REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
12-04-2011

2. REPORT TYPE
Technical Paper

3. DATES COVERED *(From - To)*
MAR 2011 - APR 2011

4. TITLE AND SUBTITLE
A Peak To Peak Frame Synchronization Algorithm For Data Frames Transmitted Asynchronously Over Fading Channels

5a. CONTRACT NUMBER
FA8720-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Ryan Shoup and Nancy List

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force
2350 El Segundo Blvd
El Segundo, CA 90245-2807

10. SPONSOR/MONITOR'S ACRONYM(S)
SMC/MCX

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Space-division multiple access communication system architectures employing frame-switching, where ingress frames from one of N ingress ports are switched to one of N egress ports, can support the simultaneous transmission of data from multiple users. If the switched inputs have undergone channel fading, the frame switch may (incorrectly) switch ingress frames to (incorrect) egress ports. When frame switch errors occur, the egress port data streams may contain an incorrect number of data frames and/or erroneous frame sequence numbers resulting in frame synchronization problem at the end receiver associated with that particular egress port. A maximum-likelibood algorithm is employed to run at the end receiver to correctly recover frames and corresponding frames sequence numbers for proper frame synchronization.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: U			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Zach Sweet
U	U	U	SAR	5	**19b. TELEPHONE NUMBER** *(include area code)* 781-981-5997

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

A Peak To Peak Frame Synchronization Algorithm For Data Frames Transmitted Asynchronously Over Fading Channels

Ryan Shoup and Nancy List
MIT Lincoln Laboratory
Lexington, MA 02420

***Abstract*— Space-division multiple access communication system architectures employing frame-switching, where ingress frames from one of N ingress ports are switched to one of N egress ports, can support the simultaneous transmission of data from multiple users. If the switched inputs have undergone channel fading, the frame switch may (incorrectly) switch ingress frames to (incorrect) egress ports. When frame switch errors occur, the egress port data streams may contain an incorrect number of data frames and/or erroneous frame sequence numbers resulting in frame synchronization problems at the end receiver associated with that particular egress port. A maximum-likelihood algorithm is employed to run at the end receiver to correctly recover frames and corresponding frames sequence numbers for proper frame synchronization.**

I. INTRODUCTION

A space-division multiple access (SDMA) system accommodates multiple users by demodulating the users' signals which are spatially separated at the input to a demodulator (or bank of demodulators, where there could be one demodulator per spatial input). One example of a SDMA system could be a terrestrial radio frequency (RF) communications repeater where the communications repeater contains multiple directional antennas. Each of the directional antennas is directed at a different user allowing the system to support multiple users. The terrestrial RF communications repeater could be modeled as a fading channel [1]. Another example of a SDMA system is a satellite system, where the inputs to the satellite are narrow RF or optical beams. The RF/optical beams have such narrow beamwidths that they are spatially separated and can support multiple users in a SDMA scheme. Optical channels typically are modeled as fading channels when the optical signals propagate through the atmosphere [2]. RF satellite channels can be modeled as fading channels when the terminals are at low elevation angles [3]. In this paper, we consider SDMA systems supporting multiple users propagating signals through independent and spatially separated fading channels.

A block diagram for the SDMA system considered in this paper is shown in Fig. 1. In Fig. 1, we consider N users labeled User 1, ..., User N. The N users propagate signals through independent and spatially separated fading channels.

The signals are then received by a "demodulating hub" which is assumed to demodulate the signal, regenerate the signal and then propagate the signal to a final receiver designated as the "end user" in Fig. 1. The links are bi-directional such that the end user propagates signals to each of the N users in the opposite, (i.e. forward), direction.

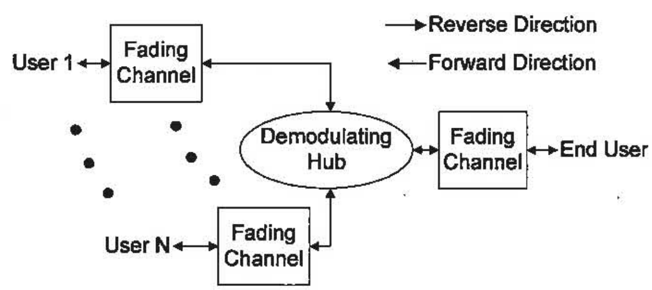

Figure 1: SDMA System Block Diagram

Signals from the N users, 1…N, in Fig. 1 are assumed to be framed such that the demodulating hub can demodulate the user frames and attain frame synchronization from each of their transmitted signals. Next, the hub switches the frames to a common output destination port connected to a second fading channel that terminates at the end user. As the link is bi-directional, the end user can transmit frames in the opposite direction. In this case, the demodulating hub separates out frames, using information in the frame headers, for each of the N users and forwards them accordingly.

