Compressive Sensing based Asynchronous Random Access for Wireless Networks

Vahid Shah-Mansouri*, Suyang Duan*, Ling-Hua Chang†, Vincent W.S. Wong*, and Jwo-Yuh Wu†
*Department of Electrical and Computer Engineering, University of British Columbia, Canada
†Department of Electrical Engineering, National Chiao Tung University, Taiwan
Emails: *{vahids, suyangd, vincentw}@ece.ubc.ca, †iamjaung@gmail.com, jywu@cc.nctu.edu.tw

***Abstract*—The theory of compressive sensing has shown that with a small number of samples from random projections of a sparse signal, one can recover the original signal under certain conditions. In this paper, we use compressive sensing to design a random access protocol for requesting uplink data channels. A wireless node transmits a pseudo-random sequence to an access point (AP) when it requires an uplink channel. The AP receives multiple sequences in a random access shared channel. Due to different propagation delays, the received signals from different wireless nodes are not synchronized at the receiver. Assume that the number of sequence transmissions is substantially less than the number of wireless nodes in the system. Under such circumstances, we design an asynchronous compressive sensing based decoder to recover the original signals in a random access setting. The key difference between our proposed decoder and those presented in the literature is that we do not require any synchronization before sequence transmission which makes our approach practical. Simulation results show the throughput improvement of our proposed scheme compared to two other random access protocols.**

I. Introduction

The compressive sensing has recently received an increasing attention for sparse data recovery in random settings [1], [2]. Compressive sensing is based on the theory that a small number of random projections of a sparse signal can contain most of its information such that they can be used to recover the original signal. The sparsity requirement is the main factor in this theory. Consider signal $\mathbf{x}$, which can be viewed as an $N \times 1$ column vector in $\mathbb{R}^N$. Any signal in $\mathbb{R}^N$ can be represented in terms of a set of basis $\{\psi_1, \ldots, \psi_N\}$. Using the $N \times N$ matrix $\Psi = [\psi_1 \ldots \psi_N]$, one can represent $\mathbf{x}$ in terms of the set of basis as $\mathbf{x} = \Psi\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^N$ is the vector of weighting coefficients. Signal $\mathbf{x}$ is K-sparse if vector $\mathbf{s}$ has no more than K non-zero elements. Now consider the projection of signal $\mathbf{x}$ to a new space using vectors $\phi_1, \ldots, \phi_M \in \mathbb{R}^N$. The new signal $\mathbf{y} = \Phi\mathbf{x}$, where $\Phi = [\phi_1 \ldots \phi_M]$ has a lower dimension than signal $\mathbf{x}$ if $M < N$. At the first glance, it is not possible to recover signal $\mathbf{x}$ using signal $\mathbf{y}$ with M measurements. However, when matrix Φ satisfies the restricted isometry property (RIP) [1], compressive sensing assures that the random projections can preserve the information of the original sparse signal. Hence, one can accurately approximate the signal from M measurements using a recovery method. For random Gaussian column vectors $\phi_1, \ldots, \phi_M$, it is shown that the exact recovery is achieved with high probability using compressive sensing if M is on the order of $K \log(N/K)$ [2].

Several algorithms have been developed in the literature of compressive sensing towards support recovery and signal reconstruction. They can be divided into optimization-based schemes and greedy algorithms. Orthogonal matching pursuit (OMP) algorithm is the most widely adopted greedy algorithm. OMP is not only simple, it is empirically competitive in terms of approximation performance [3], [4]. Optimization-based schemes include ℓ_q norm reconstruction, $q = 0, 1, 2$ [5], [6]. While ℓ_2 norm reconstruction results in a non-sparse signal and ℓ_0 norm reconstruction is NP-complete, ℓ_1 norm approach provides K-sparse solution by solving a convex optimization problem as

$$\hat{\mathbf{s}} = \text{argmin } \|\mathbf{s}\|_{\ell_1}, \text{ subject to } \Phi\Psi\mathbf{s} = \mathbf{y}. \tag{1}$$

