schemes like Block DFE [13] may have better performance than a standard DFE. But in wireless ATM, applying Block DFE to an ATM cell has high computation cost. The reason is that the size of an ATM cell is fairly large (in our simulation, assuming 8PSK, the size of a cell is 144 symbols), using Block DFE will involve solving the inversion of very large matrices which has high computation cost. For the fading channel we are considering, linear equalizers do not perform well. As a trade off between performance and complexity, DFE is more appropriate.

3) LMS Versus RLS: In BCE-DFE, when we have detected symbols of a cell, we use LMS to estimate the channel. In fact, both LMS and RLS are applicable in BCE-DFE. We choose to use LMS is because it is much simpler to implement than RLS, although LMS converges slower than RLS. We compensate this drawback of LMS by doing multiple runs of LMS for a cell to get a better convergence.

4) Utilization of Coding and Multiple Runs of Channel Estimation and Symbol Detection Cycle: Since there are usually coding for a whole cell, we can take advantage of this by doing coding check for the initial detection $\tilde{s}$. If it is correct, $\tilde{s}$ will be used as the final detection; if it is not, multiple runs of the symbol detection and channel reestimation cycle are performed. The rationale behind performing multiple such cycle is the following. For iterative algorithms like BCE-DFE, initial channel estimation (either obtained blindly or passed from the previous cell) usually contains some errors. The key for such iterative algorithms to work is that as long as the channel estimation error does not exceed some threshold, most of the detected symbols will be correct. For example, even if symbol detection error rate is 0.1 by using the initial channel estimate, 90% of the symbol detections are correct. Thus this detection can be used as training symbols to estimate channel again and produce a better channel estimate than the initial one. With an improved channel estimation, better detections can be produced. Therefore, this iterative process can improve both channel estimation and symbol detection. If the detection using initial channel estimate already contains no error, then there is no need to do this iterative process because no improvements can be made.

IV. Simulation Results

A. Simulation Setup

1) Channel Model: The channel model used in this paper is a two-ray multipath fading channel. Assuming the baseband input and output of the channel are $x_b(t)$ and $y_b(t)$, respectively, we have the following relation:

$$y_b(t) = \alpha_1(t)x_b(t-\tau_1) + \alpha_2(t)x_b(t-\tau_2) + n(t) \qquad (11)$$

where τ_1 and τ_2, $\alpha_1(t)$ and $\alpha_2(t)$ are delays and gains of each ray, $n(t)$ is the additive noise. We assume τ_1 and τ_2 do not

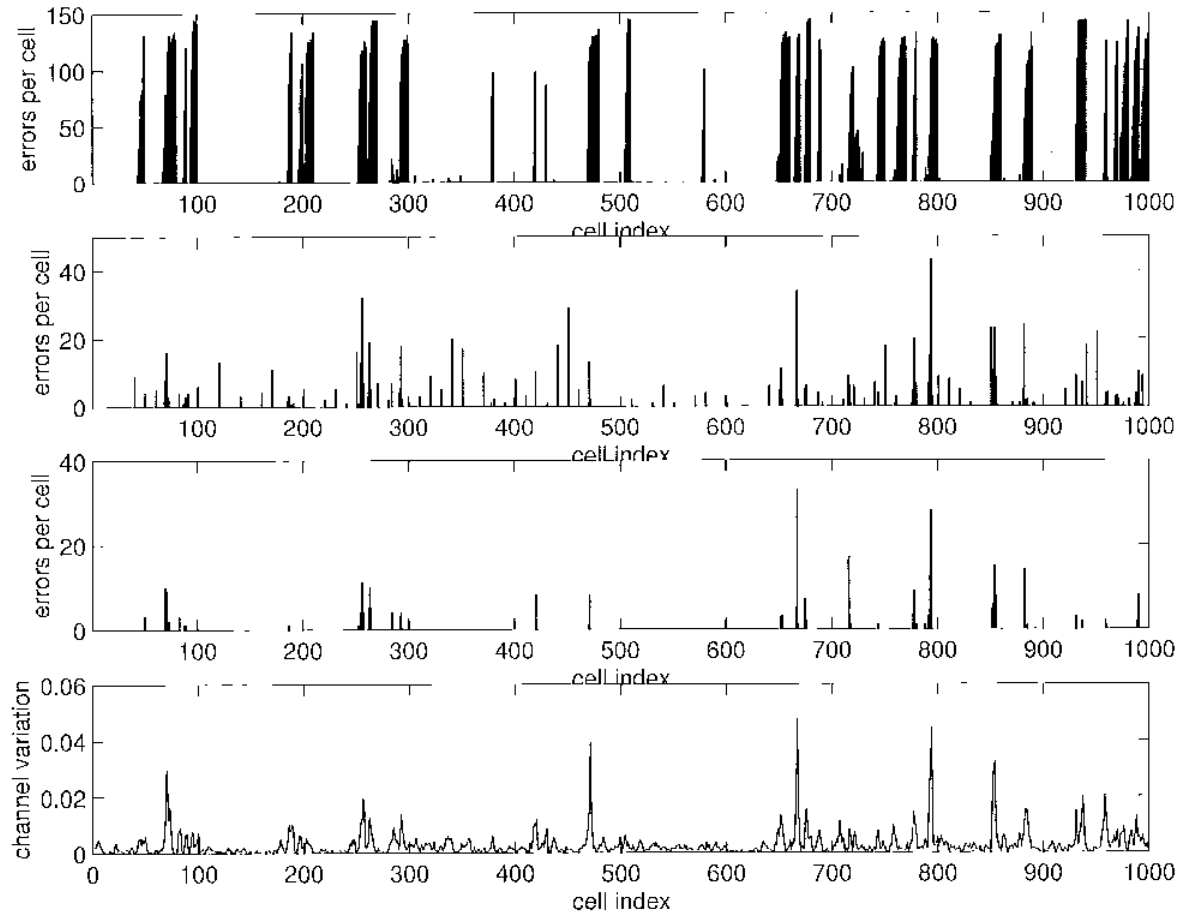


Fig. 10. Error pattern comparison, SNR = 27 dB.

change or change very slowly with respect to the data rate such that we can assume they remain the same for a data burst. However, $\alpha_1(t)$ and $\alpha_2(t)$ are two independent complex Gaussian random processes with a power spectrum specified by the Watterson Model [28] and they can change even within a cell's duration. The testing normalized maximum Doppler frequency is $f_m T = 2.08 \times 10^{-4}$ as suggested by MIL-STD-188 [1] for Watterson Model, T is the symbol period. We also assume a squared root raised cosine filter with roll-off factor 0.25 is used at both the transmitter and receiver. The modulation is 8PSK.

The Watterson Model is widely used in HF data communications. The main reason we use this model is to test the BCE-DFE algorithm in a severe ISI and fast time-varying environment.

