TurfCast: A Service for Controlling Information Dissemination in Wireless Networks

Xinfeng Li, Jin Teng, Boying Zhang, Adam C. Champion and Dong Xuan
Department of Computer Science and Engineering, The Ohio State University
{lixinf, tengj, zhangboy, champion, xuan}@cse.ohio-state.edu

Abstract*—Recent years have witnessed mass proliferation of mobile devices with rich wireless communication capabilities as well as emerging mobile device based information dissemination applications that leverage these capabilities. This paper proposes *TurfCast*, a novel information dissemination service that selectively broadcasts information in particular "turfs," abstract logical spaces in which receivers are situated. Such turfs can be temporal or spatial based on receivers' lingering time or physical areas, respectively. TurfCast has many applications such as electronic proximity advertising and mobile social networking. To enable TurfCast, we propose two supporting technologies: TurfCode and TurfBurst. TurfCode is a nested 0-1 fountain code that enables the broadcaster to transmit either all information or none at all to receivers. TurfBurst exploits the Shannon bound to differentiate among receivers: those who cannot receive information fast enough receive none at all, even if they linger near the broadcaster. We implement TurfCast on real-world devices and conduct experiments in both indoor and outdoor environments. Our experimental results illustrate TurfCast's potential for controlling information dissemination in wireless networks.

I. Introduction

This paper proposes TurfCast, a service that aims to provide fine-grained control over broadcasting in wireless networks. Currently, broadcasting in wireless networks tries to widely disseminate information in a largely homogeneous manner. As we illustrate below, circumstances arise in which dissemination of differentiated content is desirable, which entails precise control over broadcasting.

A. The TurfCast Concept

TurfCast is a novel service that leverages receivers' "turfs" to *selectively* disseminate information. Turfs are abstract logical "spaces" in which certain receivers are situated. Example such turfs can be temporal, based on the amount of time receivers linger, or spatial, based on receivers' varying locations or territories. TurfCast's key idea is to disseminate differentiated information to different turfs via broadcasting. Only "qualified" people, i.e. those in one particular turf, can receive a certain amount of information.

TurfCast's impetus arises from the mass proliferation of mobile devices and their emergent applications. Such devices, especially mobile phones, are highly pervasive in society. Billions of people worldwide use them. Mobile device capabilities have grown far beyond "dumb terminals" to encompass rich wireless network communications capabilities such as Bluetooth, WiFi, and near field communications. Many mobile device based information dissemination applications are emerging that leverage these wireless capabilities. One typical class of such applications is *electronic proximity advertising*. Wireless devices like access points (APs) can broadcast electronic advertisements to customers with mobile devices in "brick-and-mortar" merchants' vicinity [1]–[3]. Naturally, merchants would like to reward loyal customers who purchase goods in their stores with special promotions, particularly during their transactions. For a given merchant, such promotions should be offered only to customers lingering in his or her store, not neighboring merchants' stores. But "traditional" broadcasting may disseminate these promotions to arbitrary passersby outside this merchant's store who do not intend to purchase anything. Thus, a mechanism that prevents such inadvertent broadcasting is needed. Clearly, the store constitutes a spatial turf as well as a temporal turf for its loyal customers. TurfCast can restrict broadcasting to only customers inside such turfs, as Fig. 1 illustrates. Other classes of mobile device applications include *mobile social networking* and *location verification systems*. There are many mobile social networking systems [4]–[6] whose users broadcast electronic personal information to facilitate social interactions. With traditional broadcasting, a user's private information can leak outside his turf, such as the room he currently occupies, to bypassing strangers, whereas TurfCast can restrict broadcasted information within this turf. Current location-based services [7]–[10] require users to verify their location with fine granularity (e.g., within 20 m). With traditional broadcasting, some devices can transmit 100 m, with coarse granularity, whereas TurfCast can provide finer granularity.

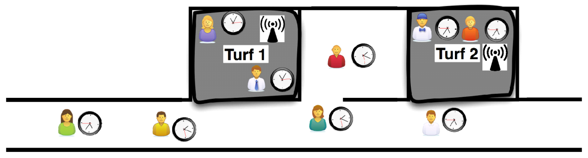


Fig. 1: TurfCast Overview.

In this paper, we consider TurfCast with respect to the time and space domains. Naturally, a receiver who remains longer with a proximate broadcaster can receive more information than a receiver who quickly passes by; also, a receiver who moves closer to the broadcaster can receive more information than a receiver who walks away. This kind of information dissemination is gradual, not "all-or-nothing," and as such, it cannot generate clear turf boundaries for selective information dissemination. Thus we need to design new technologies to support TurfCast.

TurfCast can include approaches such as power control and management and cryptography based approaches like [11], [12]. By reducing transmission power, information dissemination can be restricted to a spatial range. Cryptographic approaches encrypt messages that can only be decrypted by users with the requisite keys. However, these approaches have drawbacks in wireless networks. Power control can only accommodate spatial control of message dissemination, not temporal control. Additionally, it can only accommodate one type of message to be disseminated, whereas in reality, we may have many types of messages. Cryptographic approaches are often computationally expensive and key management is difficult to carry out in a fluid network of mobile receivers. A better approach is needed.

B. Contributions

To enable our TurfCast concept, we propose the following two key supporting technologies:

– *TurfCode:* A new type of nested 0-1 fountain code. We choose fountain codes because wireless channels are typical erasure channels. Fountain codes, as a widely used type of erasure codes, can deliver information to "qualified" receivers more reliably. On the other hand, "disqualified" receivers should not be able to receive any information, even though they may try hard to "peek" at the information in the encoded packets. We find that current fountain codes allow premature decoding to occur. In Section III, we will discuss in detail this kind of premature decoding. In order to foil peeking attempts, improvements over current fountain codes are necessary to help satisfy the requirement of 0-1 communication, i.e., receivers obtain either all information or none at all. As our enhanced fountain code is capable of such communication, we term it a *0-1 fountain code*. We may nest several levels of 0-1 fountain codes to enable a flexible and hierarchical control scheme for information broadcasting.

– *TurfBurst:* We propose using the Shannon bound to differentiate users at different locations. We rely on the reception quality, i.e., the Signal to Noise Ratio (SNR), and the reception rate as distinguishing metrics to select a certain group of people. The broadcaster provides no information to receivers that cannot collect packets fast enough, even if they linger nearby. To achieve this, the broadcaster tries to exceed the receiver's Shannon information bound, i.e., overwhelm the receiver, for a very short period of time. If the receiver's SNR is too low, his information bound will prevent him from receiving enough packets to decode the message.