The fading channel often corrupts the frame header to the extent that the frames cannot be identified by the demodulating hub. When a particular channel is in a deep fade, frame identification by the demodulating hub thus becomes unreliable. Reliable frame detection is required both for correct frame switching and for synchronization purposes. Deinterleavers, for example, may have frame synchronization requirements. If the demodulating hub misidentified a frame transmitted by the end user to the i^{th} user shown in Fig. 1, it is possible that the deinterleaver associated with that i^{th} user would not only have an incorrect number of frames but also frames in the incorrect order. The output of the deinterleaver would be in error from the time that the first input to the deinterleaver was out of sequence. It is desirable to design an

This work was sponsored by the Department of the Air Force under Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

algorithm that is robust in the sense that it can detect lost or misidentified frames such that synchronization is maintained.

This paper will examine the performance of a frame synchronization algorithm suitable for the system shown in Fig. 1. Section II discusses the communications systems channel model. Section III discusses the frame synchronization algorithm. Section IV provides simulation and analysis. Finally, section V provides the conclusions.

II. CHANNEL MODEL

A block diagram of the communication channels shown in Fig.1 is provided in Fig. 2. These communication links consists of either a bi-directional link between the ith user and the hub or the end user and hub.

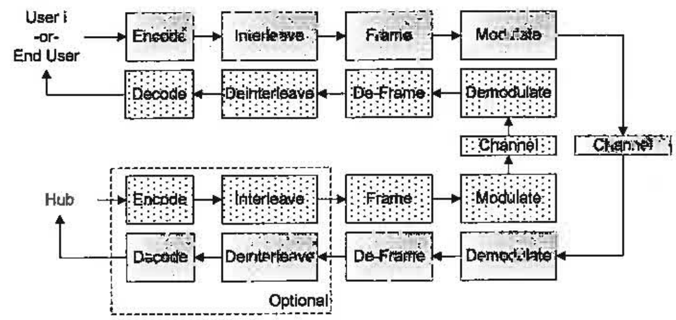

Figure 2: Physical Layer Model for the ith User

As shown in Fig. 2, the ith user encodes, interleaves, and frames user data. The data is then modulated before retransmission to the hub. The hub demodulates and deframes the data. At this point, the hub can either deinterleave and decode the data, or simply forward the frames to the end user where deinterleaving and decoding would take place. In either case, frame synchronization is needed for correct deinterleaving and decoding. In the former case, the frame synchronization process is performed at the hub and only needs to correct for errors that occur over a single hop or fading channel. The latter case could be used to simplify the design of the hub, which might be appropriate for a hub located on a satellite. In this case the hub forwards frames from the output of the de-framer directly to the output modulator. And, the deinterleaving and decoding is assumed to be done by the end user (or the ith user for traffic propagating in the opposite direction). When the deinterleaving is done in this "end to end" manner, the frames presented to the deinterleaver at the end user are corrupted by multiple independent fading channels, rendering the job of differentiating upon user frames more difficult. In the forward direction, the hub makes individual decisions on frames then demultiplexes the input user's frames to the user's output port. In this case, two fading channels are encountered with a per-frame hard decision performed in between. We consider only the former case in this paper, where the hub processes the data and implements a frame synchronization algorithm.

The framing structure used in all of the communications links is provided in Fig. 3. The first component is the "unique word." Each of the N users in the system receives a different unique word. In the reverse direction, the unique word is used by the end user to determine which of the N users a particular frame came from in the forward direction. Correlations are performed at the receivers in the hub, the N user receivers, and at the end user. The unique word that has the highest correlation with a particular frame is used to map the frames to the different users. It is assumed that steady state tracking conditions and symbol timing has been achieved during an acquisition phase so that the frame boundaries are already known. In this state, the only thing unknown is which user a particular frame belongs to at the receiver.