2) Network Testing Parameters: For wireless ATM cells, we assume a cell is 56 bytes long with 48 bytes payload and 6 bytes header. The preamble associated with reservation and control messages has length of 192 symbols as specified in MIL-STD-188. We simulate uplink data transmission and there is no preamble for each data burst. The size of each burst is 10 cells. Cell header can be known, half known or totally unknown.

3) DFE Parameters: The DFE used in simulations is a minimum mean-squared error (mmse) finite impulse response (FIR) DFE. Denote the channel length as L symbols long, the forward filter length is $3L$. The feedback filter length is determined by the delay of the DFE (i.e., the symbol index difference between the output and the input of the DFE) and the forward filter length. Denote the delay as d, forward filter length as L_f, the feedback filter length L_b can be determined by

$$L_b = L_f + L - d - 2. \tag{12}$$

We can find the optimum delay by minimizing the mean-squared error (mse) of the output of the DFE as analyzed in [3].

B. Symbol Error Rate Performance Comparison and Header Length Effect

We compare the symbol error rate (SER) performance between BCE-DFE and DDCE-DFE. Taking into account possible unknown fields in the header, we include three cases for BCE-DFE: known header length = 0 symbols, corresponding to not knowing any header; known header length = 8 symbols, corresponding to only knowing half of the header; known header length = 16 symbols, corresponding to knowing the whole header. For DDCE-DFE, we consider header = 16 symbols and header = 0 symbols. The SER versus SNR performance is shown in Fig. 9. We can see clearly, no matter how

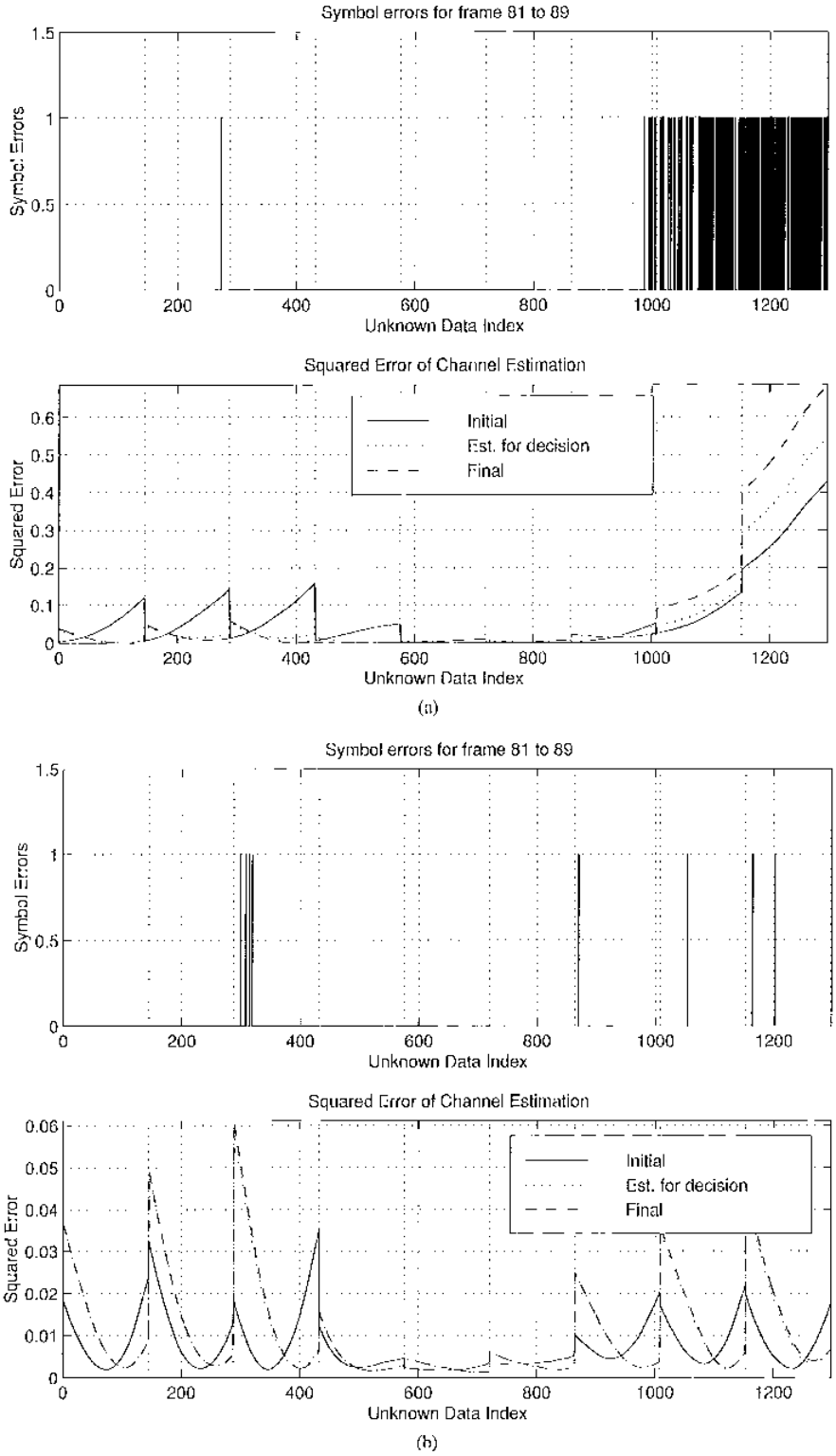


Fig. 11. Symbol error versus normalized channel estimation error for cell 81 89. (a) DDCE-DFE. (b) BCE-DFE.

long is the known header, the SER performance of BCE-DFE is better than DDCE-DFE.

For BCE-DFE, at lower SNR, the known header length does not affect the performance too much as can be observed from Fig. 9. However, as SNR goes higher, the header effect starts to show up in the performance. We can see that the longer the known header length, the better the SER performance. It is interesting to observe that