Compressive sensing has been adopted in various fields including image compression, data streaming, medical signal processing, and digital communications. In communication systems, compressive sensing has been used in different areas. In sensory systems, the compressive sensing is employed vastly for data collection. In such schemes, a fusion center can fully recover the signal field information from the set of measurements of spatially distributed sensor nodes [7], [8]. In [9], a random access compressive sensing protocol is proposed for underwater sensor networks. The goal is to design a power-efficient random data collection scheme. The underlying assumption is that most physical phenomena have sparse representations in the frequency domain. In [10], a compressive sensing based medium access control (CS-MAC) protocol is proposed. The access point (AP) assigns a random sequence to each node. The requests for granting uplink transmissions are sent simultaneously by various users in a synchronous manner. Similar algorithms have been proposed in [11] and [12], where compressive sensing is employed to recover sparse simultaneous sequence transmissions. The key assumption in the design of these algorithms is synchronous request transmissions.

In this paper, we relax the assumption of synchronous sequence transmissions. We consider a scenario where there are multiple wireless users and one serving AP. Wireless nodes try to obtain uplink access via a random access mechanism. Although the maximum possible round trip delay from the AP to the wireless nodes is known, the information about the location and the actual round trip delay for individual nodes is not available at the AP. In this paper, we first

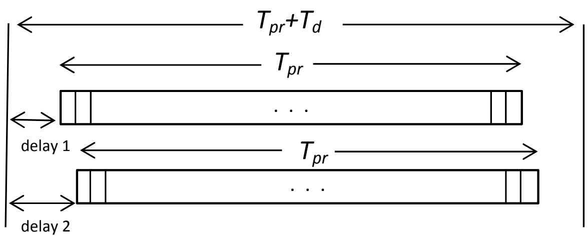

Fig. 1. Preamble transmission in a random access shared channel.

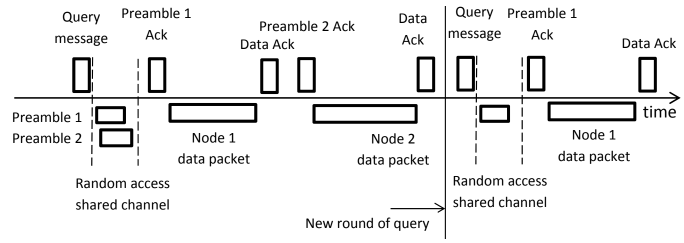

Fig. 2. The asynchronous random access protocol.

design a compressive sensing based random access method. Then, we use the idea of compressive sensing to design a decoder which can simultaneously receive multiple asynchronous connection requests and decode them successfully. Wireless nodes randomly select sequences called preambles from a pre-determined set and request an uplink connection by transmitting the preambles simultaneously at certain time intervals. The preambles are combined on the air and received as a single signal at the receiver. The simulation results show that our proposed protocol can achieve better performance than CS-MAC [10] and carrier sense multiple access with collision avoidance (CSMA/CA) protocol. The main difference of our work compared to the works in [10]–[12] is the relaxation of synchronization assumption. To this end, we present a compressive sensing random access protocol which is more practical and can be applied to a wider range of applications by relaxing the synchronization assumption used in [10]–[12].

The rest of this paper is organized as follows: The system model presented in Section II introduces the random access model. In Section III, we present our compressive sensing decoder. Performance evaluation and comparison are presented in Section IV. Conclusions are given in Section V.

II. System Model

In this section, we present our compressive sensing based random access method. We consider a wireless local area network with U wireless users and an AP as server. All the wireless nodes connect as clients to the AP for Internet access or file exchange. There is only one AP in the system and all the wireless nodes are within range of this AP. In this paper, we only focus on the uplink scenario. However, the random access mechanism can be used to initiate and request downlink access in a similar manner.

A. Asynchronous Random Access

Wireless nodes are distributed around an AP. They compete to obtain uplink access via a random access method. This is carried out by transmission of preambles at certain time instances. At certain time intervals which we call *random access shared channel*, wireless nodes are allowed to send their preambles to request an uplink transmission. Such time interval is shared between all the users. It is announced by the AP and lasts for a certain time duration. The AP decodes the received signal and assigns uplink channels for all the requesting nodes. When all the nodes successfully transmit their packets, the AP announces a new random access shared channel by transmitting a new query message.