To facilitate an asynchronous design, the N users and end user are actually assigned two unique words. The second unique word is associated with a fill frame. Fill frames allow frames to be added and dropped at the hub which simplifies the switching of user data frames since the different users can transmit asynchronously. The fill frames allow for a constant bit rate for the links from the N users to the hub and a constant bit rate for the links from the hub to the end user. With fill frames, the N user data rates can be arbitrary and asynchronous in both the forward and reverse directions, while the actual line rate on the links remain constant.

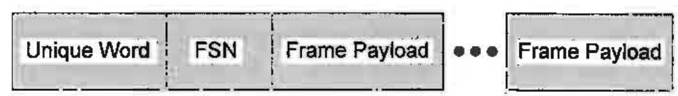

Figure 3: Frame Structure

Also shown in Fig. 3, is the presence of a frame sequence number (FSN). Each of the users is assigned a starting FSN. For each subsequent frame transmitted at each of the transmitters, the FSN is incremented. Thus the unique word is used to differentiate frames from the different users, and the FSN is used to determine the sequence of frames for a particular user. This sequence of frames is then provided to downstream communications functions such as the deinterleaver. The deinterleaver recovers the frame payload shown in Fig. 3 before sending the data to the decoder. Decoding and deinterleaving are a well known means of mitigating the effects of channel fading [2].

A. Channel and Corresponding System Assumptions

Figure 4 provides an illustration of the fading channel model used in this analysis. The channel routinely fades to levels of -10 dB and occasionally fades to levels of -15 dB. The inset diagram of Fig. 4 provides a histogram of the frequency of the channel fades. For the purposes of frame identification and reliable communications, the following assumptions are assumed: (a) T_{COH}>>F_D, and (b) I_D >> T_{COH}, where T_{COH} is the coherence time of the Channel, F_D is the frame duration, and I_D is the interleaver delay span. These assumptions describe a system where each frame sees the same fade level and each bit in the deinterleaved codeword is uncorrelated. Therefore, the channel gain level seen by each individual frame header is constant. Another assumption is that the number of unique word bits (denoted UW_{bits}) >> N. The unique words themselves can either be obtained via output of a random number generator or via codewords in a codebook, such as a BCH code. In either case, the distance between codewords $\approx N/2$ when $UW_{bits} >> N$. This assumption is reasonable and results in a negligible bandwidth penalty when $UW_{bits} << F_{bits}$, where F_{bits} is the total number of bits within an entire frame. Using this assumption on Hamming distance, estimates of per-frame misidentification probabilities can be used to set the frame synchronization algorithm parameters for a given communication channel.

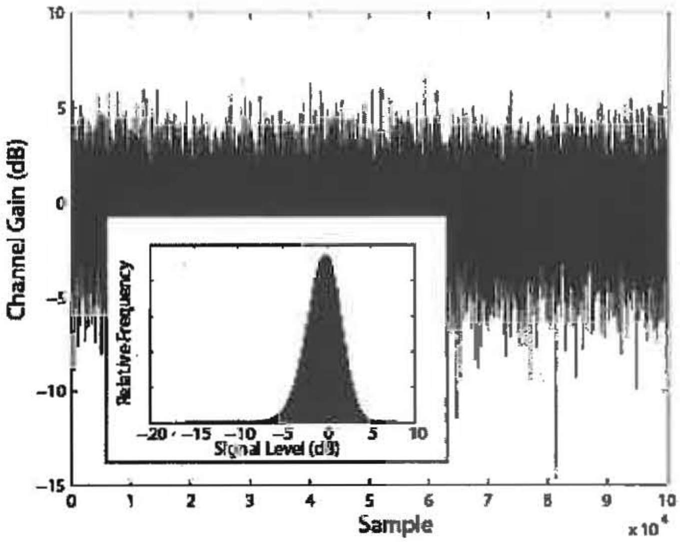

Figure 4: Channel Fading

B. *Summary of Channel Induced Framing Errors*

In the reverse direction, frames (both user data and fill) are transmitted from each of the N users to the hub shown in Fig. 1. When the frames are received at the hub from each spatially separated user, frame errors occur when user frames are misidentified as fill frames and vice-versa. The hub performs deinterleaving and decoding then re-interleaves, re-encodes and sends the frames to the end user.

In the forward direction, the hub can also misidentify fill frames and user frames. Since frames destined to N users are transmitted between the end user and the hub in the forward direction, the hub can erroneously decide that a frame destined for user N-1 is actually destined for user N. These misidentifications may occur when the channel undergoes deep fades which render the unique words and FSNs contained within the frame headers unreliable. In the following section we will describe an algorithm which recovers frame identification and sequencing in the presence of severe channel fading.