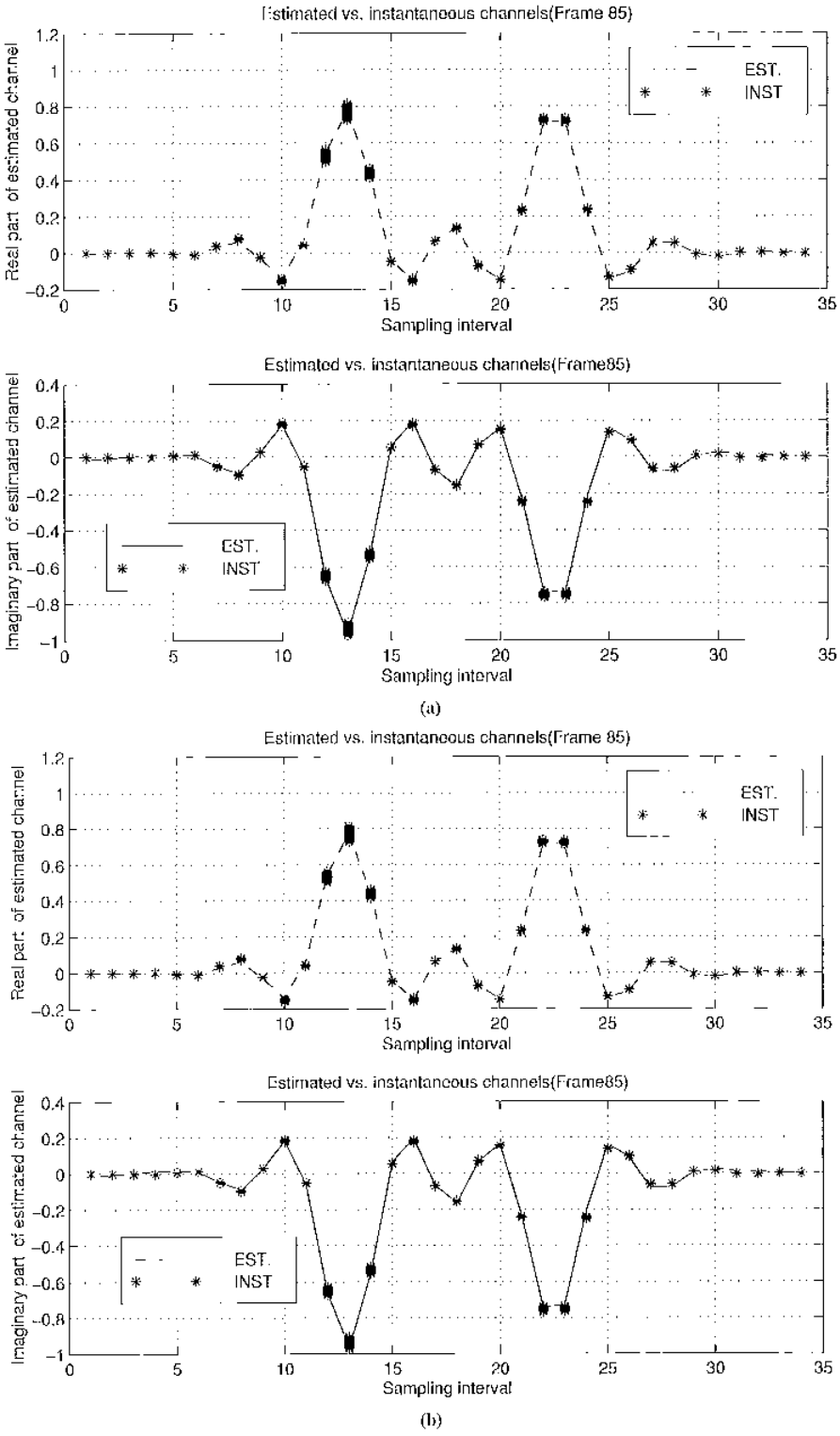


Fig. 12. Channel estimation of cell 85. (a) DDCE-DFE. (b) BCE-DFE.

BCE-DFE with half of known headers (length of 8) actually has just slightly worse performance than with whole known headers (length of 16). This suggests that BCE-DFE is not very sensitive to header length as long as known header is longer than certain threshold which remains to be investigated. We can also observe that at SNR = 27 dB, the SER of BCE-DFE is close to 10^{-3} when header length is 16 and 8. Since for 8PSK bit-error rate (BER) is

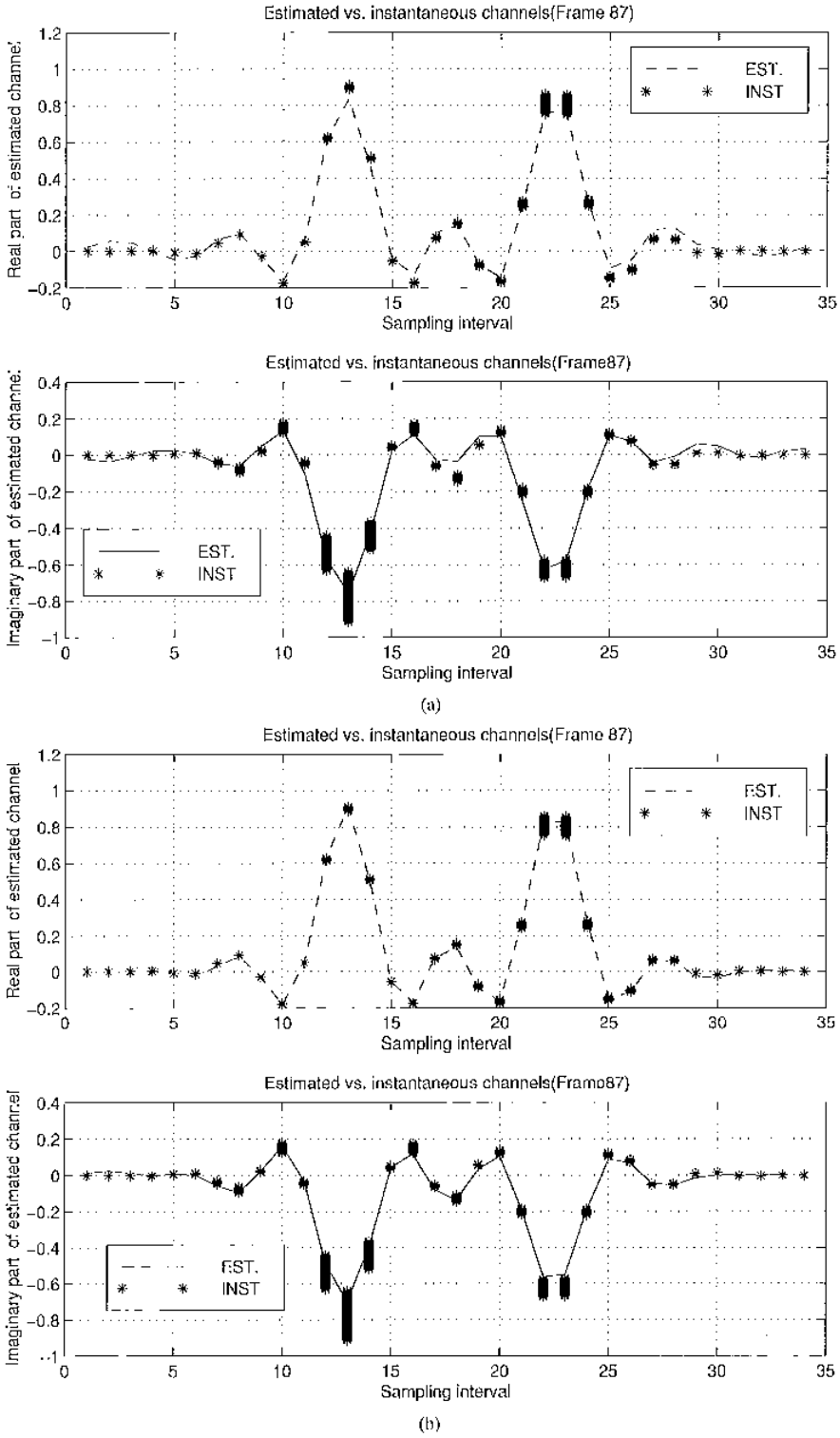


Fig. 13. Channel estimation of cell 87. (a) DDCE-DFE. (b) BCE-DFE.

roughly equal to 1/3 of the SER, the correspoding BER's are below 10^{-3} which satisfies MIL-STD-188 BER requirements for our testing channel. As suggested in [7], further BER improvements can be obtained by using Forward Error Control (FEC) coding.