TurfCast, along with TurfCode and TurfBurst, enable us to control information dissemination in wireless networks. To the best of our knowledge, this work is the first to do so. In TurfCast, the information is disseminated periodically. In one period, the sender transmits the information in bursts and then waits for a certain time. The information is coded using TurfCode. A user needs to stay at a location within the sender's turf and wait for enough periods in order to obtain the desired information. We implement a TurfCast system on real-world mobile devices, including both Linux laptops and Android smartphones, and conduct extensive indoor and outdoor experiments. Our experimental results show that our TurfCast concept and supporting technologies work well on a realistic testbed.

The rest of this paper is organized as follows. Section II discusses related work. Section III details key techniques that enable TurfCast, including TurfCode and TurfBurst. Section IV presents our prototype TurfCast system implementation and evaluation. Finally, Section V presents discussions and concluding remarks.

II. Related Work

In this section, we review the literature closely related to our work. There are several types of information dissemination, such as multicasting [13] and anycasting [14] both in wired and wireless networks. Here, we only focus on wireless broadcasting. We also discuss some background on fountain codes.

– *Wireless broadcasting:* Reliability and overhead of data transmission are two key challenges in wireless broadcasting [15], [16]. The salient features of fountain codes make them a potential solution to address these challenges simultaneously. By adding extra information in encoded data packets, fountain codes enable the information receiver to recover the original data in a lossy channel without retransmitting all packets. Based on the principles of fountain codes, approaches [17], [18] have been proposed to improve the reliability and reduce the transmission overhead of wireless broadcasting. For example, Kumar et al. [17] propose FBcast, a new broadcast protocol that improves sensor networks' wireless broadcast reliability using LT codes. TurfCast differs from these works as it uses and adapts fountain codes in a novel way. Specifically, to the best of our knowledge, TurfCast is the first work that considers using fountain codes to realize controlled information dissemination.

– *Fountain codes:* Fountain codes are sparse graph codes for channels with erasures. They are an important technique to reduce redundancy caused by data retransmission. An idealized digital fountain should have the following properties [19]: (1) A sender can generate a potentially infinite supply of encoding packets from the original data. Encoding packets should be generated in constant time per packet given the original data; and (2) A receiver can reconstruct a message that would require k packets to send once any k encoding packets have been received. The reconstruction time should preferably be linear in k. Approximations to a digital fountain can be obtained from the idealized version by loosening the requirements in various ways. Four representative codes are: (1) Reed-Solomon codes [20]; (2) Tornado codes [21]; (3) Luby Transformation (LT) codes [22]; and (4) Raptor codes [23]. Specifically, LT codes and Raptor codes are rateless codes, which means no fixed rate need be determined in advance, and an infinite stream of encoding packets can be constructed from the message data.

III. TurfCast Design

TurfCast has two key technologies, TurfCode and TurfBurst. In this section, we discuss their details as well as using them to realize controlled information dissemination.

A. TurfCode

TurfCode's origin lies in fountain codes, but greatly differs from them. Our key observation is that conventional fountain codes allow for partial information recovery, or more formally, *premature decoding* if the number of blocks to break down a message is small. Thus it is difficult to realize 0-1 communication. In this light, we design TurfCode, a new type of fountain code suitable for such communication. TurfCode can also take the form of a nested 0-1 fountain code to enable more flexible information control.

1) Premature Decoding of Fountain Codes: Fountain codes are based on linear combinations of input blocks. Such codes can generate a finite or infinite number of linearly combined input blocks, typically using bit-wise XOR operation. We can view the encoding process as a vector operation. If $\mathbf{c} = (c_1, c_2, \cdots, c_n)^T$ is the vector of encoded blocks and $\mathbf{m} = (m_1, m_2, \cdots, m_k)^T$ is the vector of original message blocks, then $c_i = r_i^B \mathbf{m}$, where r_i^B is the abstraction of the linear operations performed. Let $B = (r_1^B, \cdots, r_n^B)^T$, we can then write $\mathbf{c} = B\mathbf{m}$. Correspondingly, the decoding process can be described as $\mathbf{m} = B^{-1}\mathbf{c}$ if we have k orthogonal encoded blocks in $\mathbf{c}$ and the rank of B equals k.

One observes that we can decode a fraction of the information, even if the number of received encoded blocks has not reached the threshold for correct decoding. We term this *premature decoding* if one can explicitly discover any m_i from the received c_is or their combinations. More precisely, we require that all c_is and their linear combinations be the result of XOR operations involving at least two m_is. Mathematically, this is equivalent to the following: no row or combination of rows in B can be an e_i, where e_i is a $1 \times k$ vector whose i-th element is 1; all others are 0. We also assume that the receiver is intent on getting the information, so he will try his best to peek into the encoded blocks. That means he may linearly combine the blocks to see whether he can recover any original message.

Theorem 1: A fountain code based on bitwise XOR operations cannot prevent receivers from decoding any of the original message blocks before they receive the whole k orthogonally encoded blocks.

Proof: Suppose that a receiver has received $k-1$ orthogonally encoded blocks. Thus he has a $(k-1) \times k$ matrix B' with rank $k-1$. Without loss of generality, we assume that the last column is not clean after Gaussian elimination, as is shown in Eq. 1.

$$B' = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 & b_{1k} \\ 0 & 1 & \cdots & 0 & 0 & b_{2k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & b_{(k-2)k} \\ 0 & 0 & \cdots & 0 & 1 & b_{(k-1)k} \end{pmatrix}_{(k-1)\cdot k} \quad (1)$$

In order to satisfy the irretrievability described above, the elements at the end of each row, i.e., b_{ik}, must be 1. Thus $r_i^B (i \in \{1, 2, \cdots, k-1\})$ has even number of "1".

Then the last encoded block c_k arrives, making the whole matrix decodable. The encoding row r_k^B must be a combination of $e_k = (0, 0, 0, \ldots, 0, 1)$ and some other rows in B'. Note that r_k^B has odd number of "1". Suppose a set of rows in B, the collection of which is denoted as R, are involved in the Gaussian elimination for these rows in B'. If R and r_k^B are received, then we can add all encoded blocks to determine e_k. This can occur because we have an erasure channel in which erasures and out-of-order packet deliveries occur.