III. Peak to Peak Frame Synchronization Algorithm

A. *Overview*

The peak to peak algorithm is an algorithm that attempts to determine the most likely sequence of frames for a particular user in the interval between two "peaks" in the fading channel. Peaks are defined as the points at which channel levels ≥ 0 dB and between which the channel levels ≤ 0 dB. Frames are accumulated in a buffer between peaks. No attempt is made to process frames between peaks.

The peaks themselves denote reliable endpoints. Since the channel level at a peak is at least 0 dB, both the FSN and unique word are assumed to be reliably decoded. At these endpoints, careful selection of UWs to provide good Hamming distance between UWs and a frame error correction code applied to the FSNs ensure that the FSN and UW can be recovered at signal levels ≥ 0 dB.

Once two peaks have been identified, the most likely sequence of frames transmitted between them can be determined. The first peak provides the starting FSN and the second peak provides the ending FSN of the sequence of frames to be recovered. Once the starting and ending FSNs are known, the most likely transmitted sequence of frames given the frames in the buffer can be determined. These frames are read out of the buffer and the FSNs are refreshed based on the results of the algorithm, and these refreshed FSNs are applied to the frames. The peak to peak approach inherently provides some controls to ensure that the algorithm is robust and limits the duration to which the algorithm would run over.

B. *Peak to Peak Algorithms*

The goal of the peak-to-peak algorithm is to find the sequence of frames, $\overline{s}$, that maximizes the probability $P(\overline{r} \mid \overline{s})$. The sequence consists of frame header correlation values for both the "peak frames" received before and after the channel fading occurs, and the frame header correlation values for the frames received during the channel fade between the peaks. Correlations with both the unique word and the FSN can be used in creating the frame header correlations. The sequence, $\overline{s}$, is described in terms of its unique word and its FSN, denoted $\overline{u}$, and $\overline{f}$, respectively.

1) Peak-to-Peak Maximum Likelihood Algorithm

Consider the nominal case where User 1 has transmitted four user frames to the hub, and the hub fully processes the received signal (i.e., performs frame synchronization, deinterleaving and decoding). The first and fourth frames register as "peaks" with channel gain >= 0dB. Therefore, the unique words contained in these frames correlate highly with User 1's unique word. The second and third frames between the peaks experience channel fading < 0dB. The FSN for the first frame decodes to FSN=0; the FSN for the fourth frame decodes to FSN=2. Since there were two additional frames received at the hub between the peaks, one of them must have been a fill frame. In this simple example, there are only two possibilities: the fill frame is the second frame in the buffer, or the fill frame is the third frame in the buffer. Equation (1) below gives the probability that the fill frame is the second frame in the buffer and (2) is the probability that the fill frame is the third frame in the buffer

$$P^1\left(\overline{r}\middle|\overline{u},\overline{f}\right) = P\left(r_1\middle|u^1,f_0\right)\cdot\sum_k P\left(r_2\middle|u^f,f_k\right)\cdot P\left(r_3\middle|u^1,f_1\right)\cdot P\left(r_4\middle|u^1,f_2\right)\cdot \quad (1)$$

Identify applicable sponsor/s here. *(sponsors)*

$$P^2(\bar{r}|\bar{u},\bar{f}) = P(r_1|u^1,f_0)\cdot P(r_2|u^1,f_1)\cdot \sum_k P(r_3|u^f,f_k)\cdot P(r_4|u^1,f_2)\cdot \quad (2)$$

where r_i is the i^{th} received frame header, u^1 is User 1's unique word, u^f is the fill frame unique word, and f_k is the k^{th} FSN. The most likely sequence vector $\bar{s}$ is then chosen based on which sequence maximizes the probability $P(\bar{r} \mid \bar{s})$. Note that the FSN of a fill frame is assumed to be random, so the correlation with the fill frame FSN is the average correlation over all possible FSNs (hence the summation over k).