As for the DDCE-DFE, knowing header does not improve the performance. The reason is that the channel estimation error propagation which will be shown soon is so severe, knowing some symbols can not correct the large amount of errors caused by the poor channel estimation.

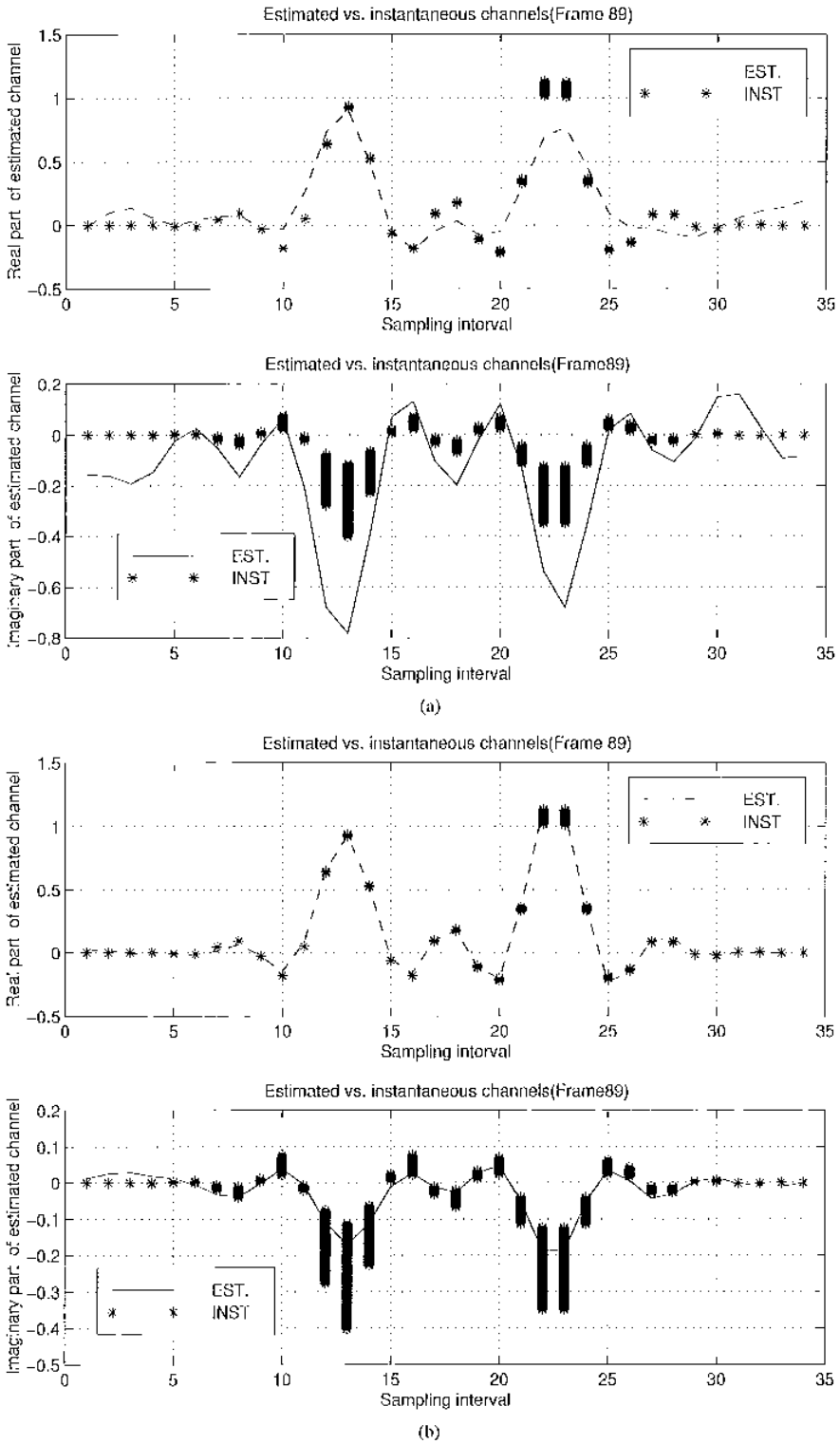
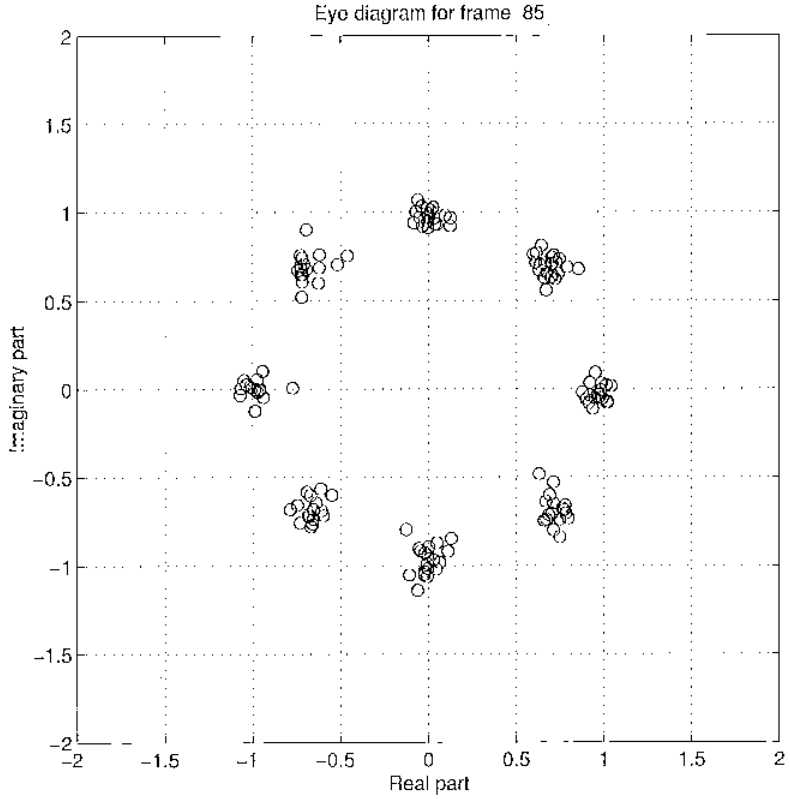


Fig. 14. Channel estimation of cell 89. (a) DDCE-DFE. (b) BCE-DFE.

