CoReCast: Collision Resilient Broadcasting in Vehicular Networks

Tanmoy Das★, Lu Chen★, Rupam Kundu★, Arjun Bakshi★, Prasun Sinha★, Kannan Srinivasan★,
Gaurav Bansal◇, Takayuki Shimizu◇
★The Ohio State University, ◇Toyota
{das.167, chen.4728, kundu.24, bakshi.11, sinha.43, srinivasan.115}@osu.edu,
{gbansal,tshimizu}@us.toyota-itc.com

ABSTRACT

Reliable and timely delivery of periodic V2V (vehicle-to-vehicle) broadcast messages is essential for realizing the benefits of connected vehicles. Existing MAC protocols for ad hoc networks fall short of meeting these requirements. In this paper, we present, CoReCast, the first collision embracing protocol for vehicular networks. CoReCast provides high reliability and low delay by leveraging two unique opportunities: no strict constraint on energy consumption, and availability of GPS clocks to achieve near-perfect time and frequency synchronization.

Due to low coherence time, the channel changes rapidly in vehicular networks. CoReCast embraces packet collisions and takes advantage of the channel dynamics to decode collided packets. The design of CoReCast is based on a preamble detection scheme that estimates channels from multiple transmitters without any prior information about them. The proposed scheme reduces the space and time requirement exponentially than the existing schemes. The system is evaluated through experiments with USRP N210 and GPS devices placed in vehicles driven on roads in different environments as well as using trace-driven simulations. It provides 15× and 2× lower delay than 802.11p and OCP (Omniscient Clustering Protocol), respectively. Reliability of CoReCast is 8× and 2× better than 802.11p and OCP, respectively.

CCS CONCEPTS

• **Networks** → *Link-layer protocols*; *Mobile ad hoc networks*;

KEYWORDS

Vehicular Networks, Preamble detection, CoReCast

ACM Reference Format:
Tanmoy Das★, Lu Chen★, Rupam Kundu★, Arjun Bakshi★, Prasun Sinha★, Kannan Srinivasan★, Gaurav Bansal◇, Takayuki Shimizu◇ ★The Ohio State University, ◇Toyota {das.167, chen.4728, kundu.24, bakshi.11, sinha.43, srinivasan.115}@osu.edu, {gbansal,tshimizu}@us.toyota-itc.com. 2018. CoReCast: Collision Resilient Broadcasting in Vehicular Networks. In *MobiSys '18: The 16th Annual International Conference on Mobile Systems, Applications, and Services, June 10–15, 2018, Munich, Germany.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3210240.3210341

1 INTRODUCTION

Every year 5 million car crashes happen on the roads in USA [56] and 30,000 people die from these crashes [56]. According to the studies published by National Highway Traffic Safety Administration (NHTSA), V2V communication can reduce the number of accidents by 80% [41] which prompted US Department of Transportation (USDOT) to launch the Connected Vehicle program to enable V2V communication. Vehicles collect important information using a range of sensors (camera, RADAR, LIDAR, GPS, etc.) installed on them. Vehicular applications share this information with all of its neighbors using broadcast communication so that all the vehicles can make effective driving and navigational decisions [54, 55, 58]. Such decisions can lead to reduction in accidents, stress-free driving [4], increased passenger comfort [53], increased fuel-efficiency [2], and reduced travel time [19]. Most of these applications generate packets at a high rate (100-200 packets/s) [1, 59] requiring low delay and high reliability. Reliability is measured by the number of neighbors that have successfully received a packet. A vehicular MAC protocol must satisfy both of these requirements.

The Dedicated Short Range Communication (DSRC) [57] standard for vehicular networks uses CSMA MAC protocol. But CSMA like collision avoiding distributed protocols [27, 35, 38, 47, 63] suffer from poor delay performance which grows exponentially [15, 46] with the number of nodes in the network, making them inadequate for many vehicular applications [11, 12, 31, 64]. Many protocols [10, 13, 23, 44, 49] emulate a TDMA like centralized solution to avoid packet collisions by forming dynamic clusters. However, they require additional information on topology for cluster formation and electing cluster-heads which requires significant communication overhead causing serious performance degradation [12, 31]. A plethora of collision-embracing MAC protocols [6, 16, 32, 45, 52, 66] are proposed for WiFi and cellular networks that leverage the wired backbone connection and the large coherence time to provide superior performance. But due to high mobility, such luxuries are not available in vehicular networks.

Given the shortcomings of the existing protocols, we propose the first distributed and collision embracing MAC protocol, CoReCast, for vehicular networks that provides both low delay and high reliability. CoReCast exploits three unique opportunities: 1) **Availability of GPS:** A vehicle is equipped with a GPS receiver because position is an important part of V2V applications. This GPS receiver is leveraged to achieve almost perfect time (300ns time offset) and

---

MobiSys '18, June 10–15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5720-3/18/06…$15.00
https://doi.org/10.1145/3210240.3210341

frequency (50Hz frequency offset) synchronization [37, 50, 51] in mobility. Near perfect time and frequency synchronization helps in packet decoding; 2) **Loose power constraint:** Power is not a strict constraint for vehicular networks. As a result, a vehicle can transmit the same packet multiple times to improve reliability; and 3) **Low Coherence time:** The coherence time of vehicular networks is very small (0.4ms [5]) because of high mobility. As a result, the wireless channel [1] between two nodes [2] changes independently in every packet transmission. In addition, the wireless channel between any two nodes is also independent at any point of time. CoReCast takes advantage of the channel independence to decode colliding packets from the neighbors.

Motivation: The main motivation of this work comes from the independence of the channels across time. The independence of the channel makes the decoding of collided packets easier. A node transmits the same packet multiple times and more than one node transmit concurrently in the same neighborhood resulting in packet collisions. At the receiver, multiple copies of the same packet are combined to increase the SINR of the packet (§ 4). The packet is decoded when the SINR is large enough. This packet decoding scheme has two advantages. 1) By combining multiple copies of the same packet, a receiver can decode packet from a node that has very bad channel (low SNR) to this receiver which improves the reliability; and 2) in vehicular networks, nodes use low data rate to ensure that all of its neighbors can receive a packet. Therefore, when a single node is transmitting the available bandwidth is not properly used. By allowing multiple nodes to transmit, the whole bandwidth is exploited for communication which improves the overall throughput and reduces the delay.

Main challenge: The main challenge in the realization of the decoding scheme is the estimation of the channels from the transmitting nodes. Due to low coherence time channel changes from one packet to another, a receiver needs to estimate the channel for every received packet. In general, when a node transmits a preamble for channel estimation there is no interference from other nodes which means same preamble can be used by all the nodes. But in vehicular networks due to low coherence time and lack of coordination, nodes can not transmit their preambles without collisions. Channels from nodes need to be estimated from colliding preambles. Therefore, a unique preamble is assigned to each node. Because of high mobility and lack of coordination, it is not known exactly which nodes are present in the neighborhood and are transmitting concurrently. Hence, the receiving node does not know which preambles are present in the colliding transmissions. In this scenario, conventional preamble detection and estimation schemes require either prohibitively large preambles (containing billions of bits) or enormous amount of memory (in order of TBs) and time (few hours).