Since the wireless nodes are not synchronized, the preambles transmitted by them are received with different delays after the query is transmitted. The amount of the delay depends on the round trip propagation delay from the wireless node to the AP. It also depends on the processing delay of the wireless node between receiving the query and transmission of the preamble. We assume that the processing delay is identical for all the nodes. Since it is a constant parameter, without loss of generality, we do not consider its effect in our model. Assuming that the preamble transmissions last for T_{pr} μs and the maximum round trip delay in the network is T_d μs, the random access shared channel lasts for $T_{pr} + T_d$ μs. Fig. 1 shows simultaneous transmission of two preambles in a random access shared channel with different delays.

In our compressive sensing based random access algorithm, the AP broadcasts a query message to declare the start of random access shared channel. Upon receiving the query message, each wireless node randomly selects a preamble from the available set of preambles and transmits that preamble with probability p. The parameter p is announced by the AP at the query message and it is used to control the number of wireless nodes requesting network access in order to assure the sparsity of the received signal. The AP then decodes the received preambles using a compressive sensing based decoder described in details in the following subsection. After decoding the received preambles, the receiver starts acknowledging those wireless nodes with sending the recovered preambles as Ack packet. Since the preambles are randomly chosen by the wireless nodes, the AP can only acknowledge the preamble not the device ID. If a wireless node receives an acknowledgement (Ack packet) containing its preamble from AP, it assures that the AP has successfully detected its preamble and allocated a channel for its transmission. The wireless node transmits its data packet right after receiving the preamble. The AP acknowledges reception of the data packet by transmitting a data acknowledgement packet. If the preamble or the Ack packet is lost, then the node will try again in the next round.

Fig. 2 shows the operation of the random access protocol and packet transmission procedures for two query messages. In the first query message, two nodes transmit their preambles and consequently their data packets. Then, the query message is transmitted again and in the new round only one of the nodes transmits its preamble. If two nodes transmit the same preamble with different delays, then the AP detects the collision and discards both. However, if both preambles are received at the same time, the AP cannot detect the collision and it acknowledges the preamble. Since both nodes receive the acknowledgement, the collision occurs for the packet transmission. In this case, none of these nodes receive an acknowledgement for their data transmission. These nodes try again in the next coming random access shared channel to acquire uplink access. Moreover, it is possible that the compressive sensing decoder at the AP detects a wrong preamble. Acknowledging a wrong preamble follows no data transmission. In this case, the wireless node does not receive an acknowledgement for its preamble transmission. It is informed by receiving the next query message which shows the end of data transmissions. The wireless node tries in the next random access shared channel to obtain transmission access.

B. Compressive Sensing Decoder

The goal of compressed sensing is to capture attributes of a signal using very few measurements. In this case, compressive sensing can be used as a decoder to recover the sparse vector. The receiver of AP samples random access shared channel lasting for $T_{pr} + T_d$ μs and obtains a sequence with length $M + \gamma$. In this case, M is the preamble length and γ is the number of samples for T_d μs, which denotes the maximum round trip delay in terms of number of bits. This sequence is delivered to the decoder for recovery of the original signal. Let $\mathbf{y} \in \mathbb{R}^{M+\gamma}$ denote the delivered signal from the AP receiver to the compressive sensing decoder. Let $\mathcal{L} = \{L_1, L_2, \ldots, L_N\}$ denote the set of N preambles which users can choose between them. Each preamble is a column vector of length M. Let p_i denote the index of the preamble transmitted by node i (i.e., $L_{p_i} \in \mathcal{L}$ is the preamble chosen by i). Then, the decoder receives vector $[\mathbf{0}_{1\times\gamma_i}\ L_{p_i}^T\ \mathbf{0}_{1\times(\gamma-\gamma_i)}]^T$ from node i, where γ_i is the round trip delay of node i in terms of the number of bits and $(.)^T$ denotes the transpose of a matrix. In such case, the received vector $\mathbf{y}$ can be written as