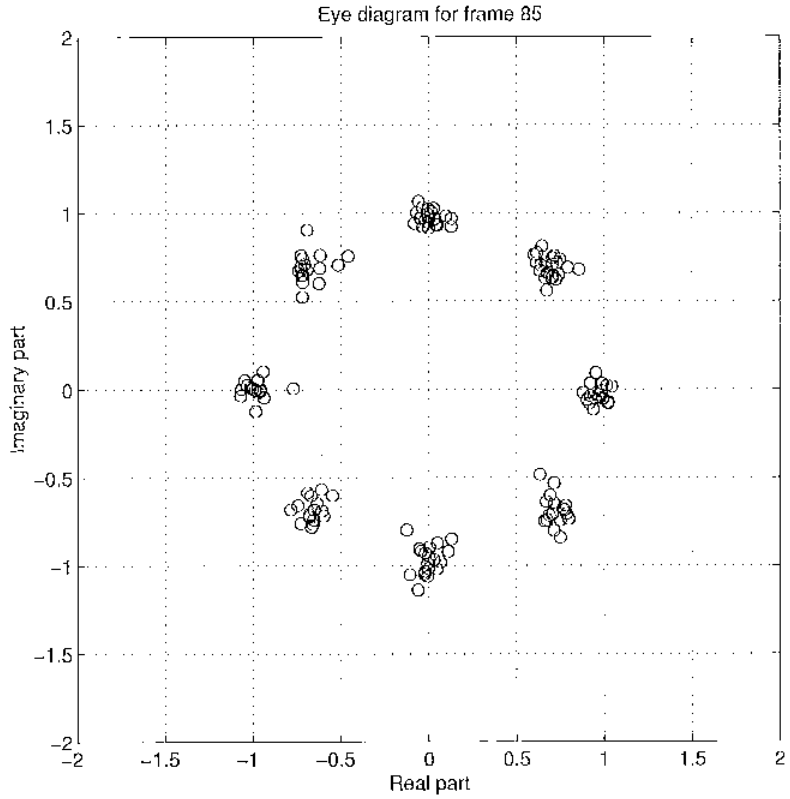
C. The Effect of Channel Estimation Error Propagation

To gain insight into the SER behavior of BCE-DFE and DDCE-DFE shown in Fig. 9, we now investigate the effect of channel estimation error propagation as discussed in Section III-A.

Fig. 10 shows symbol error patterns for 1000 cells. The top one is the pattern of DDCE-DFE, the second one is BCE-DFE



(a)



(b)

Fig. 15. Eye diagram of cell 85. (a) DDCE-DFE. (b) BCE-DFE.