Preamble detection: We propose a novel Compressive Sensing (CS) based preamble detection and estimation scheme in § 2.3. The proposed scheme uses a novel divide-and-conquer based solution to identify the transmitting nodes and estimate their channels. This scheme has following three advantages: 1) it exponentially reduces the space and time requirements; 2) it uses a significantly smaller preamble (hundreds of bits) than the existing schemes; and 3) it does not require any knowledge about the neighborhood. The proposed scheme can estimate the channels from colliding preambles very accurately as long as the number of transmitting nodes is less than a threshold. Our approach (§ 2) ensures that the number of concurrent transmissions is less than the threshold.

System description: CoReCast is designed on top of the preamble detection scheme. CoReCast is a synchronous MAC comprised of two phases. In the *estimation phase*, nodes calculate a backoff probability to control the number of concurrent transmissions. In the *data transmission phase*, nodes repetitively transmit the same packet. A packet contains a preamble which is used to estimate the channel from the transmitter every time it transmits a packet.

Contributions: This paper makes the following key contributions:

1) It presents a new preamble detection scheme for estimating channels from multiple colliding transmitters. The proposed scheme significantly reduces the length of the preambles and requires exponentially lower time and space than the existing schemes.
2) A collision embracing MAC protocol, CoReCast, is designed that leverages the preamble detection scheme to provide lower delay and higher reliability.
3) Through extensive experiments, we show that GPS devices can provide near-perfect time (300ns) and frequency synchronization (50Hz) even when the GPS signal is poor. The synchronization capability is exploited for packet decoding.
4) Performance of CoReCast is evaluated using a testbed containing three cars. A car is equipped with USRP N210s and Jackson Fury GPS device. Trace driven simulation is used to evaluate the performance of CoReCast in larger networks. CoReCast provides 15× lower delay than 802.11p and 2× lower delay than OCP (Omniscient Clustering Protocol). Reliability of CoReCast is 8× better than 802.11p and 2× better than OCP.

2 PREAMBLE DETECTION AND CHANNEL ESTIMATION

The main design component of CoReCast is preamble detection and channel estimation. This component is used to detect the colliding transmitters and estimate their channels. These estimated channel values are used to decode the collided packets. As mentioned before, the channels need to be estimated from colliding preambles that are processed to determine the identity of the nodes and their channels. The major constraints in the design of the preamble detection scheme are 1) due to lack of coordination, the scheme must work without any knowledge (identities of the neighbors) of the neighborhood; and 2) a receiver must estimate the channel for every received packet because the channel changes from one packet to another due to low coherence time. Our proposed design satisfies all the constraints.

2.1 Complexity of Detection and Estimation

Each node is assigned a unique B bit MAC address. So, the maximum number of nodes that can be supported is $N = 2^B$. A node is also assigned a unique preamble of length L. The set of N preambles are known to all the nodes. Consider $n \ll N$ nodes are transmitting

---

[1] Wireless channel is a complex number with a phase and an amplitude.

[2] Node and vehicle are used interchangeably.

at the same time. The objective is to estimate the channels from these transmitting nodes using colliding preambles. The L received samples from n nodes can be represented as:

$$\boldsymbol{y} = \boldsymbol{P}\boldsymbol{h} + \boldsymbol{w}. \tag{1}$$

Here $\boldsymbol{y}$ ($L \times 1$) is received samples, $\boldsymbol{P}$ is an $L \times N$ matrix (as n is not known) where each column contains one preamble, $\boldsymbol{h}$ is $N \times 1$ channel vector and $\boldsymbol{w}$ is $L\times 1$ noise vector. For preambles that are not present, the corresponding entries in $\boldsymbol{h}$ will be 0. Here $\boldsymbol{y}$ is observed, $\boldsymbol{P}$ is known and $\boldsymbol{h}$ is to be determined. By solving Equation 1, all the n preambles are detected and the channels associated with these preambles are also estimated. There are two significant challenges. 1) A node needs to store all the $N = 2^B$ preambles requiring a huge amount of memory (e.g., for $B = 48$, there are 2^{48} preambles). We propose a mapping function that generates a unique preamble from a MAC address. So, the nodes do not need to store the preambles but generate them as needed; and 2) the biggest challenge is the processing of the matrix $\boldsymbol{P}$, due to huge size ($L \times 2^B$) it can not be processed in one step.

2.2 Potential Approaches

Many approaches exist for colliding preamble detection. The adaptation of them in the vehicular networks have some significant difficulties that are described below.

1) Locally unique assignment: A service can be deployed to assign locally unique preambles in a neighborhood and to make a node aware of all the preambles present in its neighborhood. Such a service requires significant resources due to the inherent mobility in vehicular networks and the lack of central coordination.

2) Globally unique assignment: Another possible approach is to assign globally unique preambles to all the nodes. There are two possible options:

a) Orthogonal Codes: The orthogonal codes like Gold Code [28] and Walsh Code [62] have been used for preamble detection. As these codes are orthogonal, the length of the preambles (L) grows linearly with the total number of nodes (N). For 2^{48} cars, Gold codes of length $2^{48} - 1$ bits or Walsh Codes of length 2^{48} bits are required for a globally unique assignment of preambles.

b) CS based preamble detection: CS is a signal processing scheme to estimate the solutions of underdetermined systems (a matrix with more number of columns than the rows). CS requires that the columns of the matrix have low cross-correlation [18, 24]. A preamble detection scheme can be designed based on CS. To satisfy the requirement of CS, the preambles should have low cross-correlation. These preambles do not have to be orthogonal but they need to be unique so that they can be detected correctly. To ensure uniqueness, $L \geq B$. The main advantage of using CS is that we can choose a suitable value for L ($B \leq L \ll N$) to reduce the length of the preambles considerably. In this case, the number of rows (L) is smaller than the number of columns (N) in matrix $\boldsymbol{P}$. So, Equation 1 becomes underdetermined system which can be solved using CS. But as mentioned before, $\boldsymbol{P}$ still requires a huge amount of memory, making it practically challenging to create and use the matrix $\boldsymbol{P}$. Thus, CS cannot be used directly.

2.3 Divide-and-Conquer based solution

In order to solve Equation 1 in a practical manner, we iteratively build up a solution for it by solving many (say d) smaller sub-problems. In a sub-problem, the size of the matrix $\boldsymbol{P}$ is much smaller than $L \times N$. All the sub-problem's solution can be combined in the end to form the complete set of L bit preambles for the n nodes that are actually transmitting.

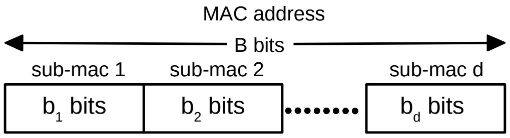


Figure 1: A MAC address of B bits is divided into d sub-macs. The first sub-mac contains first b_1 bits of the MAC address, the second sub-mac contains the next b_2 bits of the MAC address and so on.

Dividing the nodes into groups to create a hierarchy: To divide the problem into d smaller sub-problems, at first all the nodes (MAC addresses) are divided into multiple groups. These groups form a tree shaped hierarchy that has $d + 1$ levels. At the root, a single group contains all the nodes. The hierarchy is created as follows. The B bit MAC address is divided into d non-overlapping smaller parts, called sub-macs (Figure 1). The i^{th} sub-mac contains b_i bits and the sum of the bits over all the sub-macs is equal to B. All the nodes whose first i sub-macs are exactly same belong to the same group in the i^{th} level.

The nodes in a group are divided into disjoint sets and form the children of that group. To divide them into disjoint sets, $(i + 1)^{th}$ sub-mac is used that has b_{i+1} bits. The nodes are divided into $2^{b_{i+1}}$ disjoint sets (considering all possible combinations of b_{i+1} bits) and they become the children of the parent group. An example of group hierarchy is presented in Figure 2.