Thus, if $|R| < k-1$, we have already gotten e_k and the original message m_k before we get the last k-th encoded block. If $|R| = k-1$, let we consider the set of $r_i^B (i \in \{2, 3, \cdots, k\})$. Because these rows are linearly independent, we can get a similar matrix B'' as B'. At least one row of B'' is computed from r_k^B, which must contain odd number of "1". So this row becomes e_i in B''. One original message has been decoded with $k-1$ orthogonally encoded blocks. $\square$

Fig. 2 illustrates the blocks prematurely decoded when the receiver gets more and more orthogonal encoded blocks, which are based on the average results of 100 runs of LT codes. As we see from this figure, premature decoding is not a significant issue when $k \geq 128$, but it becomes much worse when k decreases. Small ks are favored by short and bursty messages, which are commonly used in the TurfCast applications mentioned in Section I.

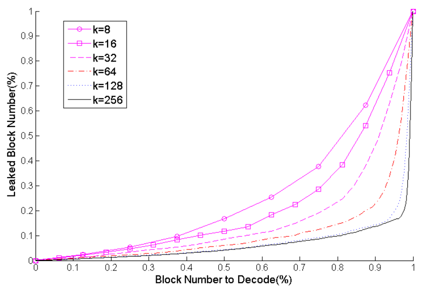


Fig. 2: Premature decoding when the LT parameters $c = 0.2$ and $\delta = 0.5$. Parameter details are expounded in [22].

In this light, a new mechanism should be designed to guarantee zero premature decoding, especially for small ks. This new mechanism's meaning follows from broadcasted information in real-world mobile environments, which is usually short, fast and dynamic. For example, a mobile advertising system might broadcast many different advertisements in a short period. Note that the new mechanism should work smoothly when k is changed from small to large. In this way, the new mechanism can properly realize 0-1 communication. An intuitive solution is to use encryption. We can feed $\mathbf{m}' = E_{key}(\mathbf{m})$ into the fountain encoder and concatenate a piece of the key to each m_i'. In this way, the receiver cannot decipher all the m_is until the whole $\mathbf{m}'$ is received.

2) 0-1 Fountain Code: In order to enable this kind of 0-1 communication without full-fledged encryption, we propose to add a pre-coding stage prior to the actual decoding stage as shown in Fig. 3. We scramble the input blocks and feed the scrambled data into the fountain encoder. We attach a fraction of the scrambling rules to each encoded block so that the information cannot be retrieved before all the scrambling rules are known. We want the scrambling to help prevent the receiver from examining the content before it is "ready," i.e., when insufficient blocks have been received.

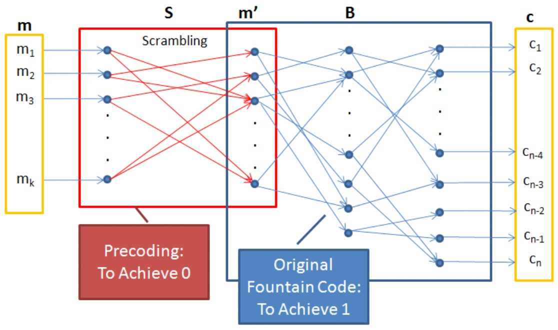


Fig. 3: 0-1 Fountain Code.

We propose tridiagonal scrambling to achieve 0-1 communication. We multiply the input block vector $\mathbf{m}$ with a tridiagonal scrambling matrix S, making the input to the actual fountain encoding $\mathbf{m}' = S\mathbf{m}$. The operations here are not bit-wise XOR, but normal integer additions and multiplications. The value of each symbol in m_i is treated as an integer. Because the number of bits for a symbol is limited, there is a limit on how large the integer can be. As we may conduct multiplications and additions involving several symbols, and put the result in a symbol of the same length, the original symbols cannot be assigned the possible maximum value, because of overflow. The "wasted" bits go into overhead.

We have the tridiagonal matrix as Eq. 3. The elements on the diagonal, i.e., s_{ii}, can be random integers in a certain interval that is related to how much communication overhead we have. If we plan to "waste" 3 bits, then the s_{ii} can be chosen from the interval $[2, 6]$ (note 1 is excluded to guarantee 0-1 communication), because there are possibly another two 1s in the row, making the maximum output 8 times the original maximum value. In our implementation, each symbol has 8 bits. So 3 bits is a slight overhead if we combine more symbols together for scrambling, i.e., the block length, m_i, is not too small. On the other hand, we must ensure that the generated S is a full rank matrix. From [24], we can have a general form of each element in S^{-1}. In the equations of that form, there is a θ_i for each row. If every θ_i is non-zero, then the matrix S has full rank, since its inverse matrix exists. θ_i can be easily computed by a recursion:

$$\begin{aligned} \theta_i &= s_{ii}\theta_{i-1} - \theta_{i-2}, n \geq 3 \\ \theta_1 &= s_{11}, \theta_2 = s_{11}s_{22} - 1 \end{aligned} \tag{2}$$

Since $s_{ii} \geq 2$, we have $\theta_2 - \theta_1 \geq s_{11} - 1 > 0$. Similarly, $\theta_i - \theta_{i-1} \geq \theta_{i-1} - \theta_{i-2}$. By mathematical induction, we know that θ_i is monotonically increasing with i and is never zero. Thus S is a full rank matrix.

$$\begin{pmatrix} s_{11} & 1 & 0 & \cdots & 0 & 0 \\ 1 & s_{22} & 1 & \cdots & 0 & 0 \\ 0 & 1 & s_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & s_{(k-1)(k-1)} & 1 \\ 0 & 0 & 0 & \cdots & 1 & s_{kk} \end{pmatrix} \tag{3}$$

We feed the scrambling output plus the scrambling matrix S into the fountain encoder, i.e., we feed $\mathbf{y} = \langle S\mathbf{m}, S\rangle$, where $\langle\cdot,\cdot\rangle$ is the concatenation operation. We embed S in $\mathbf{m}'$ to let the receiver know. In practice, the length of a block can be large, e.g., 1 KB or more. On the other hand, one column of S only requires k bits, so the overhead is not large. Furthermore, the computation of S^{-1} only requires $O(n^2)$ time [24], which is much less than the time complexity of decoding a general matrix B, approximately $O(n^3)$. In Theorem 2, we prove there is no explicit premature decoding with scrambling.