$$\mathbf{y} = \sum_{i=1}^{U} x_i [\mathbf{0}_{1\times\gamma_i}\ L_{p_i}^T\ \mathbf{0}_{1\times(\gamma-\gamma_i)}]^T \tag{2}$$

where x_i is the received power from node i and is equal to zero if node i has remained silent in this random access shared channel. The received signal from node i is the cyclic shift of vector $[L_{p_i}^T\ \mathbf{0}_{1\times\gamma}]^T$ with shift γ_i. Since γ_i is not known to the decoder, the set of all cyclic shifts of $[L_{p_i}^T\ \mathbf{0}_{1\times\gamma}]$ are used for decoding purposes. In this case, the number of preambles is extended from N to $N(\gamma+1)$. The new set of preambles can be written using the original preambles as

$$\begin{aligned}\tilde{\mathcal{L}} \quad = \quad & \{L_1(0), \ldots, L_1(\gamma), L_2(0), \ldots, L_2(\gamma), \\ & \ldots, L_N(0), \ldots, L_N(\gamma)\}, \end{aligned} \tag{3}$$

where $L_k(l)$, $k = 1, \ldots, N$ is the vector $[L_k^T\ \mathbf{0}_{1\times\gamma}]^T$ with l cyclic shifts. To build the compressive sensing decoder, one needs to obtain the user identification system equation. The user identification system model can be obtained using the new set of preambles $\tilde{\mathcal{L}}$ in (3) as $\mathbf{y} = A\mathbf{x} + \mathbf{n}$, where A is the preamble matrix (i.e., sensing matrix), $\mathbf{x} \in \mathbb{R}^{M+\gamma}$ is the vector of received power levels, and $\mathbf{n} \in \mathbb{R}^{M+\gamma}$ is the noise vector. Vector $\mathbf{x}$ is the sparse vector, where its ith element, x_i, is equal to λ_i if node i transmits the preamble, and is zero otherwise. λ_i is the received power from node i. The sensing matrix $A \in \mathbb{R}^{(M+\gamma)\times N(\gamma+1)}$ can be written as

$$A = \begin{bmatrix} L_1 & 0 & & L_U & 0 \\ & \ddots & \cdots & & \ddots \\ 0 & L_1 & & 0 & L_U \end{bmatrix}.$$

In this paper, we adopt the orthogonal matching pursuit (OMP) algorithm, for active user identification mainly due to its relatively low complexity [3]. OMP is a greedy approach to recover information from projection of sparse signals. The OMP algorithm employed is presented in Algorithm 1.

Algorithm 1 OMP Algorithm

1: input $\mathbf{y}, \tilde{\mathcal{L}}$
2: set $\mathbf{r} := \mathbf{y}$ and $P_{sub} := [\ ]$
3: **while** true
4: find ν and C_ν as $\nu := \underset{\tilde{L}_l \in \tilde{\mathcal{L}}}{\operatorname{argmax}} \left\langle \tilde{L}_l, \mathbf{r} \right\rangle$ and $C_\nu := \max_{\tilde{L}_l \in \tilde{\mathcal{L}}} \left\langle \tilde{L}_l, \mathbf{r} \right\rangle$
5: **if** $C_\nu < \epsilon$
6: break loop
7: **end if**
8: $P_{sub} := [P_{sub}\ \ \nu]$
9: Find $\mathbf{z}$ such that $P_{sub}P_{sub}^T\mathbf{z} = P_{sub}^T\mathbf{y}$
10: $\mathbf{r} := \mathbf{y} - P_{sub}\mathbf{z}$
11: **end while**