There are two important properties in this hierarchy. 1) The groups at any level divide the N nodes into disjoint sets, except at level 0 which has a single group; and 2) As the level increases the number of groups at a level increases. The group at the level 0 has 2^{b_1} children, a group at the first level has 2^{b_2} children and so on. The number of groups at level i, ($i \geq 1$) is $\prod_{j=1}^{i} 2^{b_j}$. These two properties imply that as the level increases, the number of nodes contained in a group decreases. Specifically, the number of nodes contained in a group at level i is $2^{B - \sum_{j=1}^{i} b_j}$. As a result, a group in the d^{th} level contains exactly one node.

A group at level i, ($i \geq 1$) is assigned a unique preamble, called sub-preamble. Given this arrangement, instead of solving a large problem to find which of the $N = 2^B$ nodes are present, the i^{th} sub-problem tries to determine which i^{th} level groups are present using the sub-preambles. The matrix processed by a sub-problem contains a much smaller number of sub-preambles than the undivided problem where a matrix contains N preambles.

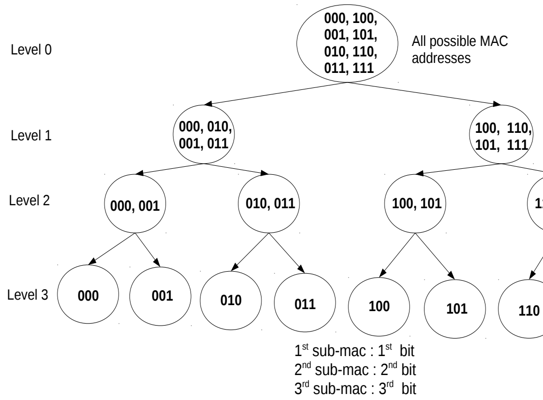


Figure 2: Here $B = 3$ and $d = 3$. There are 8 nodes (MAC addresses) and 3 sub-macs each containing a single bit. The i^{th} sub-mac contains the i^{th} bit of the MAC address. At level 0, a single group contains all the nodes. These nodes are divided into groups according to the 1^{st} sub-mac which contains the 1^{st} bit of the MAC address and has two different values '0' and '1'. As a result, nodes are divided into two groups: first group contains all the nodes whose 1^{st} bit is '0' and second group contains all the nodes whose 1^{st} bit is '1'. A group at level 1 is again divided into two groups according to the 2^{nd} sub-mac (2^{nd} bit of MAC address) and so on. At the last level, a group contains a single node.

The sub-tree rooted at a group in the i^{th} level that is found to be absent in the i^{th} sub-problem is ignored in the future sub-problems. When a group is detected by a sub-problem, the nodes present in that group are the possible choices for the transmitting nodes. To reduce the number of possible choices, these nodes are divided into disjoint sets which are represented by the children of the detected group. The next sub-problem detects which of these children groups are active.

Assigning sub-preambles to groups: Except the group at the top level, sub-preambles are assigned to all the groups. These sub-preambles must satisfy the following properties.

1) Uniqueness: All the groups in level i must have a unique sub-preamble so that they can be detected correctly in i^{th} sub-problem. There are $2^{\sum_{j=1}^{i} b_j}$ groups at level i. To ensure uniqueness, the length (l_i) of the i^{th} sub-preamble should be $l_i \geq \sum_{j=1}^{i} b_j$.

2) Low cross-correlation: The sub-preambles assigned to the groups at the same level must have low cross-correlation so that the requirement of the CS is satisfied.

The sub-preambles are generated using a mapping function. At level i, the mapping function takes the first i sub-macs as input and maps it to a unique sub-preamble of length l_i. A pseudo random number generator (PRNG) [61] is used as the mapping function which produces a sequence of bits given a seed value as input. A PRNG is used because of following two reasons: 1) for a given seed value it always produces the same sequence of bits which is required for the uniqueness in the mapping; and 2) PRNG produces a random sequence of bits which ensures low cross-correlation between sub-preambles. All the N nodes use the same mapping function so that they can determine the sub-preamble associated with a group easily.

Except the level 0, the groups are disjoint in every level and a node can only belongs to a single group at any level. Therefore, a node belongs to exactly d groups in the hierarchy starting from level 1. As a result, the node has d sub-preambles associated with it. The preamble of the node is created by concatenating these d sub-preambles. The length of the preamble is $L = \sum_{j=1}^{d} l_j$.

Identifying the transmitting nodes: The colliding nodes (n) transmit their L bit preambles and the receiver uses the L received samples to identify the nodes, their individual preambles, and the channels from them. The first sub-problem's goal is to use the first l_1 samples of the received signal, which contain the first level sub-preambles of the transmitting nodes, to identify which first level groups are present. At the first level, there are only 2^{b_1} possible groups. Therefore, it uses CS to solve Equation 1 for a smaller P of size $l_1 \times 2^{b_1}$ (instead of the original $L \times 2^B$). The h vector it solves for contains information about the following:

1) The presence or absense of the different groups at the first level.

2) For the groups found to be present, h also represents the combined channel value for the nodes in that group.

Suppose p_1 groups are found to be active. For a detected group, at least one of its children must be active but this information is not known apriori. So, all the children of all the p_1 groups are considered in the second sub-problem. There are $p_1 2^{b_2}$ such children and the sub-preamble length is l_2. It requires a matrix $\boldsymbol{P}$ of size $l_2 \times p_1 2^{b_2}$. Similarly, if p_{i-1} groups are detected in $(i-1)^{th}$ sub-problem, the size of matrix $\boldsymbol{P}$ in i^{th} sub-problem is $l_i \times p_{i-1} 2^{b_i}$. After solving the last sub-problem, the active groups at the d^{th} level are identified. As each group at the d^{th} level contains a single node, they uniquely identify the transmitting nodes.

Channel estimation: As the nodes and their preambles are correctly detected in the last stage, only then the channels from the transmitted nodes can be estimated. That is to say, the h vector computed in all other sub-problems are not channel estimates of unique nodes, but those of nodes belonging to the same group. Therefore, once the full MAC address and preamble of each actively transmitting node has been discovered in the last stage of the divide and conquer approach, a final channel estimation must be performed. This estimation process reformulates Equation 1 with P as an $L \times n$ matrix, where L is the length of the preambles, and n is the number of nodes actively transmitting.

More Improvement: There is one more advantage in the proposed design. As stated earlier, if a group is found to be active, then the channel value that CS estimates represents the combined channel of all the active nodes in that group. At least one of the children of that group has to be present in the next stage. The combined channel values of the children groups must be same as the channel value of the parent group because the nodes contained in a group and the union of the nodes contained in all of its children are exactly same. These constraints regarding channel values are added at the end of the **P** matrix as new rows. These additional rows do not violate the constraints of CS. Proof is provided in [22]. These extra constraints helped us to detect more colliding nodes while using smaller preambles.

2.4 Analysis

In this section, we present the analysis for the space complexity, time complexity and detection accuracy of the preamble detection scheme. To optimize the space and time complexity, we need to find the optimal values for l_i and $b_i, \forall i$. But l_i and b_i can have only integer values which means this optimization problem is an integer programming problem and very difficult to solve. As a heuristic, B bits of the MAC address is divided equally among $d, \ 1 \leq d \leq B$ parts thus $b_i = \frac{B}{d} = b, \ 1 \leq i \leq d$ and the length of sub-preambles in every level is same i.e., $l_i = l$. The preamble length is $L = dl$.