Theorem 2: Let S' be a matrix containing some rows of S, where S is the $k \times k$ scrambling matrix with rank k. Then, no linear combinations of the rows in S' can yield an e_i unless S' is of rank k.

Proof: Suppose the receiver has received $n < k$ encoded blocks. It then has an n by k matrix S'. We need to show there is no linear combination of rows in S' that can yield an e_i.

1) Neither the first row (r_1^S) nor the last row (r_k^S) is involved in the combination. The left-most "1" and the right-most "1" can never be eliminated because both these two columns both have only one non-zero element for combination.

2) One of r_1^S and r_k^S is used for combination. Let us consider the case of r_1^S first. r_2^S must be combined with r_1^S. Otherwise, s_{11} and the right-most "1" of some row can not be eliminated, resulting in at least two non-zero elements in the final combination result. S' can be arranged into an upper triangular matrix, as is shown in Eq. 4 (only the upper-left part). Because $s_{ii} > 1$, each row has two non-zero elements and every row "mismatches" at least one column with others. Hence, the left-most and the right-most columns have non-zero elements after linear combination. The proof for the case of r_k^S is similar (consider the lower triangular matrix in Eq. 4).

3) Both r_1^S and r_k^S are used for combination. Then S' can be arranged as Eq. 4. Since there is at least one row missed in S', the upper triangular matrix starting from r_1^S and the lower triangular matrix starting from r_k^S mismatch at least one column. So there are no any two rows in S' having the non-zero elements in the same columns. Hence, there is no way to get e_i based on the rows in S'. □

$$\begin{pmatrix} s_{11} & 1 & 0 & \cdots & 0 & 0 \\ 0 & s_{22} - \frac{1}{s_{11}} & 1 & \cdots & 0 & 0 \\ 0 & 0 & s_{33} - \frac{1}{s_{22} - \frac{1}{s_{11}}} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & s_{(k-1)(k-1)} - \frac{1}{s_{kk}} & 0 \\ 0 & 0 & \cdots & 0 & 1 & s_{kk} \end{pmatrix}_{n\cdot k} \tag{4}$$

Example: We will go through the whole process with the following example. In this example, we have $k = 4$ and the text message “Hello,world!”. If the block length is 3, we get the original input:

$$m = \begin{pmatrix} 72('H') & 101('e') & 108('l') \\ 108('l') & 111('o') & 44(',') \\ 119('w') & 111('o') & 114('r') \\ 108('l') & 100('d') & 33('!') \end{pmatrix}$$

Suppose the generated B is an identity matrix I. (Though unlikely, this is possible.) Without a 0-1 fountain code, the receiver will receive exactly $I \cdot \mathbf{m} = \mathbf{m}$ and all information is prematurely decoded.

Now with the scrambling matrix S between B and $\mathbf{m}$, we may have $I \cdot (S \circ \mathbf{m}) = \mathbf{c}$, as is shown below. As the operations in the two multiplications are different, we use different symbols.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \left(\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \circ \begin{pmatrix} 72 & 101 & 108 \\ 108 & 111 & 44 \\ 119 & 111 & 114 \\ 108 & 100 & 33 \end{pmatrix} \right)$$
$$= \begin{pmatrix} 0 & 253 & 58 & 4 \\ 1 & 152 & 179 & 54 \\ 1 & 199 & 178 & 49 \\ 1 & 80 & 55 & 180 \end{pmatrix}$$

It is easy to see that any linear combinations of one to three rows in c cannot yield any element in m, as no such rows in S can yield an e_i.

3) Multiple-step 0-1 Fountain Code: A single 0-1 fountain code can achieve a step function of information acquired vs. information transmitted (Fig. 4(a)). There might be different levels of control over information dissemination, as is shown by the contour of L1, L2 and L3 in Fig. 4(b). If $u(x)$ is the standard unit step function, the function in Fig. 4(a) can be written as $a_1 u(x - x_1)$, and the one in Fig. 4(b) can be written as $\sum_{i=1}^{l} a_i \cdot u(x - x_i)$, where l is the step number. A naïve way to achieve these different levels is to use separate fountain coding schemes. Tighter control means more blocks to decode, and vice versa. However, this scheme may not work in the cases where we do not want the receiver to have higher level information until all lower level information is received. This requirement is natural in our TurfCast system, as we want information acquisition to increase with turf priority.

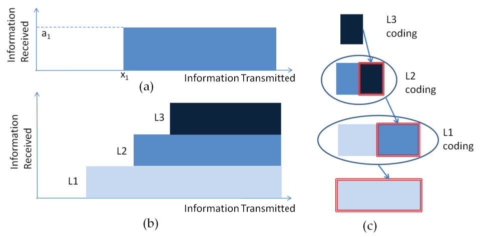


Fig. 4: Single step function, multiple step function and the nested coding scheme.

In this light, we propose a nested coding solution. Each bar of information in Fig. 4(b) is encoded with its own 0-1 fountain code. The coding of darker levels is nested in the coding of lighter levels, as is shown in Fig. 4(c). In Fig. 4(c), each square is a block and a square with double lines is an encoded one. After an L3 block is encoded, it is combined with an original L2 block, and then the aggregate block is encoded with an L2 fountain code. Similarly, L1 coding will encode the original L1 block and the encoded aggregate of L2 and L3 blocks. For example, if an L2 message requires 64 blocks and an L1 message requires 32 blocks to decode, then we can put one L2 encoded block in one L1 original block for coding. After receiving two L1 messages, one L2 message can be decoded. Sometimes we need non-integer ratio nesting to achieve a particular information reception ratio, e.g., 4:3 in Fig. 5. Any three encoded L2 blocks are split into two parts with 1.5 blocks in each part. One original L1 block plus 1.5 encoded L2 blocks are encoded together. Finally, after receiving four L1 messages (totally $6 \times 32 = 194$ blocks), three L2 messages can be decoded ($194/64 = 3$).

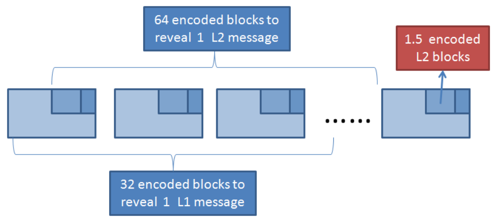

64 encoded blocks to reveal 1 L2 message
1.5 encoded L2 blocks
......
32 encoded blocks to reveal 1 L1 message

Fig. 5: Nesting in multiple-step 0-1 fountain code.