The received signal $\mathbf{y}$ and the set of preambles $\tilde{\mathcal{L}}$ are the inputs to the algorithm. OMP algorithm is an iterative approach in finding the set of preambles in signal $\mathbf{y}$ and their corresponding power levels. The matrix P_{sub} contains the set of preambles detected so far by the algorithm. In each iteration, OMP finds the preamble ν that has the largest correlation with the residual of the received signal $\mathbf{r}$ as $\nu := \operatorname{argmax} \left\langle \tilde{L}_l, \mathbf{r} \right\rangle$ and the corresponding value of the correlation as $C_\nu := \max \left\langle \tilde{L}_l, \mathbf{r} \right\rangle$. If the correlation is less than a threshold ϵ, then the algorithm stops. Otherwise, the detected preamble ν is added at the end of matrix P_{sub} as a new column. OMP then finds vector $\mathbf{z}$ which is the corresponding power levels of the preambles detected so far and stored in P_{sub} by solving the system of linear equations of $P_{sub}P_{sub}^T\mathbf{z} = P_{sub}^T\mathbf{y}$. The remaining vector $\mathbf{r}$ is updated at Step 10 by removing the detected preambles from the received signal. We notice that in this scheme, instead of removing the contribution of each preamble at each iteration, the residual is obtained by removing the contribution of already detected preambles from the received signal. This modification will be helpful in preventing overestimating the contribution of one single preamble because we estimate the contribution of already decoded preambles together.

C. Code Selection

As dictated by compressive sensing, for exact recovery, the number of measurements (i.e., the number of samples taken by the receiver) needs to be set such that the sparsity requirements are achieved. This requires that the number of sampled bits M be substantially higher than the number of active users K. In other words, signal $\mathbf{x}$ can be recovered if it is K-sparse where $K << M$ and $K << N$. Moreover, to guarantee signal recovery, it is known that the sensing matrix A must satisfy certain sufficient conditions characterized in terms of the restricted isometry property (RIP) [1]. The sensing matrix A satisfies RIP of order K if there exists $0 < \delta < 1$ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \quad (4)$$

holds for all K-sparse x. The constant δ is called the restricted isometry constant (RIC) of the sensing matrix. For matrix A with random i.i.d. Gaussian entries, it can be shown that the RIP in (4) is satisfied with high probability if $M \geq cK\log(N/K) << N$ for a small positive constant c. This is in fact the worst case performance. In general, there is no practical algorithm for verifying whether a given measurement matrix has this property. To achieve the best performance out of the compressive sensing decoder, it is desirable to select preambles which results in low cross-correlations thereby a small RIC. Gold codes [13], which are also known as Gold sequences, are widely used in communication systems (e.g., CDMA systems and satellite navigation) because of their unique autocorrelation properties. We use such codes to achieve higher performance by using lower number of preamble (i.e., measurement) bits.

A series of Gold codes with length 2^m-1 can be constructed by using a pair of m-sequences called preferred pair. Different pairs of m-sequences can result in different series of Gold codes. In total, 2^m+1 Gold codes are generated using a preferred pair each 2^m-1 bits long. The cross-correlation function (with various cyclic shifts) for any pair of sequences in the family of Gold sequences generated with a given preferred pair was proven by Gold [13] to be three-valued with possible values $\{-1, -\pi(m), \pi(m)-2\}$ where $\pi(m)$ is

$$\pi(m) = \begin{cases} 2^{(m+1)/2}+1, & \text{for odd } m \\ 2^{(m+2)/2}+1, & \text{for even } m. \end{cases} \quad (5)$$

The Gold codes from different Gold code groups have, however, poor correlation properties. The peak value of the autocorrelation function is 2^m-1. For a Gold code with size 2^m-1, the maximum off-peak cross correlation with circular shifts is $\pi(m)$. The new set of preambles $\tilde{\mathcal{L}}$ which are zero-padded at the beginning and at the end with total of γ zeros. The upper bound for the cross correlation of the set of preambles in $\tilde{\mathcal{L}}$ would be $\pi(m)+\gamma$.