Space: The space complexity depends on the maximum size of the matrix $\boldsymbol{P}$ used across all the sub-problems. If the $(i-1)^{th}$ sub-problem detects p_{i-1} sub-preambles, then in the i^{th} sub-problem p_{i-1} extra rows are added to the matrix $\boldsymbol{P}$ as new constraints. So, the size of the matrix $\boldsymbol{P}$ becomes $(l_i + p_{i-1}) \times p_{i-1}2^{b_i}$. The number of detected sub-preambles at any level should be smaller than the length of the sub-preamble (l) [7] i.e., $p_{i-1} < l$. According to this constraint, the required matrix size is at most $2l \times l2^b$ in any sub-problem. The space requirement is $O(l^2 2^{\frac{B}{d}})$. Space requirement is reduced by a factor greater than $\frac{2^{B-\frac{B}{d}-1}d^2}{L}$.

Time: Computation time required by a sub-problem depends on the size of matrix **P**. Our implementation uses an algorithm based on interior-point method in [30] to solve the CS problem. The runtime complexity of the algorithm is $O\left(l^{\frac{5}{2}} 2^{\frac{3b}{2}}\right)$ for a sub-problem. The runtime complexity for d sub-problems is $O\left(dl^{\frac{5}{2}} 2^{\frac{3B}{2d}}\right)$. Runtime is reduced by a factor of $2^{\frac{3B}{2}(1-\frac{1}{d})} d^{\frac{3}{2}}$.

Detection accuracy: The preamble detection accuracy deteriorates as the number of levels in hierarchy increases. This poor performance of preamble detection can be explained by the fact that in any level i, a sub-preamble of length $l = \frac{L}{d}$ is created by a random mapping from a space of $2^{i\frac{B}{d}}$ to $2^{\frac{L}{d}}$. As d increases the size of $2^{\frac{L}{d}}$ decreases but the size of $2^{i\frac{B}{d}}, 1 \leq i \leq d$ remains similar, resulting in sub-preambles that are similar to each other. Consequently, the cross-correlation among the sub-preambles reduces which in turn adversely affect the performance of CS [17].

In summary, as the number of stages increases the space requirement and the computation time (number of operations) decreases exponentially as shown in Figures 3(c) and 3(d) but the detection accuracy reduces as shown in Figure 3(b) and Figure 3(a). If a threshold is provided for acceptable false positive and false negative ratios, then we can correspondingly determine the largest number of stages that meet those requirements in order to optimize space and time complexities.

3 CHALLENGES FOR IMPLEMENTATION

Due to high mobility and lack of coordination, there are other nontrivial problems that need to be addressed for the implementation of CoReCast. This section focuses on addressing these problems.

3.1 Controlling the number of colliding transmissions

The proposed preamble detection scheme can detect a limited number of colliding nodes with high accuracy as shown in Figures 3(a) and 3(b). But the number of nodes in a neighborhood can be larger than this number. To control the number of concurrent transmissions a backoff probability is calculated. It requires two steps that are described below.

Estimation of node density: All nodes are divided into G groups based on a hash computed using their MAC addresses. A packet containing G bits are transmitted from all the nodes. If a node belongs to $g^{th}, 1 \leq g \leq G$ group then the g^{th} bit in that packet is set to 1 and rest of them are set to zeros. By processing the received data, a receiver determines the number of bits that has a zero value. Then it estimates the number of nodes (N_E) in its neighborhood using a fast estimation technique described in [65]. Figure 4 shows percentage of error for the estimation technique which is quite small. The accuracy of the estimation depends on the value of G and the maximum number of nodes in the neighborhood.

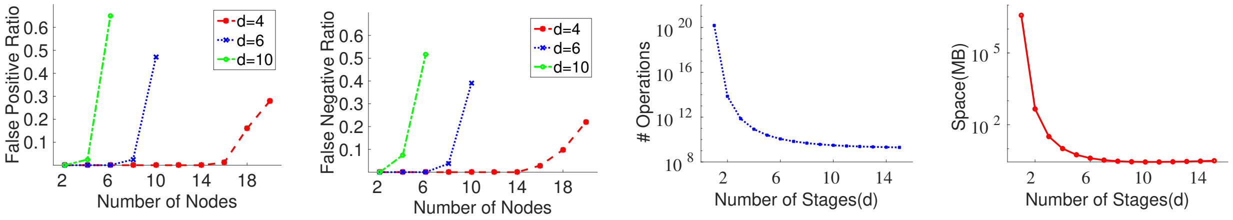
Estimation of skipping probability: Suppose N_S is the number of concurrent transmissions the preamble detection scheme can handle. Hence the probability of skipping (P_s) a transmission is computed as follows:

$$P_s = 1 - \frac{N_S}{\alpha N_E}, \alpha > 1 \tag{2}$$

The factor of α in the denominator is used to ensure that we are not operating near the inflexion point (in Figures 3(a) and 3(b)). As the nodes transmit independent of each other with probability $\frac{N_S}{\alpha N_E}$, the number of concurrent transmitters can be modeled as a binomial random variable X with parameters $\frac{N_S}{\alpha N_E}$ and N_E. For the case of $\alpha > 2$, using Chebyshev's inequality we can derive the probability that number of concurrent transmitters is greater than N_S as follows:

$$\begin{aligned} P(X \geq N_S) &= P(|X - \frac{N_S}{\alpha}| \geq N_S - \frac{N_S}{\alpha}) \\ &\leq \frac{\frac{N_S}{\alpha}(1 - \frac{N_S}{\alpha N_E})}{(N_S - \frac{N_S}{\alpha})^2} = \frac{1 - \frac{N_S}{\alpha N_E}}{\frac{N_S}{\alpha}(\alpha - 1)^2} \end{aligned} \tag{3}$$

By choosing a suitable α, we can make the probability of more than N_S simultaneous transmissions as small as possible. The skipping probability (P_s) is discretized into S levels and transmitted in a packet containing S bits. This information is transmitted to the neighboring nodes. A node uses the maximum skipping probability among all the neighbors (including itself) as its selected backoff probability, P_b. The most conservative skipping probability within the neighborhood is used to ensure that the constraint for the maximum number of concurrent transmissions is likely not violated in any neighborhood. By choosing appropriate values for G and S, the overhead for backoff probability estimation can be reduced as small as possible.



(a) False Positive Ratio for preamble detection.

(b) False Negative Ratio for preamble detection.

(c) Number of operations for different values of d.

(d) Space requirement for different values of d.

Figure 3: Performance for preamble detection for $B = 32$ and $L = 450$. The SNR of the preambles can vary from 6dB to 15dB. Number of operations and space requirement is also presented.

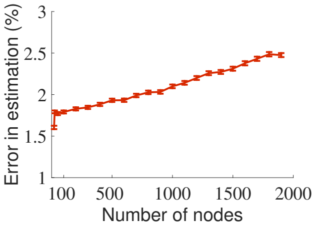


Figure 4: Performance of node density estimation.