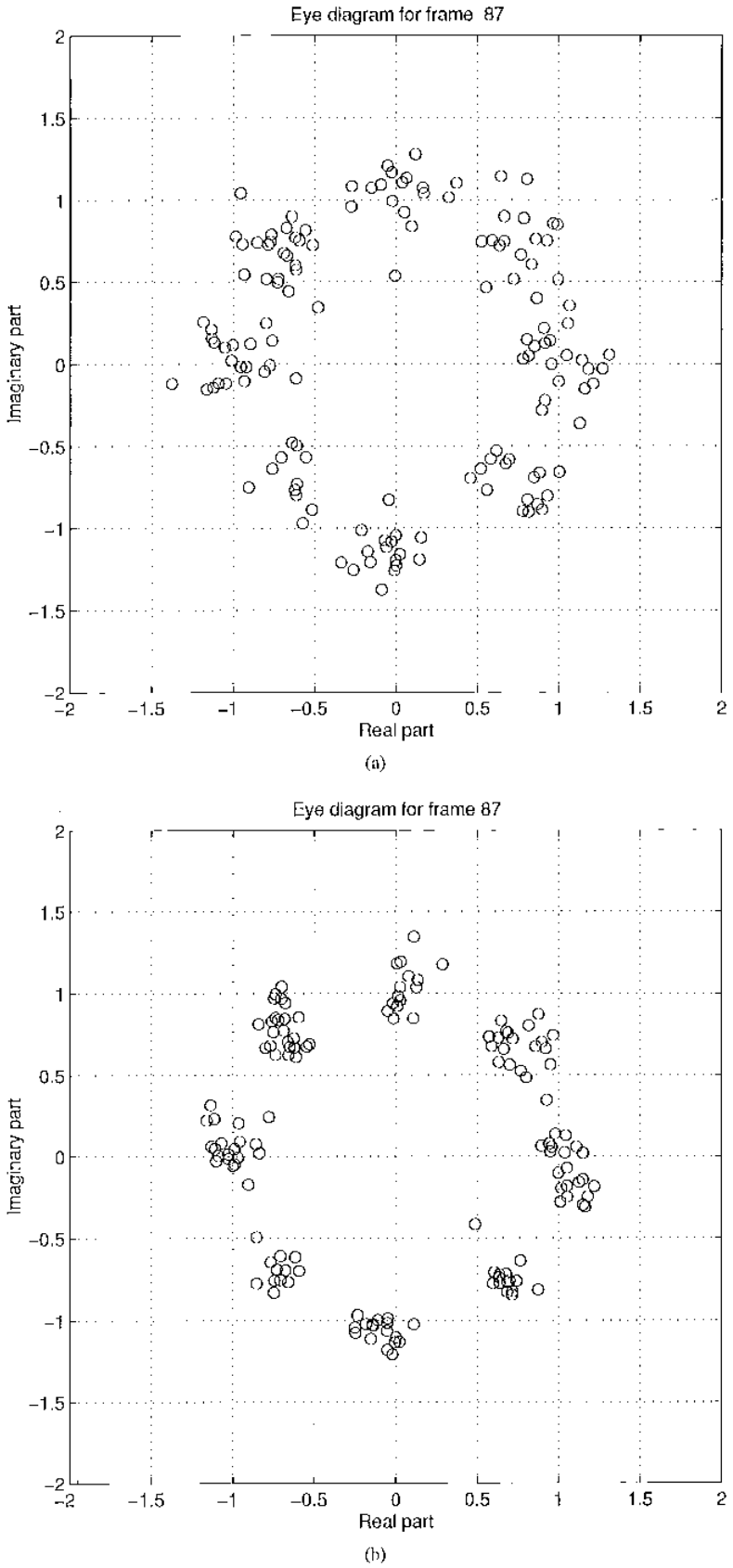
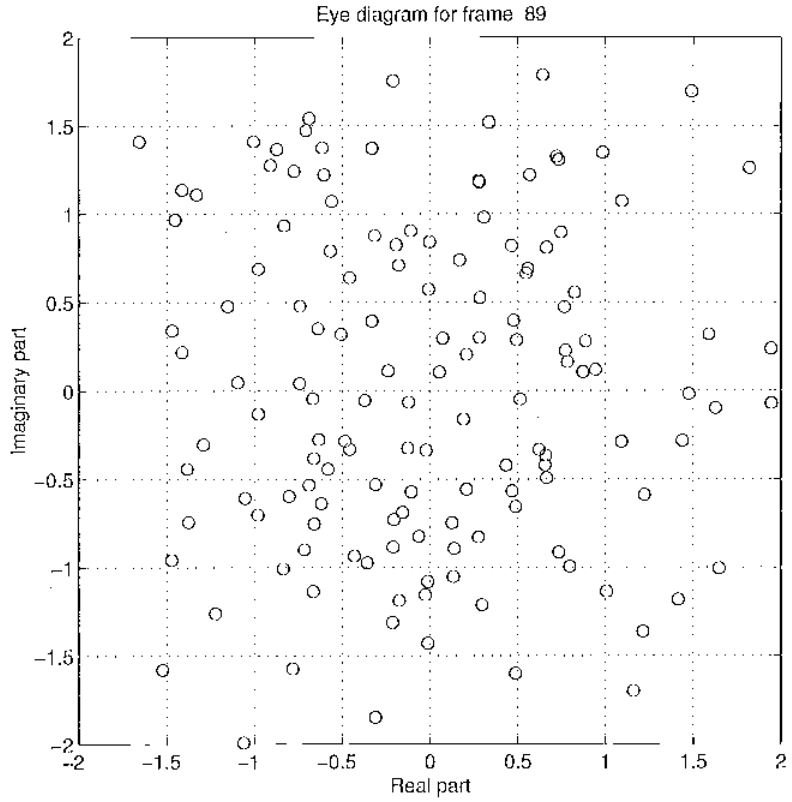
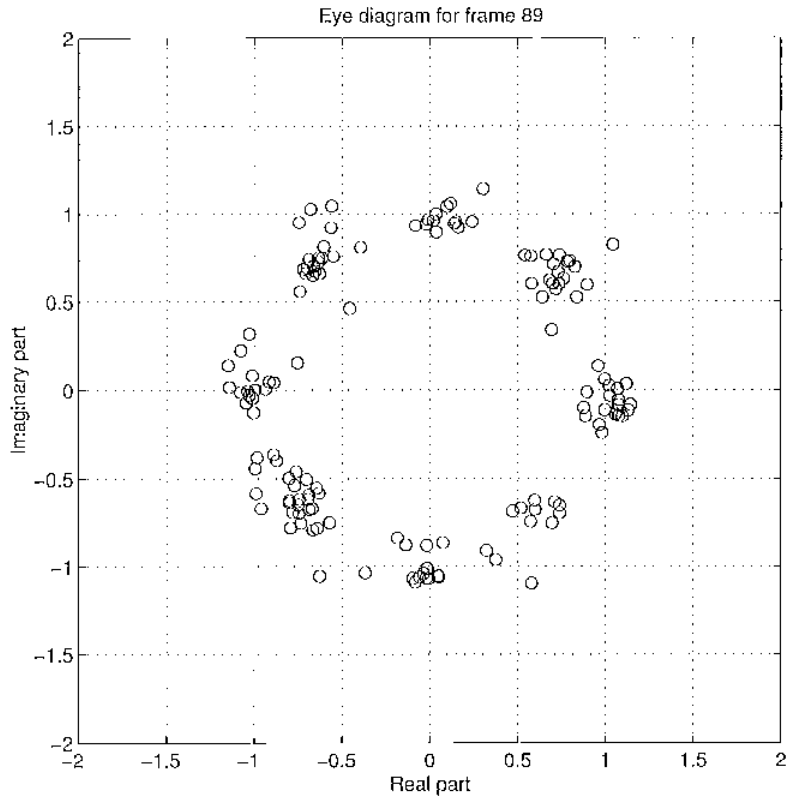


Fig. 16. Eye diagram of cell 87. (a) DDCE-DFE. (b) BCE-DFE.



(a)



(b)

Fig. 17. Eye diagram of cell 89. (a) DDCE-DFE. (b) BCE-DFE.

without known header, the third one is BCE-DFE with whole known header. The last one measures channel variation [1] during each of the 1000 cells. The index in Fig. 10 is in cells.

We can observe that in DDCE-DFE, errors are in blocks, especially when channel varies as shown in the channel variation plot at the bottom of Fig. 10. Keep in mind that in our simulation, one data bursts is 10 cells each, so the error block size is limited by the length of the actual data burst, for longer data burst, the error block could be longer. While in BCE-DFE, even without cell header knowledge, the errors are isolated. In fact, we can see that most of the detection errors happen when there are severe channel variations for BCE-DFE.

To demonstrate the effect of channel estimation error propagation, we now examine one data burst in detail. Fig. 11 shows the symbol error distribution in the burst of cell 81–89 and the normalized channel estimation error[2] for these nine cells. In Fig. 11(a) is DDCE-DFE, in Fig. 11(b) is BCE-DFE. Note the index in Fig. 11 is in symbols, one cell has 144 symbols. From cell 81 to 86, there is no big channel variation, thus for DDCE-DFE, there is only one symbol detection error shown in the upper symbol error distribution plot and the channel estimation error is small as shown in the lower plot (the solid line is the initial channel estimation error, the dashed one is the channel estimation error before final detection and the dotted line is the final channel estimation error). When the channel starts to vary from cell 87 (correspond to the channel variation peak before cell index 100 shown in the bottom plot in Fig. 10), the number of symbol detection errors in cell 87 starts to increase, thus in the channel estimation error plot, solid line, dashed line, and dotted line start to increase. As a result, cell 88 has a poor initial channel estimate, thus more symbol detection errors occur in cell 88 than in 87 and a worse initial channel estimate for cell 89. In cell 89, almost all symbols are wrong as shown in the upper plot, all three lines (solid, dashed, and dotted) in channel estimation error plot keep increasing. However, for BCE-DFE, the symbol detection errors remain isolated and the channel estimation errors are small even the channel varies after cell 87.