Equation (3) gives the number of possible sequences in the buffer, where T is the difference between the FSN of the second peak and the FSN of the first peak, and U_F is the number of fill frames transmitted (U_F = total number of frames received minus (T+1)). In this example, only two sequences, P^1 and P^2, are possible.

$$\text{Number of Sequences} = \binom{T-1}{U_F}. \quad (3)$$

In general, the maximum likelihood sequence for the transmitted frames is given in (4), where the maximization is done over all of the possible sequences that could be contained in the buffer.

$$\text{ML Seq} = \max_j \left(P(\bar{r}|\bar{s}_j)\right)\cdot \quad (4)$$

There are a few computational difficulties in determining the true maximum likelihood sequence. Firstly, the number of sequences can be large which makes a direct computation of (4) impractical. Secondly, a direct computation of (4) requires the summation over all possible FSNs for each of the fill frames which may be either large or unknown.

To simplify the peak-to-peak algorithm, the correlation with fill frames can rely on the correlation with fill frame unique words and ignore the FSN correlations as shown below:

$$\sum_k P(r_i|u^f,f_k) \approx P(r_i|u^f)\cdot \quad (5)$$

This simplification to the maximum likelihood sequence estimation is a modified "Viterbi" [4] or "trellis" peak-to-peak algorithm.

2) Peak-to-Peak Trellis Algorithm

To illustrate the use of the Viterbi algorithm in determining frame sequencing, consider the forward link from the end user to the hub in the nominal case where the number of users in the system is N=2. In this example, the demodulating hub needs to differentiate fill frames and data frames for the two different users. We can use a trellis to help determine which of the frames received at the hub during the channel fade are for User 1 and which of the frames were fill frames or frames for User 2. Fig. 5 shows a trellis that represents the possible states between the peaks of the fading channel for User 1.

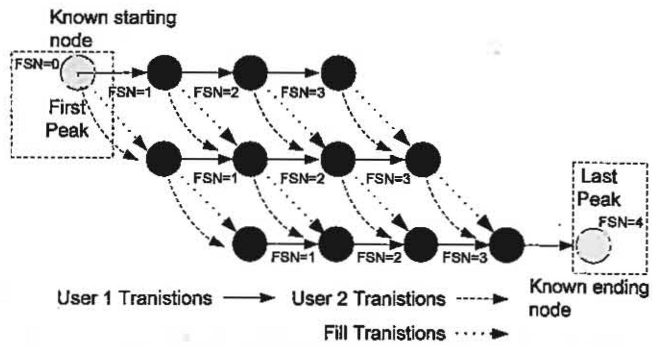

Figure 5: Example Viterbi Trellis

The state of the trellis before and after the channel fade is known. Horizontal branches in the trellis indicate the path corresponding to a frame being received for User 1 during a channel fade. Diagonal branches in the trellis correspond to paths taken when either a fill frame or a frame for User 2 is received. The horizontal paths are weighted using the correlation of the received frame with the unique word and FSN for User 1 that would be expected at that location in the trellis. The diagonal paths use the simplified correlation described in (5).

The example shown in Fig. 5 is for a case where seven frames were transmitted during a channel fade (i.e., between peaks) and the FSN for User 1 before the fade (i.e., first peak) was zero and the FSN for User 1 after the fade (i.e., second peak) was four. Using the FSNs at the peaks, we calculate that five User 1 data frames were transmitted during the channel fade. The other two frames transmitted during the fade were either fill frames or User 2 data frames.

For each node in the trellis, the "edges" entering a node can be pruned by eliminating the path that results in a higher overall path weight from the beginning of the trellis to that particular node. With the edges pruned according to path weight, the best path is the path that goes from peak-to-peak within the trellis

3) Simplified Peak-to-Peak Trellis Algorithm

A simplification to the peak-to-peak trellis algorithm is to use the FSNs at the peak points and ignore the FSN correlations for all transitions within the trellis. This approach simplifies the trellis computations considerably since it only requires correlating the received frames with the possible user and fill frame unique words. Performance simulations described in the following section show that this simplification produces useful, but suboptimal, results since a significant amount of information contained in the FSNs is not used.

4) Max Unique Word Correlation Peak-to-Peak Algorithm

The simplest method for recovering user frames relies simply on picking the frames with the maximum correlation to the each user's unique word. Using the FSNs at the peaks, the number of user frames received during a channel fade is known, and that number of frames is chosen from the received frames with the highest correlation to the user's unique word.