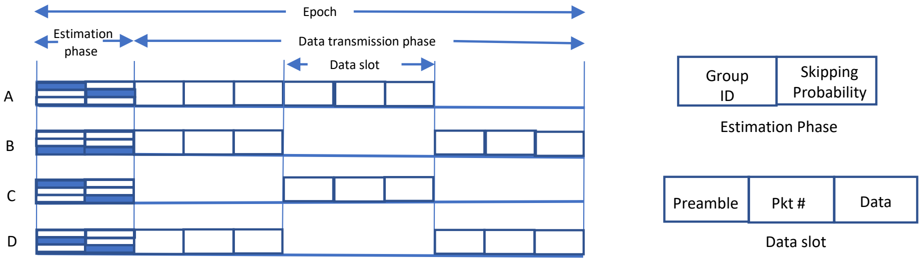


Figure 5: Transmissions from four nodes.

3.2 Distinguishing different packets from the same node

A receiver needs to distinguish two different packets from the same transmitter because nodes in the network can introduce a new packet at any time. After a packet is successfully decoded, its contribution is removed from the stored samples collected in the past and also from the samples that would be received in future. Removal of the samples from decoded packet, improve the SINR of other undecoded packets that can be decoded faster. A decoded packet should be subtracted only from those received samples where it is present. The presence of a preamble tells that a packet from the node is present in the received samples, but it does not distinguish two different packets from the same node. A packet number is used to distinguish two different packets. Since we only need to distinguish two consecutive packets, an alternating bit would suffice. But as packets from multiple nodes are potentially colliding, we use a sequence of '0's and a hash of the MAC address to encode the alternating bit.

3.3 Half-duplex vs Full-duplex communication

CoReCast can be implemented using both half-duplex and full-duplex communication. In half-duplex communication a node can not transmit and receive simultaneously which reduces the opportunities for receiving colliding packets from the neighbors. In contrast, when nodes use full-duplex communication they can transmit and receive at the same time which significantly increases the number of opportunities for a node to receive colliding packets from the neighbors. These additional colliding packets enhances the decoding performance.

Challenges for full-duplex communication: Full-duplex communication depends on self-interference cancellation which has been shown to work in static environments [3, 9, 21, 34], but few works have made an attempt to incorporate it into vehicular networks. There are three major components in full-duplex communication that cancel out the self-interference signal from transmit antenna to receive antenna. Digital cancellation is one of them. A node needs to estimate the self-interference channel to implement digital cancellation. But estimation of self-interference channel poses two problems: 1) unlike WiFi networks, there is no coordination in vehicular networks to safeguard the self-interference channel estimation against external interference from other nodes. In our experiments, the self-interference channel is measured jointly along with the channels from other nodes. For joint channel estimation the detected preambles and the preamble transmitted by the node are used; and 2) frequency of estimating the self-interference channel is determined by the coherence time. If the coherence time is small, we need to estimate the channel frequently. But the self-interference channel is dominated by the signals coming from the line-of-sight path and the path reflected off the roof of the car. The signal from these two paths do not change over time. As a consequence, we find out that the coherence time for self-interference channel is quite large.

4 PUTTING IT ALL TOGETHER

In this section, we provide the holistic design of CoReCast and some practical considerations that are important for the realistic implementation of the system.

4.1 Consolidated MAC layer

CoReCast is designed for MAC layer broadcasting. CoReCast is a synchronous MAC that leverages GPS devices for time synchronization. Time is divided into epochs and an epoch is divided into two phases: estimation phase and data transmission phase. One example of transmissions from four nodes is shown in Figure 5. The operation of the two phases are as follows.

Estimation Phase: In the estimation phase, the backoff probability to control the colliding transmissions is determined. This phase is divided into two slots. The first slot is used to estimate the number of nodes in a neighborhood and the second slot to advertise the calculated skipping probability to the neighboring nodes.

Data Transmission Phase: The data transmission phase consists of multiple data slots during which the nodes transmit their data packets. The backoff probability is used by a node to decide whether or not to transmit a packet in a slot. A packet is retransmitted a fixed number of times. The number of retransmissions determines the reliability and delay. As the number of retransmissions increases the reliability increases but the delay increases too. Depending on the requirement of the application, the number of retransmissions can be determined.

Decoding of packet over multiple slots: The signal received in each data slot is the collided packets from multiple transmitters. At the end of a data slot, the samples from recent data-slots (say, M data-slots) are jointly analyzed to decode as many packets as possible. For a node with K neighbors, the received data $\mathbf{R}$ can be modeled as $\mathbf{R} = \mathbf{AX} + \mathbf{W}$. $\mathbf{A}$ is a $M \times K$ channel matrix which is estimated over M data slots using the preambles in the collided packets. The $i^{th}, 1 \leq i \leq M$ row of matrix $\mathbf{A}$ contains the channel values of the neighbors during the i^{th} data slot. If a neighbor is not transmitting during the i^{th} data slot then its corresponding entry is zero. $\mathbf{X}$ and $\mathbf{W}$ are the data and noise matrices, respectively. We can estimate the matrix $\mathbf{X} \approx \mathbf{A}^{+}\mathbf{R}$ (here $\mathbf{A}^{+}$ is the pseudo-inverse of $\mathbf{A}$).

But not all the packets can be decoded at the first attempt. The packets with good SINR are decoded at first and those packets are subtracted out from all the data slots. Then a second attempt is made to decode the remaining packets. Due to near perfect frequency synchronization among the nodes, packet decoding and subtraction become much easier. CoReCast provides better throughput because it combines the information from multiple slots to improve the SINR of a received packet. In Figure 6 an example of packet decoding process is shown. Six colliding packets are decoded within three slots.

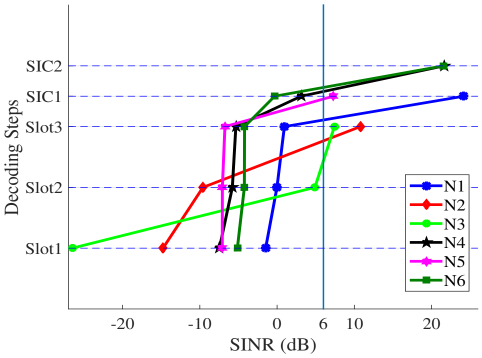


Figure 6: Six colliding packets are decoded within three slots. As the information from multiple slots are combined, the SINRs of the packets monotonically increase in each slot. In slot 3, packet #2 and #3 are decoded. Using SIC they are subtracted out from the previous slots. Then #1 and #5 are decoded and subtracted out using SIC. After that #4 and #6 are decoded. Thus, 6 packets are decoded in 3 slots. The threshold SINR for packet decoding is 6 dB.

4.2 Practical considerations

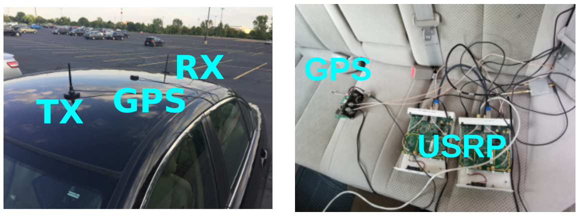
There are some practical issues to consider for the real deployment that are described here.

Number of data slots in an epoch: In each epoch, a backoff probability is estimated depending on the number of nodes in the neighborhood which is used to control the transmissions in a data slot. The neighborhood of a node must not change significantly during one epoch. Assume the maximum speed of a node is 85mph (≈38m/s). If we consider the head-on traffic the relative velocity between two vehicles is at most 76m/s. In our design, one epoch contains 20 data slots and one data slot is 0.4ms long. Within 8ms, the distance between two vehicles can increase by at most 0.6m. Hence, the change in the neighborhood is insignificant during one epoch.