B. TurfBurst

TurfBurst is the another key supporting technology of TurfCast. TurfBurst sends packets in a short and bursty way such that the receiver's reception quality must suffice to handle the bursts.

1) Distance, SNR and Reception Rate: Typically, for wireless channels, the farther a recipient is from a broadcaster, the poorer the recipient's communication quality. This is a direct result of attenuating Signal to Noise Ratio (SNR). SNR can be defined as the ratio of signal power over noise power. The signal power is the expected value of (squared) Received Signal Strength Indication (RSSI). Statistically, there is an underlying mathematical equation [25], which can be written in the form of Eq. 5.

$$RSSI = P_0 - \alpha \cdot 10 \log(d/d_0) + N, \tag{5}$$

where P_0 is the transmission power measured at a reference distance d_0 , α is the attenuation coefficient, d is the distance, and N is the noise power, usually modeled as an additive white Gaussian process. As SNR is closely related to $RSSI$ and N, SNR can be expressed as a function of d. Also, SNR plays a central role in determining the maximum information transmission rate. For a specific device, it is not difficult to get the information rate as a function of SNR. Combined with the SNR vs. d curve, we can get the relation between this rate and d.

2) Information Dissemination in Bursty Traffic: With reception quality control, we want users whose reception qualities differ to receive information of different priorities. Higher priority messages should reach receivers with poorer reception quality who may be farther away from the sender. Our solution uses the following insight: the worse signal quality a receiver has, the less information he can get. We can modify the sending duration to change the rate of information transmission in order to saturate the wireless channel for a very short period of time. If the receiver's error rate is very high, he will not be able to get enough blocks for decoding.

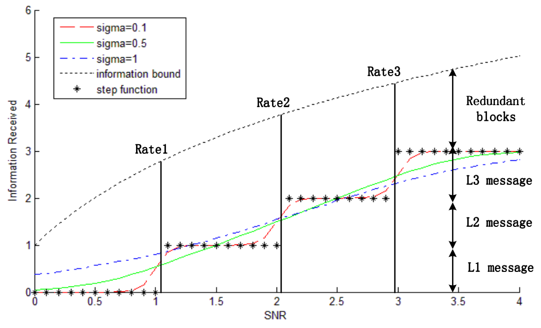


Fig. 6: Schematic diagram of TurfBurst. The whole blocks received at Rate 1, 2, 3 can be used to decode the L1, L1+L2, L1+L2+L3 messages respectively. The information sending rate should exceed Rate 3.

There is a major difference between temporal control and spatial control, i.e., the latter has step smoothing due to probabilistic reception of wireless packets, as shown by the lower curves in Fig. 6. In essence, the information volume at a certain SNR is an average value over a long period of time. In a short period of time, the receiver may receive an disproportionate amount of information regarding his SNR. Thus the sharp boundaries of the step function $e(x)$ ($e(x) = \sum_{i=1}^{l} a_i \cdot u(x - x_i)$) are thus softened to become $e'(x) = E[e(x + N)]$. The noise N, which can be thought of as an additive Gaussian random process $Normal(\mu, \sigma)$, plays an important role in determining $e'(x)$. When σ is large, which generally indicates small SNR and large d , the smoothing effect is more evident, as is shown by the different lines in Fig. 6. This implies that with larger noise, we need to make each step higher and longer, i.e., make $a_k - a_{k-1}$ and $x_k - x_{k-1}$ large enough to counteract the large σ value.

3) TurfBurst with Barriers: We note that the smoothing effect is actually based on the assumption of a strong relation between information rate and distance (Eq. 5). In other words, the noise N significantly affects the step function by gradually attenuating signal strength. However, physical barriers would cause dramatic attenuation within a very short distance, e.g., walls in buildings. Several previous studies [26], [27] have considered the influence of indoor environment on signal attenuation and demonstrated the feasibility of indoor localization based on distinguishable RSSIs. For example, when passing through a wall, a wireless signal attenuates by 3.1 dB on average [26]. In TurfBurst, a receiver outside one room could not receive enough packets to decode a message sent from an AP in the room, while another receiver inside could get it, even though they are separated by a wall. Physical turfs, such as rooms and galleries, will significantly weaken the smoothing effect and sharpen TurfBurst's step function. In Section IV, we demonstrate the strength of the "turf effect" in a building.

C. Realizing Controlled Information Dissemination

We envision that there are messages of several different priorities to be sent. Each priority level has distinctive requirement, i.e., when the message should be received and how far the message should reach. Also messages with lower priority cannot be received before those with higher priority. For clarity of explanation, we illustrate the temporal and spatial control mechanism using two priority levels. We term the level with higher priority L1 and the other L2.

The basic control scheme is shown in Fig. 7. TurfCast has two parts. The first one is used for temporal control, which is the upper part of Fig. 7. There is a T_i, which is used to time the sending of encoded blocks. Only after enough T_i periods have elapsed can the receiver decode the message. The second one is used for spatial control. Within one T_i, the sender broadcasts a certain amount of encoded blocks, which spans T_s time, and wait for the next period. During the T_s time, the sender sends out k^s encoded blocks, which are all L1 encoded blocks with nested L2 encoded blocks, as is shown in the bottom of Fig. 7. Then $T_s = T_t \cdot k^s + \Delta t \cdot (k^s - 1)$, where T_t is the time to transmit a block via a wireless card and Δt is the interval between two blocks. We carefully design the parameters T_i and Δt to control the information volume for both L1 and L2 sent in a given period of time. Below, we discuss how temporal and spatial controls are implemented.

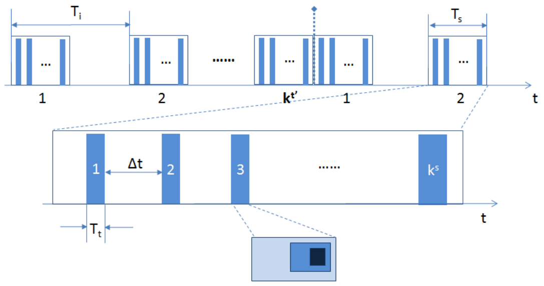


Fig. 7: Timing for TurfCast.