Figs. 12,13, and 14 show the instantaneous channel versus channel estimate before detection for cell 85, 87, and 89 for both DDCE-DFE and BCE-DFE. The upper two plots show real part of the channel, and the lower two plots show the imaginary part. For cell 85, the channel does not vary too much, for both algorithms the estimated channels are close to the instantaneous channel. From cell 87, the channel starts to vary, the channel estimate of DDCE-DFE for cell 87 deviates from the instantaneous channel at some taps, while the channel estimate of BCE-DFE is close to the instantaneous channel. The channel estimation error of DDCE-DFE becomes significant for cell 89

[1]Assume there are M samples during a cell at the receiver. For every sample, there is a channel vector $\mathbf{h}(k) = [h_0(k)\ h_1(k)\ \cdots\ h_L(k)]^t$, $k = 1,\ldots,M$, L is the channel order and t denotes transposition. We can define a mean channel vector $\bar{\mathbf{h}} = [\bar{h}_0\ \bar{h}_1\ \cdots\ \bar{h}_L]^t$ where $\bar{h}_j = 1/M \sum_{k=1}^{M} h_j(k)$. The channel variation for a cell's duration is defined as $\sum_{i=1}^{M} \sum_{j=0}^{L} |h_j(i) - \bar{h}_j|^2 / M \sum_{j=0}^{L} |\bar{h}_j|^2$.

[2]For a cell, there is a channel estimate $\hat{\mathbf{h}} = [\hat{h}_0\ \hat{h}_1\ \cdots\ \hat{h}_L]^t$. For every sample time during the cell period, there is an instantaneous channel $\mathbf{h}(k) = [h_0(k)\ h_1(k)\ \cdots\ h_L(k)]^t$, $k = 1,\ldots,M$, assuming total M samples per cell. The normalized channel estimation error for a sample time k is defined as $(\sum_{i=0}^{L} |h_i(k) - \hat{h}_i|^2)/(\sum_{i=0}^{L} |h_i(k)|^2)$.

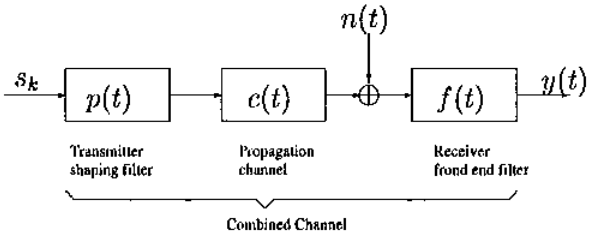


Fig. 18. Combined channel model.

as shown in Fig. 14, but the channel estimate of BCE-DFE only has errors on some small taps in the imaginary part.

As a result of the different channel estimation performance, Figs. 15, 16, and 17 show corresponding eye digrams of these three cells. Because of the poor channel estimation, it is not surprising to see DDCE-DFE has large detection error for cell 87 and 89. While for BCE-DFE, the good channel estimation ensures good detection.

V. Conclusion

We have studied the equalization problems in TDMA wireless ATM systems. Aiming at eliminating training symbols associated with equalizations for uplink data bursts, we exploit ATM protocol information for equalization. Specifically, MAC and DLC protocols are exploited to provide known cell headers to BS's receiver and enable us to convert the uplink data burst into a block transmission structure with payload of a cell as the unknown data block and header as the known training block. Blind channel estimation is also incorporated into the protocol-aided approach. A BCE-DFE algorithm is developed to take advantage of the converted block transmission structure and blind channel estimation. For fast-fading channels, the new algorithm can overcome the channel estimation error propagation effect suffered by many conventional block equalization schemes. Simulation results show that the BCE-DFE can achieve good equalization performance without transmitting training symbols for uplink data bursts.

Appendix
Channel Basis Construction in Wireless ATM

If we assume the channel is time-invariant during a cell period, the baseband multipath channel impulse response can be described by

$$c(t) = \sum_{i=1}^{m} \alpha_i \delta(t - \tau_i) \tag{13}$$

where m is the number of rays of the channel, α_i and τ_i are gain and delay of of each ray. Denote $p(t)$ as the transmitted shaping filter, $f(t)$ as the receiver front end filter, we can have the combined channel impulse response as

$$\begin{aligned} h(t) &= p(t) * c(t) * f(t) = p_c(t) * c(t) \\ &= \sum_{i=1}^{m} \alpha_i p_c(t - \tau_i); \\ p_c(t) &= p(t) * f(t) \end{aligned} \tag{14}$$

where $*$ denotes convolution. If input information sequence is s_k, we can have the continuous-time output of the combined channel

$$y(t) = \sum s_k \sum_{i=1}^{m} \alpha_i p_c(t - kT \cdots \alpha_i) + n(t) \qquad (15)$$

where T is symbol duration and $n(t)$ is additive noise. Fig. 18 depicts such a combined channel model.

Assuming the channel spreads L symbol intervals and the receiver sample with a rate $1/T$, we have the discrete-time channel impulse response

$$\mathbf{h} = \mathcal{B}\mathbf{g} \qquad (16)$$

where

$$\mathbf{h} = \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_L \end{pmatrix}$$

$$\mathcal{B} = \begin{pmatrix} p_c(-\tau_1) & p_c(-\tau_2) & \cdots & p_c(-\tau_m) \\ p_c(T-\tau_1) & p_c(T-\tau_2) & \cdots & p_c(T-\tau_m) \\ \vdots & \vdots & \vdots & \vdots \\ p_c(LT-\tau_1) & p_c(LT-\tau_2) & \cdots & p_c(LT-\tau_m) \end{pmatrix}$$

and

$$\mathbf{g} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}.$$

For a time-varying multipath channel, α_i cay vary dramatically even within a cell's duration; but τ_i, which depends on the physical propagation environment, changes slowly comparing with transmission data rate [30]. Because of the high data rate of wireless ATM, it is reasonable to assume that τ_i remains unchanged during a data burst. Since $p_c(t)$ is known, if we are given τ_i and m, we can construct $\mathcal{B}$, i.e., the channel basis.

In this paper, we assume known m, thus only need to estimate τ_i to obtain the basis. Because wireless ATM is a connection-oriented network, there will be call setup messages exchanged between the base station and mobiles before communication starts. Mobile users send their setup messages during the uplink control subframe and these messages carry sufficient long preambles which can be used to estimate τ_i.

Although basis does not change dramatically, we still need to update it whenever possible. The source of update comes from the training symbols associated with the reservation and other control messages from a mobile. In wireless ATM, a mobile usually has to send a reservation message to the base station through the uplink control subframe before it transmits a data burst. Also from time to time, control messages from a mobile are sent to the base station through the same uplink control subframe. The receiver at the base station can use the training symbols associated with these messages to update the channel basis for that particular mobile.

REFERENCES

[1] *Interoperability and Performance Standards for Data Modems*, Sept. 1991. MIL-STD-188-110A.

[2] A. Acampora and M. Naghshineh, "An architecture and methodology for mobile executed handoff in cellular ATM networks," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1365–1375, Oct. 1994.

[3] N. Al-Dhahir and J. M. Cioffi, "MMSE decision-feedback equalizers: Finite-length results," *IEEE Trans. Inform. Theory*, vol. 41, pp. 961–975, July 1995.