Use of ACK message: There are two issues in using ACK that are explained here: 1) In a broadcast network, implementation of ACK is difficult because each node needs to transmit an ACK for a successfully received packet. Hence, for a packet that is received by all the neighbors, the number of ACK messages is the same as the number of neighbors. So, the total number of ACK messages is very large in a broadcast network and consumes a lot of resources; and 2) Use of ACK does not improve the performance significantly. Say there is a perfect implementation of the ACK mechanism which informs a node when its packet is received by all the neighbors. After a node is informed, it can start transmitting a new packet, but this new packet creates additional interference for the packets that are not yet decoded. As a result, the delay for that node decreases but the delay for the other nodes increases. On the other hand, CoReCast does not use ACK messages and allows a node to retransmit a packet only a fixed number of times. In CoReCast, when a specific packet from a node is decoded once, it is canceled out from the later slots even if received. The cancellation of decoded packets reduces the interference for other undecoded packets which can be decoded faster. As a result, the average delay performance of the whole system improves.

5 EXPERIMENTS

We build a prototype of CoReCast using the USRP-N210 SDR platform and SBX daughterboard. We use 2.49GHz carrier frequency to avoid potential interference and operate with 10MHz bandwidth. For time and frequency synchronization, we use Jackson Fury GPS[3] [37]. Experiments are performed on city roads in U.S. (speed limit varies from 15mph to 65mph) with normal traffic. The cars move in both same and opposite directions. Figure 7 shows our setup in one car.



(a) Outside.

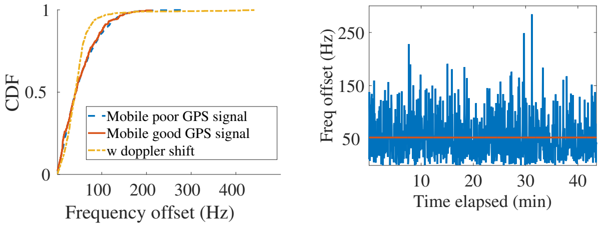
(b) Inside.

Figure 7: Experiment setup.

5.1 Micro Benchmark

We first check benchmark performance of components required to realize CoReCast. Following experiments test the GPS synchronization capabilities, performance of self-interference cancellation and the channel between two cars.

GPS based synchronization: In the design of CoReCast, we assume vehicles can achieve both time and frequency synchronization. A natural question is whether the use of GPS can achieve the desired level of synchronization. Our following experiments give positive answers to it.



(a) Frequency synchronization.

(b) Frequency synchronization with poor GPS signal. Mean frequency offset is 50Hz.

Figure 8: Frequency synchronization performance using GPS meets our requirements.

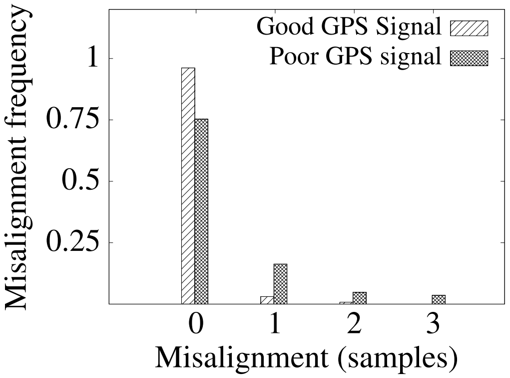


Figure 9: Time synchronization performance.

Frequency synchronization: The frequency synchronization performance is evaluated using two different setups to measure the effects of the doppler shift and poor GPS signal. In the first setup, we configure one car to transmit a constant sample value and another car to receive. In this setup, the frequency offset is affected by the doppler shift due to the relative speed between two cars. By examining the received samples, we can estimate the frequency offset using the technique described in [67]. In presence of doppler shift, the mean frequency offset value is close to 50Hz. But few times the frequency offset can be very high or low as doppler shift can introduce both positive and negative frequency shift as shown in Figure 8(a).

In the second setup, the receiver and transmitter are both on the same car so there is no doppler shift between them. But in this setup we intentionally disconnect the GPS antenna from the transmitter for several minutes (5-45 minutes) and then reconnect it again to model the effect of poor or unavailable GPS signal. Figure 8(a) shows the CDF of the frequency offset between the two nodes under different conditions. Without any doppler shift, frequency offset value is close to 50Hz even when the GPS signal is not available. These results are not surprising as the communication system uses crystal oscillators for clock which have extremely small (≈ 80pHz/day [36]) frequency drift. Once the crystal based oscillator's output is corrected it can remain synchronized even at the absence of the GPS signal.

Time synchronization: To test the time synchronization capability, we place two transmitters in one car (to minimize the effect of propagation delay) with each synchronized to a different GPS, while another car acts as the receiver. The Pulse-per-second (PPS) rising edge from the GPS is used to achieve time synchronization. A transmitter sends a preamble of length 500 samples to the receiver after every 200ms. The two preambles sent from two transmitters are uncorrelated. The start of the received preambles are detected using cross-correlation. This experiment is done over several hours. To model the situation when the GPS signal is not available or has very poor signal strengths, one of the GPS device at transmitter side is disconnected from the GPS antenna for several minutes (5-30 minutes) and then again reconnected. We repeat this process multiple times during our experiments. Figure 9 shows the time mismatch in terms of samples (at 10 MHz). When the GPS signal

---

[3] In fact, it is a GPS disciplined oscillator (GPSDO), we use GPS for simplicity.

strength is good, sample level synchronization is achieved 96% of time and when GPS signal strength is poor sample level synchronization is achieved 75% of time.

System performance under poor GPS signal: The frequency synchronization performance over time when the GPS signal is not available for 45 minutes is shown in Figure 8(b) which shows that the mean value of frequency offset remains same throughout. It should be noted that for 802.11 the residual frequency offset is 250Hz at 2.4GHz even after frequency offset correction [48]. Based on this observation, we conclude that *frequency synchronization meets our requirements*. In case of time synchronization, the maximum misalignment in samples when GPS signal is not available is 3 samples (300ns). Considering the fact that the CP length is usually much longer than this (e.g., 8 or 16 samples in IEEE 802.11), *the time synchronization capability of GPS is sufficient to meet our requirements*.

Self-interference Cancellation: We implement a real-time full-duplex system. The SBX daughterboards contains a single transmit and receive chain. A USRP N210 contains a single daughterboard. Our full-duplex system is based on the design proposed in [25] which includes two transmit chains and one receive chain. To implement this system, two USRPs are used as a single node. There are two transmit chains in this setup. First one is used for transmission and the second one is used for analog cancellation. These two USRPs are connected by a MIMO cable so that they can share the same clock. The transmit signal is 65-70dB higher than the noise floor. The self-interference cancellation consists of three components that are described below.

1) Antenna separation: Transmit and receive antennas are placed far apart as shown in Figure 7(a). Due to this separation, the transmit signal power reduces by 20dB.

2) Analog cancellation: Analog cancellation tries to cancel the direct path signal from transmit to receive antenna. The performance of the analog cancellation is shown in Figure 11. The analog cancellation achieves upto 23dB of cancellation on an average and maximum cancellation achieved is 30dB. The analog cancellation is not sufficient to cancel out the whole self-interference signal.