Temporal Control: If we want a message to be delivered at time point t_0, then we split the messages into $k = k^t k^s$ blocks, encode these blocks and send them. As fountain codes normally incur an overhead, receivers need $k' = (1 + \epsilon)k$ ($\epsilon > 0$) encoded blocks, which is the expected number for a message to be decodable [22]. So the sender can calculate the time, and send the last one of the $(1 + \epsilon)k$ blocks exactly at t_0. We control the transmission interval of encoded blocks, which is denoted by T_i in Fig. 7. If we set $T_i = t_0/(k^{t'} - 1)$ where $k^{t'} = (1 + \epsilon)k^t$, we can ensure that the receiver cannot decode the whole message before t_0.

Since L1 messages need to be received before L2 messages, message decoding needs to be incremental. In this context, we use the nested codes in TurfCode. Higher priority messages are encoded in the outer layer, containing encoded blocks of lower priority messages. Then only after the outer level messages are decoded, can a fraction of the inner message blocks to be seen. This ensures the prioritized delivery. To quantify the delivery time, suppose the L1 message needs $(1+\epsilon_1)k_1$ blocks to decode at time t_1, and the L2 message needs $(1+\epsilon_2)k_2$ blocks to decode. Each L1 block contain n encoded L2 blocks. Then the decoding time for the L2 message is $t_2 = t_1 \cdot \frac{(1+\epsilon_2)k_2}{n(1+\epsilon_1)k_1}$.

Spatial Control: For spatial control, we also need to time block transmission. We control Δt with a block of time T_s to send blocks, which are small dark marks in the boxes, fast enough. If the expected number of blocks to decode is k^s, then $\Delta t = (T_s - T_t k^s)/(k^s - 1)$. With appropriate Δt, we may reach the maximum rate a receiver can receive. Then the exceeding part is all lost. Note that the information sending rate should be fixed for one particular TurfCast setting. Hence, Δt can control the outermost layer with a specified SNR and rate, as is shown in Fig. 6. The spatial control of inner layers is related to the nesting ratio n and the fountain code parameters, k and ϵ. To fill the gap between the bound and the step function, we may need to duplicate the original messages or add random information. We need to change our coding parameters in either S or B for different periods. Otherwise, people can compensate for slow motion by lingering longer. In order to foil these attempts, we should change to a different set of coding parameters at regular interval. As is shown in Fig. 7, we change the parameters after $k^{t'}$ periods. Note that T_i is not necessary between two bunches of blocks with different coding parameters because these blocks cannot be gathered together for decoding.

Double Controls: We discuss how to achieve information control via temporal and spatial control. Suppose that the L1 message is of size M_1, the L2 message is of size M_2, and the basic block length for L1 and L2 messages are l_1 and l_2. L1 message allows the receiver to be far away and has a rate r_1 as an information bound, while L2 message can only be received by users who has r_2 ($r_2 > r_1$) or faster speed. We further require the user to be nearby for at least time t_1 to receive the L1 message and time t_2 to receive the L2 message. k_i^s ($i = 1, 2$) and k_i^t ($i = 1, 2$) are defined as before. Each encoded L1 message block contains n encoded L2 message block. Then we have the following equations:

$$\begin{array}{rclrcl} k_2^s & = & n \cdot k_1^s & T_i & = & t_1/(k_1^{t'} - 1) \\ T_s & = & (l_1 + nl_2) \cdot (k_1^s)/r_1 & M_j & = & k_j^t k_j^s l_j \ (j = 1, 2) \\ t_1/t_2 & = & k_1^t/k_2^{t'} = r_1/r_2 & \Delta t & = & T_s/(k_1^s - 1) \end{array} \tag{6}$$

The last two equations come from:

$$\begin{array}{rl} t_1/t_2 & = nk_1/(1+\epsilon_2)k_2 = nk_1^t k_1^s/(1+\epsilon_2)k_2^t k_2^s = k_1^t/k_2^{t'} \\ \Delta t & = (T_s - T_t k_1^s)/(k_1^s - 1) \approx T_s/(k_1^s - 1), \end{array}$$

where T_t is the time to physically transmit one block, which is normally negligible. In reality, we can fix k_1^t, k_1^s and n, and then all other parameters.

IV. Implementation and Evaluation

A. Implementation

We implement a TurfCast system on real-world mobile devices, including Linux laptops and Android smartphones. Our system is implemented in both the MAC and network layers. At the MAC layer, we rewrite `aireplay-ng`, one of the `aircrack` tools, to inject MAC layer packets on one laptop that acts as a broadcast server, and we use the `tcpdump` and `libpcap` libraries on a second laptop to sniff the encoded packets on the client side. Our MAC layer codes are written in C. In order to sniff MAC layer packets, the client laptop's wireless card (802.11 b/g) needs to run in monitor mode on the same channel as the server. At the network layer, to avoid TCP's unnecessary packet retransmissions, we use UDP to transmit our TurfCast messages. Our network layer implementation runs on laptops and Nexus S smartphones running Android 2.3.3; both the phones and laptops connect to a single access point. Our network layer code is written in Java.

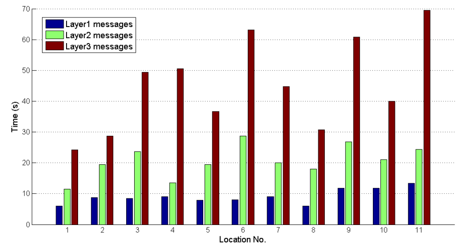
B. Evaluations

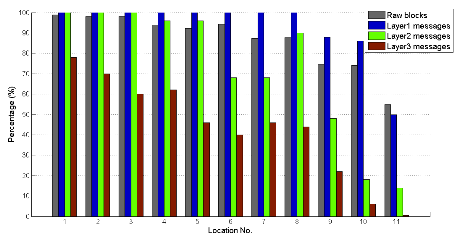
We conduct indoor and outdoor experiments to validate our TurfCast design. Smartphone-based experiments are designed to test the impact of mobility and demonstrate the feasibility of deploying TurfCast on current commodity smartphones. Our evaluations use two metrics: (1) Decoding time variation for multiple layer encoding (temporal control); and (2) The variation of the decoding probability of receiving information at different locations (spatial control).