III. Performance Evaluation

In this section, we perform simulations to evaluate the performance of our proposed scheme. We consider an infrastructure-based IEEE 802.11n network. We use binary phase shift keying (BPSK) modulation for preamble transmission with the transmission rate of 6.25 Mbps. Using this transmission rate, the bit transmission interval (i.e., chip rate) is 0.16 μs and $T_{pr} = 20.32$ μs. The approximate maximum outdoor range for IEEE 802.11n standard is 250 meters. The maximum round trip delay is 1.67 μs (i.e., $T_d = 1.67$ μs) and consequently the random access shared channel is 22 μs long. Thus, the maximum delay is $\gamma = 10$ bits.

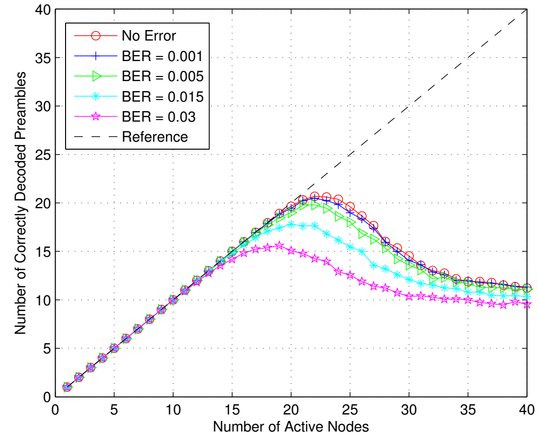

Fig. 3. The number of successful signal recovery under different number of active nodes and bit error rates.

Gold sequences are used as random access preambles. As we aim to accommodate as many nodes as possible, we try to increase the length of the sequences. However, longer sequences lead to longer preamble transmission time, and consequently a lower throughput. We choose 127 as the sequence length in consideration of both aspects. A preferred pair of m-sequence of length 127 can generate 129 Gold sequences by XORing different phases of the original m-sequences. The preamble set is further extended with 10 bits of zero-padding so that the receiver is equipped with $129(10+1) = 1419$ different preambles in the lookup table. The transmission rate of the preambles are 6.25 Mbps while the data packets and the acknowledgements are transmitted at rate 54 Mbps in accordance to IEEE 802.11n standard. The length of the data packets is 1024 bytes, while preamble and data acknowledgements are 16 bytes. There are 100 users in the system ($U = 100$). To adjust the number of active users within each random access shared channel, we vary the probability of preamble transmission p. The average number of active users is pU in this case.

We first consider the successful rate during one random access shared channel with varying bit error rate (BER). Fig. 3 shows the number of preambles detected successfully within one random access shared channel compared to the number of requesting nodes. With no bit error, the receiver is able to decode up to around 20 preambles. Increasing the number of transmitted preambles further will result in performance degradation. This is due to the violations of sparsity requirements. In situation like such, the receiver will acknowledge some preambles that are not sent, and it fails to detect some others that are actually sent. For active users larger

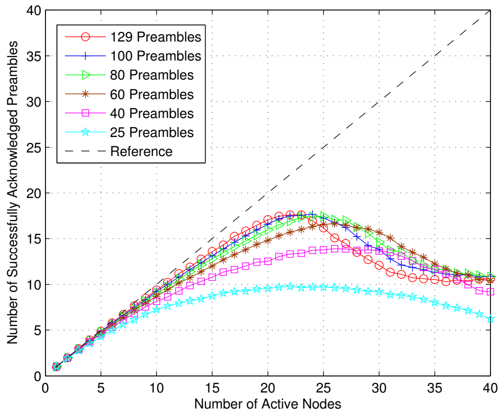

Fig. 4. The number of successfully acknowledged preambles under different number of active users and preambles.

than 30, the behavior of compressive sensing decoder is more random and a percentage of preambles are always detected which results in a steady state operation due to violating sparsity condition. We notice that the number of active users is increased by changing the parameter p.

It is possible that some preambles are reserved for contention-free access instead of contention-based random access. In case of emergency where some nodes have important data to transmit, they will choose reserved preambles which have higher priority. As a result, not all the preambles may be available. In Fig. 4, we vary the number of available preambles and obtain decoding performance for different number of preambles. Results show that reducing preamble numbers will affect the overall performance of the system. When 129 preambles are available, the system can acknowledge 15 preambles out of around 17 transmitted preambles. In the case of 60 preambles, it is 15 out of 20. When fewer preambles are available, collision will happen more frequently. We notice that if two users select the same preamble and transmit with different delays, the decoder can distinguish between them. However, the AP discards that preamble since if the preamble is acknowledged, the data transmission will collide.