3) Digital Cancellation: Digital Cancellation tries to cancel the remaining self-interference signal after analog cancellation. Digital cancellation uses the self-interference channel that is measured jointly using the detected preambles transmitted by the other nodes and the preamble transmitted by the node itself. Figure 10, shows the performance of digital cancellation for different time duration between channel estimation. The channel is estimated after 1ms (referred to as Mobile at 1ms) and after 2 seconds (referred to as Mobile at 2s) when the car is moving. The performance of digital cancellation is compared with the scenario when the car is static. Digital cancellation achieves up to 25dB cancellation. It also reveals that when cars are moving, the cancellation is around 1dB worse than the static scenario, with a median cancellation of 21dB. After 2 seconds, the median performance goes down by 3dB with a much larger variance. Recall that in the design of CoReCast, we have the opportunity to retrain the self-interference channel whenever a node transmits a packet. We conclude that *self-interference cancellation is also feasible on vehicles even when the self-interference channel is estimated infrequently.* This result is expected since the line-of-sight path and the path reflected of the roof of vehicle dominate the self-interfering channel. These paths do not change over time.

Combining the three components, we can achieve 70dB of cancellation which enables the cars in our experimental setup to communicate with each other. It should be noted that this cancellation performance is similar to [26, 34] but better cancellation performance can be achieved using the design proposed in [9].

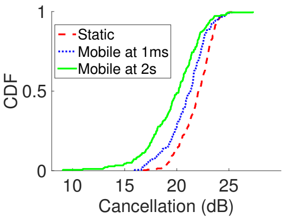


Figure 10: Digital cancellation performance in mobility.

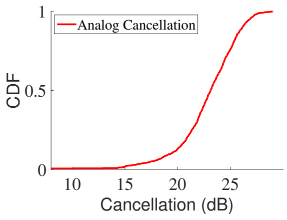


Figure 11: Analog cancellation performance.

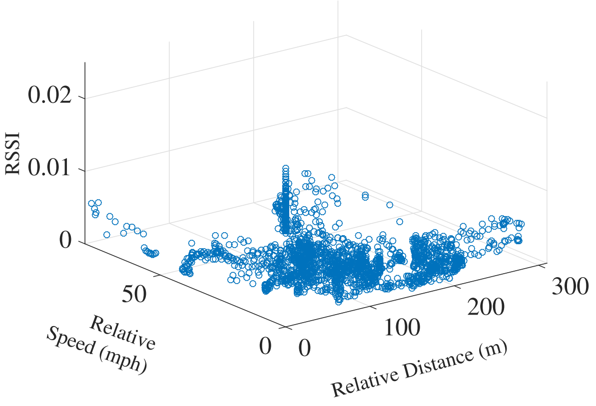


Figure 12: Channel between two cars.

Channel between two cars: To estimate the channel between two cars, a 500 sample preamble is sent from one car to another car repeatedly. At the receiver, this preamble is used to estimate the channel between the two cars using correlation. The position and the speed of the two cars are measured using two GPS devices. The cars move in the same and opposite directions under normal traffic conditions. The RSSI values for the channel is presented in Figure 12. These experimental results show that as the distance or relative speed between two cars increases the channel becomes worse (RSSI decreases). These results corroborate the results presented in [5]. These channel values are used to model the channel between two cars in § 6.

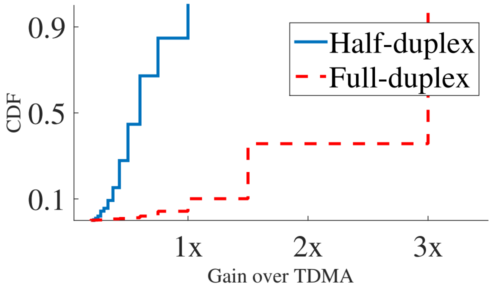


Figure 13: Throughput gain of CoReCast.

5.2 Throughput

Throughput is measured using a test bed containing three cars. In the vehicular networks, a node needs to communicate with all of its neighbors irrespective of the channel condition between two nodes. As a result, data rate is kept small (5Mbps, 1/2 convolutional encoding, BPSK) to ensure that all the neighbors can receive the transmitted packet. It is well known that if the channel between two nodes changes significantly during a packet transmission, the decoding process becomes difficult. In our experiments, we set the packet duration to be 0.4ms, which is the expected channel coherence time to avoid this problem. The experiments are performed using both full-duplex and half-duplex designs. The performance of CoReCast is compared against an optimal TDMA protocol where a node requires 3 slots to transmit and receive packets from all the neighbors. The results are shown in Figure 13.

Full-duplex: In full-duplex communication, a node can transmit and receive at the same time. Therefore, all the three nodes transmit simultaneously. Most of the times (64%), all the nodes can decode the packets from their neighbors within a single slot resulting in 2.25× gain over TDMA on an average.

Half-duplex: Nodes can not transmit and receive simultaneously in half-duplex communication. As a result, nodes use a backoff probability to control their transmissions. We repeat the same experiment using different backoff probability values and observe that optimal performance is achieved when backoff is 0.33. Due to lack of coordination, same set of nodes transmit multiple times but they can not receive packets from each other at the same time. Valuable time is wasted because of these redundant transmissions and the total time required by all the nodes to decode packets from the neighbors increases. On an average, nodes require 6 slots to decode packets from all the neighbors. This performance is 2× worse than the TDMA protocol.

Gain vs overhead of full-duplex: Full-duplex communication has extra overhead of channel estimation and cancellation which require extra processing leading to higher power consumption than half-duplex communication. But vehicles have loose power constraint and the throughput performance of CoReCast with full-duplex communication is 4.5× than half-duplex communication. So full-duplex communication with CoReCast is the preferred choice.

6 TRACE-DRIVEN SIMULATION

Our on-road experiments clearly show CoReCast can work in practical settings. However they are performed using a small set of nodes. In this section the performance of CoReCast is evaluated using simulation in a larger network with hundreds of cars. These simulations evaluate the scalability and the effect of mobility over time and space.

6.1 Simulation Setup

Packet level simulators like NS-3 [42] are unable to simulate any physical layer characteristics. CoReCast is a protocol whose performance is dependent on two critical physical layer techniques – decoding packets from colliding samples and SIC. For realistic evaluation, a sample level simulator is designed using MATLAB. A packet consists of samples which are sent from one node to another node. Samples are distorted due to the effect of channel and noise before they are processed at the receiver. Following traces from the experiments are used to model the realistic experimental setup. Nodes use full-duplex communication.

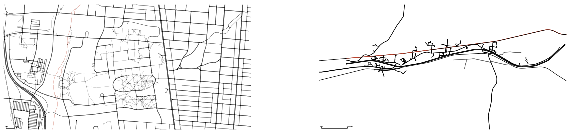
1) **Channel values**: At first the channel values collected during the experiments are partitioned into smaller sets according to the relative speed and relative distance. In simulation, the relative speed and relative distance between two cars are measured and according to those values the channel is uniformly selected from the set that has similar relative distance and relative speed.
2) **Time and frequency synchronization trace**: Time synchronization traces collected from GPS are used to model the packet transmission time from a node and frequency synchronization traces are used to model the frequency offset in the received packet.

The traffic simulator *SUMO* [43] is used to generate traffic traces that contain the speed and location of the vehicles. Figure 14(a) is a map for road topology in an urban locality that covers an area of 7.83km^2, and Figure 14(b) is a map for highway that covers an area of 138.36km^2. The relative distance and relative speed between two cars are shown in Figures 14(c) and 14(d). Neighbor density is shown in Figure 15(a). The maximum speed is 65mph. The packet size is 250 bytes including the overhead. The packet transmission time is 0.4ms. Every simulation is repeated for 15 iterations.