1) Parameter Setting: We set three layer messages, L1, L2 and L3. $k_1 = 16$ and $k_2 = k_3 = 32$. L1, L2, and L3 block lengths are 128, 256, and 512 bytes, respectively. Each encoded L2 block contains one encoded L3 block, and each encoded L1 block contains two encoded L2 blocks. The parameters for LT codes are $c = 0.2$, $\delta = 0.5$ and $\epsilon_1 = 0.61$, $\epsilon_2 = \epsilon_3 = 0.64$. For simplicity, we consider $2 \times k_i$ blocks are expected to decode a message. We set $k_1^s = 16$, $t_1 = 5s$, $\Delta t = 5ms$. According to Eq. 6, we have $k_2^s = k_3^s = 32$, $k_1^t = k_2^t = k_3^t = 1$ and $T_i = t_1 = 5s$, $t_2 = 2 \cdot t_1 = 10s$, $t_3 = 2 \cdot t_2 = 20s$. Because reception of inner messages rely on successful decoding of outer messages, we increase $k_2^{t'}$ from 2 to 3 in order to improve the reception probability of L3 messages. When measuring the decoding probability, T_i is set as same as T_s, i.e., there is no sleep time between blocks except Δt, and 50 nested L1+L2+L3 messages are broadcast continuously.

2) Indoor Experiment: The indoor experiments are conducted in a building on a university campus. We set up a broadcast server in one room and test the information reception performance at 11 different locations on the same floor, as is shown in Fig. 9. The temporal and spatial performance is presented in Fig. 8. We set the timeout threshold to 120 s. Every time datum is averaged over 10 runs and the timeout data are excluded.

From Fig. 8a, we observe that when devices are near the server, the three layer messages can be decoded at about 6 s,


(a) Indoors: Temporal control at different locations


(b) Indoors: Spatial control

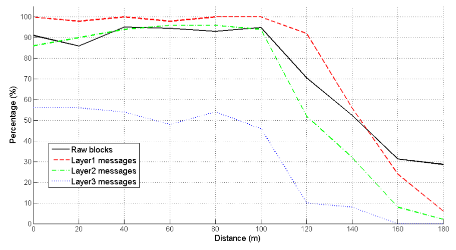

(c) Outdoors: Spatial control

Fig. 8: Indoor and outdoor performance of 3-layer TurfCast messages.

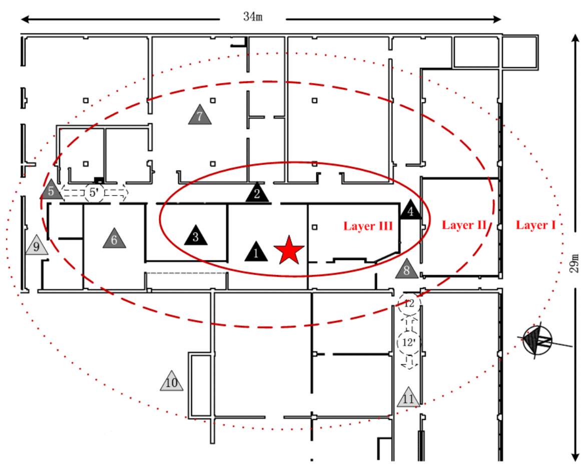


Fig. 9: Indoor map. The broadcasting server is at the star; client laptops and smartphones are at the triangles and circles. The black levels of triangles reflect three different levels of message reception probabilities: 1) L1, L2, L3 $> 50\%$; 2) L1, L2 $> 50\%$ and L3 $< 50\%$; 3) L1 $> 50\%$ and L2, L3 $< 50\%$.

11 s, and 21 s. The small deviation of 1 s is due to the last set of blocks' transmission time and processing delay. When far away from the server, especially separated by several walls, the time to decode a message increases significantly due to severe packet loss, particularly for L2 and L3 messages. L3 messages show the greatest variation as locations change.

Based on the decoding probability results in Figs. 8b and 9, we see that the influence of physical barriers is noticeable in information reception. Two walls or two rooms suffice to distinguish two different messages, e.g., Location 1 vs. 6. A receiver has 78% probability to decode an L3 message at Location 1 but only 40% probability at Location 5. Furthermore, we find that corners also play an important role at determining the information reception results, for example, Locations 5 vs. 9. A receiver has 96% probability to decode L2 at Location 5 but only 40% at Location 9, only one corner apart.

Smartphone Experiment: We select 6 locations in our indoor testbed, two of which are mobile (see Fig. 9). Two of the authors wandered at Locations 5' and 12' at 1 m/s, holding two Nexus S phones. Four other Nexus S phones are put statically at Locations 1, 5, 8, and 12. The broadcast server continuously transmits 30 nested 3-layer messages and each smartphone records the time to receive each message.

Comparing the results of Locations 8 and 12 in Fig. 10, we can clearly observe barriers' impact in sharpening the step function, e.g., the door between these locations. The message reception probabilities at different layers are significantly distinguished behind one door. The probability of L3 message reception decreases from 80% to 13%, while that of L1 only decreases by 13%. We show the impact of mobility with Locations 5 vs. 5' and Locations 12 vs. 12' in Fig. 10. Generally, mobility decreases the message reception probabilities regardless of the movement direction. For temporal control, mobility evens the distribution of received messages over time, although the average performance does not change much.

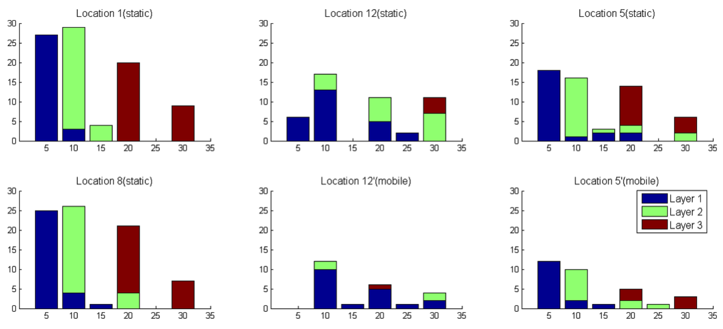
3) Outdoor Experiment: The results of our outdoor experiments are shown in Fig. 8c. The information loss starts at 100 m. At 120 m, the probability of decoding L1 is 92%, while L2 is 52% and L3 is only 10%. We notice that outdoor information loss is "smoother" than indoor loss. We find that there are more fluctuations, e.g., at 80 m, which are likely due to environment changes and passersby. This demonstrates that noise greatly affects TurfCast's step feature outdoors.