IV. Simulation and Analysis

For the simulations, the following parameters were used: $UW_{bits} = 256$, Encoded $FSN_{bits} = 256$, Number of Users = 10, Link = End User to Hub (Figure 1). The channel model employed is the channel shown in Figure 4. The modulation considered was differential phase shift keying (DPSK) which is non-coherently demodulated and hard decision decoded. The ratio of the channel coherence time to the frame duration, $T_{COH}/F_D \approx 10$. The following algorithms described in the previous section were simulated:

1. Peak-to-peak maximum likelihood algorithm including the averaging over FSNs for fill frames
2. Peak-to-peak trellis algorithm including FSN correlations for user frames but using the approximation in (5) for the alternate user data frames and fill frames
3. Peak-to-peak trellis algorithm using only unique word correlations
4. Maximum unique word correlation peak-to-peak algorithm without the trellis

In addition, the following algorithms were simulated as points of comparison:

5. Peak-to-peak algorithm with no attempt to optimize user data frame recovery
6. Genie-aided peak-to-peak trellis algorithm where the FSNs of alternate users and fill frames are known within the trellis

Peak-to-peak algorithms (1-4) were described in detail in Section III. The peak-to-peak algorithm described in (5) simply locates peaks, determines the number of user frames (M) between the peaks, and designates the first M frames received between the peaks to be the user's data frames. All the remaining frames in between the peaks are assumed to be fill frames or alternate user frames. This is perhaps the simplest approach possible. Finally, the genie-aided algorithm described in item 6 is not a practically implementable algorithm, but is useful as a point of comparison. This algorithm should perform the best since more information is available in the receiver in this case. This algorithm shows how well a multi-user synchronization algorithm might perform if the synchronization process attempted to jointly optimize frame recovery for all users rather than separately for each single user.

Fig. 6 shows the performance for each of these algorithms in terms of the frame identification error rate as a function of the number of photons per frame symbol.

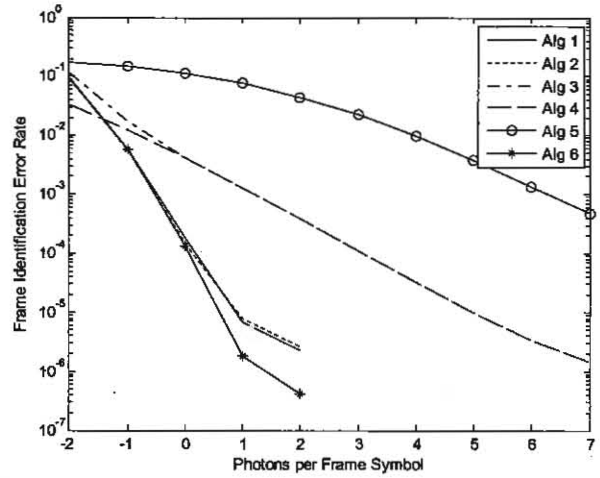

Figure 6: Algorithm Performance Comparison

V. Conclusions and Future Work

The peak-to-peak algorithms (1) and (2) listed in Section IV and described in Section III perform the best. Peak-to-peak algorithm (1) is impractical for systems where a large set of FSNs would need to be considered. Fortunately, the approximation employed in the peak-to-peak algorithm (2) provides similar performance. A significant simplification to the peak-to-peak algorithm is obtained by eliminating the correlations with FSNs between the peaks. Peak-to-peak algorithms (3) and (4) make use of this simplification. Their performance shown in Fig. 6 is not as good as that of peak-to-peak algorithms (1) and (2), but depending on a particular channel's fading statistics it may be suitable for some communication links. Finally the performance of the simplest peak-to-peak algorithm (5) shown in Fig. 6 is only suitable for benign channel fading conditions.

Future work includes investigation of the case where the hub itself does not perform the frame synchronization algorithms and pushes this synchronization to the end users. This results in a multi-hop fading synchronization problem which was discussed in Section II. The overall end-to-end bit-error rate performance for specific interleaver and coding parameters can also be investigated for this case.

References

[1] M.E. Humphrey, "Design considerations for mobile wideband wireless network architectures," Milcom 97 Proceedings, Nov 1997.

[2] B. Moision and A. Biswas, "Memory Requirements to Mitigate Fading Losses on an Optical Channel," IPN Progress Report, 42-170, August

[3] W.J. Vogel and J.Goldhirsh, "Multipath fading at L band for low elevation angle, land mobile satellite scenarios," IEEE Jornal on Selected Areas in Communications, Vol 13 No 2, February 1995.

[4] A. J. Viterbi, "A Personal History of the Viterbi Algorithm," Signal Processing Magazine, IEEE Vol 23 No 4,