[4] J. P. Aldis, M. P. Althoff, and R. Van Nee, "Physical layer architecture and performance in the WAND user trial system," in *Proc. ACTS Mobile Summit*, Granada, Spain, Nov. 1996, pp. 196–203.

[5] S. Ariyavisitakul and L. J. Greenstein, "Reduced-complexity equalization techniques for broadband wireless channels," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 5–15, Jan. 1997.

[6] E. Ayanoglu, K. Y. Eng, and M. Karol, "Wireless ATM: Limits, challenges, and proposals," *IEEE Personal Commun.*, pp. 18–34, Aug. 1996.

[7] E. Ayanoglu, P. Pancha, A. Reibman, and S. Talwar, "Forward error control for MPEG-2 video transport in a wireless ATM LAN," *ACM/Baltzer Mobile Networks and Applications J.*, vol. 1, pp. 235–244, Dec. 1996.

[8] F. Bauchot *et al.*, "MASCARA, a MAC protocol for wireless ATM," in *Proc. ACTS Mobile Summit*, Granada, Spain, Nov. 1996, pp. 647–651.

[9] S. Crozier *et al.*, "Reduced complexity short-block data detection techniques for fading time-dispersive channels," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 255–265, Aug. 1992.

[10] D. M. J. Devasirvatham, "Time delay spread and signal level measurements of 850 MHz radio waves in building environments," *IEEE Trans. Antennas Propagat.*, vol. AP-34, pp. 1300–1305, Nov. 1986.

[11] K. Y. Eng *et al.*, "A wireless broadband ah-hoc ATM local-area network," *ACM/Blatzer Wireless Networks J.*, vol. 1, no. 2, pp. 161–174, 1995.

[12] F. M. Hsu, "Data directed estimation techniques for single-tone HF modems," in *Proc. IEEE MILCOM'85*, pp. 12.4.1–12.4.10.

[13] G. K. Kaleh, "Channel equalization for block transmission systems," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 110–121, Jan. 1995.

[14] R. Hasholzner, C. Drewes, and J. S. Hammerschimidt, "Multiple access and equalization issues for 156 Mbps ATM radio in the local loop," in *Proc. ACTS Mobile Summit*, Aalborg, Denmark, Oct. 1997.

[15] G. Kalivas, M. El-Tanany, and S. Mahmoud, "Channel characterization for indoor wireless communications at 21.6 GHz and 37.2 GHz," in *Proc. Int. Conf. Univ. Personal Commun.*, Ottawa, Ontario, Canada, Oct. 1993, pp. 626–630.

[16] M. Karol, Z. Liu, and K. Eng, "Distributed queuing request update multiple access (DQRUMA) for wireless packet (ATM) Networks," in *Proc. Int. Conf. Commun.*, Seattle, WA, June 1995, pp. 1224–1231.

[17] B. J. Kim and D. Cox, "Blind equalization for short burst communications over frequency selective wireless channels," in *Proc. VTC*, vol. 2, Phoenix, AR, May 1997, pp. 544–548.

[18] P. Narasimhan *et al.*, "Design and performance of radio access protocols in WATMnet, a prototype wireless ATM network," presented at the Proc. ICUPC97 Conf., San Diego, CA, Oct. 1997.

[19] B. C. Ng, M. Cederval, and A. Paulraj, "A structured channel estimator for maximum-likelihood sequence detection," *IEEE Commun. Lett.*, vol. 1, pp. 52–55, Mar. 1997.

[20] D. Petras, "Medium access control protocol for wireless, transparent ATM access," in *IEEE Wireless Commun. Syst. Symp.*, Long Island, NY, Nov. 1995, pp. 79–84.

[21] J. Porter *et al.*, "The ORL Radio ATM System, Architecture and Implementation,", vol. 96.5, ORL Tech. I Rep., Jan. 1996.

[22] D. Raychaudhuri *et al.*, "WATMnet: A prototype wireless ATM system for multimedia personal communication," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 83–95, Jan. 1997.

[23] D. Raychaudhuri and N. D. Wilson, "ATM-based transport architecture for multiservices wireless personal communication networks," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1401–1414, Oct. 1994.

[24] R. Steele, Ed., *Mobile Radio Communications*. New York: IEEE Press, 1992.

[25] L. Tong and J. Bao, "Equalization in wireless ATM," presented at the Proc. 35th Allerton Conf. Communication, Control, and Computing, Monticello, IL, Oct. 1997.

[26] L. Tong and S. Perreau, Multichannel blind channel estimation: From subspace to maximum likelihood methods, in IEEE Proc., 1998. to appear in (AUTHOR: UPDATE?).

[27] J. Walrand and P. Varaiya, *High-Performance Communication Networks*. San Francisco, CA: Morgan Kaufmann, 1996.

[28] C. C. Watterson, J. T. Juroshek, and W. D. Bensema, "Experimental confirmation of an HF channel model," *IEEE Trans. Comman. Technol.*, vol. 18, pp. 792–803, Dec. 1970.

[29] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *IEEE Trans. Signal Processing*, vol. 43, pp. 2982–2993, Dec. 1995.

[30] H. Zeng and L. Tong, "Blind channel estimation using the second-order statistics: Algorithms," *IEEE Trans. Signal Processing*, vol. 45, pp. 1919–1930, Aug. 1997.

Jeffrey Q. Bao (S'96) received the B.S. and M.S. degrees in electrical engineering in 1991 and 1994, respectively, both from Shanghai Jiao Tong University, Shanghai, China. He is currently a Ph.D. candidate at the Department of Electrical Engineering, University of Connecticut, Storrs.

From February 1994 to August 1996, he was a System Engineer with Motorola (China)'s Cellular Infrastructure Group. Since August 1998, he has been a Visiting Student at the School of Electrical Engineering, Cornell University, Ithaca, NY. His research interests include signal processing for wireless communication systems, medium access control in wireless packet networks, and performance analysis of such systems.

Lang Tong (S'87–M'91) received the B.E. degree from Tsinghua University, Beijing, China, in 1985, and the M.S. and Ph.D. degrees in electrical engineering in 1987 and 1990, respectively, from the University of Notre Dame, Notre Dame, IN. He was Postdoctoral Research Affiliate at the Information Systems Laboratory, Stanford University, Stanford, CA, in 1991.

Currently, he is an Associate Professor in the School of Electrical Engineering, Cornell University, Ithaca, NY. Prior to his joining Cornell, he was with West Virginia University, Morgantown, and later with the University of Connecticut at Storrs. His research interest includes statistical signal processing, adaptive receiver design for communication systems, signal processing for communication networks, and system theory.

Dr. Tong received Young Investigator Award from the Office of Naval Research in 1996, and the Outstanding Young Author Award from the IEEE Circuits and Systems Society.