V. Discussion and Final Remarks

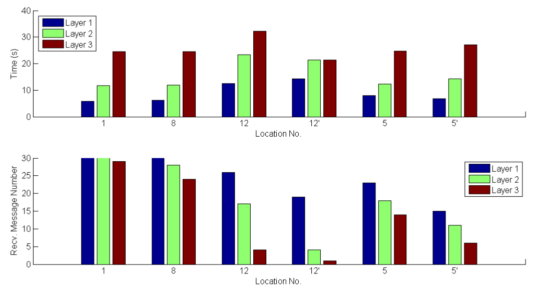
There are some issues that merit attention.

– *Accuracy of spatial control.* Our spatial control is based on SNR. In practice, factors such as interference and environmental changes impact SNR. They make it very difficult to achieve complete spatial control of information dissemination. However, with the help of physical barriers, we can decrease the influence of unpredictable noise and sharpen the spatial turf. Further, our nested 0-1 coding technology relaxes the communication rate's sensitivity to SNR, which makes TurfCast more robust in comparison to existing information dissemination schemes.

– *Security and privacy.* In this paper, we introduce the concept of controlled information dissemination. If applications can generate significant revenue, they may attract malicious users who abuse them to obtain extra benefits. For instance, in a mobile location verification application, malicious users may copy information from their peers to falsely "verify" their locations. To help defend against this attack, receivers can randomly respond to the server as "checkpoints". Providing strong security and privacy protection against attacks on TurfCast forms an important part of our future work.



(a) Distribution of each received message over time. x-axis is the elapsed time (s); y-axis is the number of received messages.



(b) Average performance of message reception.

Fig. 10: Smartphone Performance of 3-layer TurfCast Messages.

This paper presented TurfCast, a novel service for controlled information dissemination. TurfCast leveraged TurfCode, a nested 0-1 fountain code that disseminates either all information or none at all to receivers in the time domain. TurfCast also leveraged TurfBurst, which exploits the Shannon bound to differentiate users based on their distance from the broadcaster in the space domain. Fountain code premature decoding was introduced and analyzed. We designed and implemented TurfCast on real-world devices. Our experimental evaluation showed the promise of our concepts and designs for controlled information dissemination.

Acknowledgment

This work was supported in part by China 973 Project 2011CB302800, the National Science Foundation of China (NSFC) under grant No. 61070221, and the US National Science Foundation (NSF) under Grant No. CNS0916584, CNS1065136, Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] A. Garyfalos and K. C. Almeroth, "Coupons: A multilevel incentive scheme for information dissemination in mobile networks," *IEEE Trans. on Mobile Computing*, vol. 7, no. 6, pp. 792–804, Jun. 2008.

[2] L. Aalto, N. Gohlin, J. Korhonen, and T. Ojala, "Bluetooth and WAP Push-Based Location-Aware Mobile Advertising," in *MobiSys*. New York, USA: ACM Press, 2004, pp. 49–58.

[3] S. Kurkovsky and K. Harihar, "Using Ubiquitous Computing in Interactive Mobile Marketing," *Personal and Ubiquitous Computing*, vol. 10, pp. 227–240, 2006.

[4] J. Teng, B. Zhang, X. Li, X. Bai, and D. Xuan, "E-Shadow: Lubricating Social Interaction using Mobile Phones," in *Proc. of IEEE Int'l. Conf. on Distributed Computing Systems (ICDCS)*, 2011.

[5] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Shmidt, "Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation," in *MobiSys*, Jun. 2008.

[6] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application," in *SenSys*, 2008.

[7] Z. Zhu and G. Cao, "APPLAUS: A Privacy-Preserving Location Proof Updating System for Location-Based Services," in *INFOCOM*, 2011.

[8] J. Manweiler, R. Scudellari, and L. P. Cox, "SMILE: Encounter-Based Trust for Mobile Social Services," in *Proc. ACM CCS*, 2009.

[9] S. Saroiu and A. Wolman, "Enabling New Mobile Applications with Location Proofs," in *HotMobile*, 2009.

[10] W. Luo and U. Hengartner, "Proving Your Location Without Giving Up Your Privacy," in *HotMobile*, 2010.

[11] S. Berkovits, "How to broadcast a secret," *Advances in Cryptology, EUROCRYPT'91*, vol. 547, pp. 535–541, 1991.

[12] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 339–348, May 1978.

[13] W. Jia, W. Zhao, D. Xuan, and G. Xu, "An efficient fault-tolerant multicast routing protocol with core-based tree techniques," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, pp. 984–1000, October 1999.

[14] D. Xuan and W. Jia, "Distributed admission control for anycast flows with qos requirements," in *Proc. of IEEE Int'l. Conf. on Distributed Computing Systems (ICDCS)*, April 2001.

[15] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *MobiHoc*, 2002, pp. 194–205.

[16] F. Wang, M. T. Thai, Y. Li, X. Cheng, and D.-Z. Du, "Fault-tolerant topology control for all-to-one and one-to-all communication in wireless networks," *IEEE Transaction on Mobile Computing*, vol. 7, no. 3, pp. 322–331, March 2008.

[17] A. Kumar, A. Paul, U. Ramachandran, and D. Kotz, "On Improving Wireless Broadcast Reliability of Sensor Networks Using Erasure Codes," in *MSN*, 2006, pp. 155–170.

[18] N. Dutsch, H. Jenkac, T. Mayer, and J. Hagenauer, "Joint Source-Channel-Fountain Coding for Asynchronous Broadcast," in *Proc. IST Mobile & Wireless Communications Summit*, 2005.

[19] M. Mitzenmacher, "Digital Fountains: A Survey and Look Forward," in *Information Theory Workshop*, 2004.

[20] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of SIAM*, vol. 8, pp. 300–304, 1960.

[21] M. Luby, M. Mitzenmacher, A. Shokrollahi, , and D. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[22] M. Luby, "LT codes," in *FOCS*, 2002, pp. 271–282.

[23] A. Shokrollahi, "Raptor Codes," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[24] R. A. Usmani, "Inversion of a tridiagonal Jacobi matrix," *Linear Algebra Appl.*, vol. 212–213, pp. 413–414, 1994.

[25] N. Banerjee, S. Agarwal, P. Bahl, R. Chandra, A. Wolman, and M. Corner, "Virtual Compass: Relative Positioning to Sense Mobile Social Interactions," in *Pervasive*, 2010.

[26] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based User Location and Tracking System User Location and Tracking System," in *INFOCOM*, 2000.

[27] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac, "Indoor location tracking using rssi readings from a single wi-fi access point," *Wireless Networks*, vol. 13, pp. 221–235, April 2007.