Finally, we compare our proposed scheme with CS-MAC [10] and CSMA/CA protocol. Since CS-MAC is a synchronous protocol, we extend the duration of the preamble chips such that the duration is longer than the maximum round trip delay which is 1.66 μs. In this case, there is no need for synchronization. However, the length of preamble would be 14 in this case with the same random access shared channel length. Fig. 5 compares the aggregate throughput at which the AP receives data from all the users. Our scheme has a better performance compared to the other two schemes. For CSMA/CA, the performance increases first. However, due to collision, the performance starts to decrease when the number of active users exceeds 7.

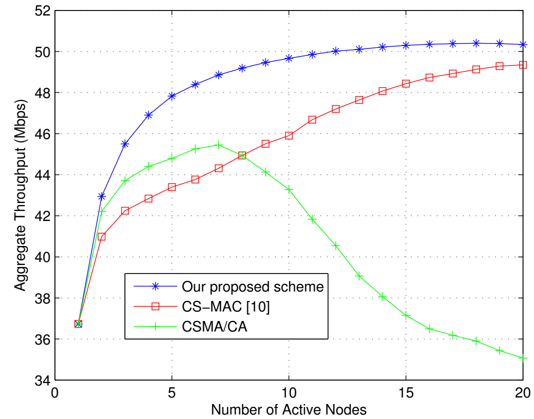

Fig. 5. Performance comparison of the aggregate throughput at which AP receives data packets under different number of active nodes.

IV. Conclusion

In this paper, we proposed a compressive sensing based random access protocol for uplink data transmission. The distinctive feature of the proposed scheme is that it allows for asynchronous transmissions of preambles caused by different propagation delays. Simulation results showed that, compared with the existing solutions, our proposed protocol is more robust against time delays and can achieve a better performance. In our method, the sensing matrix of the identification system admits certain Toeplitz structure. For future work, we will study the restricted isometry property of such structured sensing matrix so as to facilitate analytical performance characterization of the proposed scheme.

References

[1] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, pp. 118–124, July 2007.

[2] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, pp. 21–30, Mar. 2008.

[3] M. A. Davenport and M. B. Wakin, "Analysis of orthogonal matching pursuit using the restricted isometry property," *IEEE Trans. on Information Theory*, vol. 56, pp. 4395–4401, Sept. 2010.

[4] Q. Mo and Y. Shen, "A remark on the restricted isometry property in orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 58, pp. 3654–3656, June 2012.

[5] E. J. Candes, J. K. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, pp. 489–509, Feb. 2006.

[6] ——, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, Mar. 2006.

[7] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proc. of Int'l Conf. on Information Processing in Sensor Networks*, Nashville, TN, Apr. 2006.

[8] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Processing Magazine*, vol. 25, pp. 92–101, Mar. 2008.

[9] F. Fazel, M. Fazel, and M. Stojanovic, "Random access compressed sensing in underwater sensor networks," in *Proc. of Allerton Conf. on Communication, Control, and Computing*, Monticello, IL, Oct. 2010.

[10] T.-H. Lin and H. T. Kung, "Compressive sensing medium access control for wireless LANs," in *Proc. of IEEE Globecom*, Anaheim, CA, Dec. 2012.

[11] A. K. Fletcher, S. Rangan, and V. K. Goyal, "On−off random access channels: A compressed sensing framework," in *Proc. of IEEE Int'l Symposium on Information Theory (ISIT)*, Seoul, Korea, June 2009.

[12] L. Zhang, J. Luo, and D. Guo, "Neighbor discovery for wireless networks via compressed sensing," *Performance Evaluation*, May 2012.

[13] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. on Information Theory*, vol. 13, pp. 619–621, Oct. 1967.