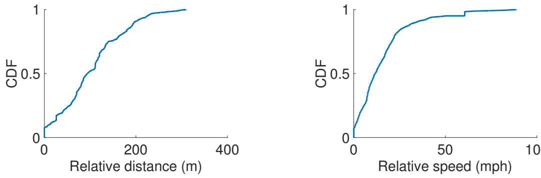
Vehicular MAC protocols can be divided into two categories: distributed and cluster based. Performance of CoReCast is compared with both types of protocols.

- **Distributed:** The distributed protocols use the transmission backoff to avoid the packet collisions. We have implemented two such protocols.
  1) **802.11p**: 802.11p DSRC protocol is implemented. The MAC layer used by this protocols is similar to CSMA/CA protocol.
  2) **UO-CSMA**: In recent years a variation of CSMA algorithm [35, 38] has been proposed that can achieve maximum throughput using a distributed scheduling algorithm. Each node is assumed to have a virtual queue. The distributed algorithm maximizes the throughput with the constraint of stabilizing the queue length across all the nodes.

- **Cluster based:** The cluster based protocols dynamically form a cluster and the nodes inside the cluster transmit one after another similar to TDMA to avoid collisions. But these protocols require large communication overhead to collect information about topology, form clusters and elect a cluster head. Additionally, these protocols can not avoid packet collisions in the overlapping area between two clusters and when a node moves from one cluster to another cluster due to mobility [12, 31].
  1) **Omniscient Clustering Protocol (OCP)**: A clustering protocol called OCP is implemented where all the nodes in the network are included in one cluster. An omniscient central scheduler knows the positions of all the nodes beforehand. As a result, OCP does not have any extra communication overhead. The scheduler chooses multiple nodes to transmit at the same time and ensures that there is no packet collision. To the best of our knowledge, OCP provides better performance than any cluster based protocols.



(a) **Local Road.** (b) **Highway.**

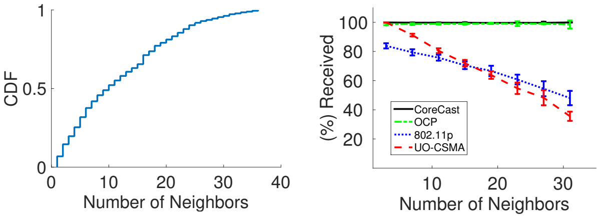


(c) **Relative distance between two cars.** (d) **Relative speed between two cars.**

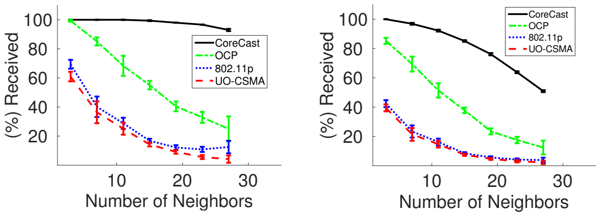
Figure 14: Road Topologies used for vehicle movement trace generation using *SUMO*. Relative speed and distance between two cars are also presented.

6.2 Reliability

Reliability is an important performance metric for V2V applications. Reliability is measured by the percentage of neighbors that are able to receive a packet from a node. Figures 15(b), 15(c) and 15(d) present the reliability of different protocols for different packet generation rates. To achieve 100% reliability, OCP requires as many slots as the number of nodes in the neighborhood. When packet generation rate is low (10 packets/s), there are many slots in between two new packets. As a result, OCP can achieve high reliability for low packet generation rate. On the other hand, CoReCast combines packets from multiple slots to provide even better performance than OCP. CoReCast provides 2× better reliability than OCP when packet generation rate is high. The reliability of 802.11p and UO-CSMA protocols decrease exponentially as the number of neighbors increases. The exponential decrease is evident when packet generation rate is greater than 100 packets/s. This exponential decrease is the result of backoffs and collisions caused by hidden and exposed terminals in the network [12, 31]. Reliability of 802.11p and UO-CSMA are not even 100% in smaller neighborhood (less than 5 nodes). As the density of the cars are not uniform (15(a)), a car from a smaller neighborhood can have some neighbors from an adjacent large neighborhood and it takes long time for 802.11p and UO-CSMA protocols to successfully transmit packets to those neighbors.



(a) Neighbor Density. Number of nodes within the transmission range. **(b) Reliability for packet generation interval of 100ms (10packets/s).**



(c) Reliability for packet generation interval of 10ms (100packets/s). **(d) Reliability for packet generation interval of 5ms (200packets/s).**

Figure 15: Reliability for different protocols.

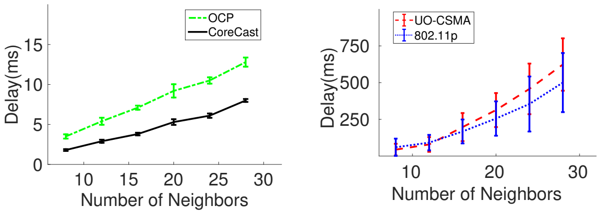
6.3 Delay

Delay is another important metric for V2V applications. Delay in reception of safety messages from neighbors can cause fatal accidents. Delay is measured as the time required for successful reception of a packet at all the neighbors. The delay performance is shown in Figures 16(a) and 16(b). The delay for 802.11p and UO-CSMA protocols increase exponentially [15, 46] as the number of neighbors increases. This increase in delay is the result of backoffs and packet collisions. CoReCast provides lowest packet transmission delay. The delay of CoReCast is 2× lower than the delay of OCP. CoReCast combines information from the colliding packets over multiple slots to decode packets faster than OCP.

7 RELATED WORK

This work is related to broadcast MAC protocols in ad hoc networks, which can be categorized as follows:

- **Contention Based:** DSRC uses CSMA as the medium access protocol. Several extensions based on CSMA have also been proposed [27, 35, 38, 47, 63]. For example, in [63], backoff time is based on priority to allow higher priority traffic



(a) Delay for CoReCast and OCP.

(b) Delay for 802.11p and UO-CSMA.

Figure 16: Delay required to reach all the neighbors. Delay grows linearly with number of neighbors for OCP and CoReCast. But it grows exponentially for 802.11p and UO-CSMA.

to access the medium faster than those with lower priority. [27] tries to predict the probability of a successful reception by using channel feedback. The authors in [47] propose a mechanism for collision notification. Although extensive efforts have been made towards CSMA based schemes, CSMA is known to suffer from exponential delay problem which is especially severe in a broadcast network.

- **Access Orthogonalization:** Orthogonalization can be done in time, frequency or code. ADHOC MAC [14] is a dynamic TDMA scheme that can allocate time slots in a distributed way, which is free from exposed and hidden terminal problems. It enables reliable transmission with feedback from other nodes. [40] further improves performance by changing the fixed slot length dynamically. However, these schemes still suffer from problems including high coordination cost.
- **Interference Embracing:** In recent years a wide range of MAC protocols [6, 16, 32, 45, 52, 66] have been proposed, especially for Wi-Fi networks, that are interference resilient. These approaches include successive interference cancellation (SIC) [32], uplink Multiuser MIMO (MU-MIMO) using zero-forcing and SIC [52], downlink MU-MIMO by precoding across APs in Enterprise WLANs (EWLANs) [45], interference alignment [16, 29], beamforming among APs in EWLANs [66], exploring hidden terminals [6], etc. However, these techniques cannot be applied directly to vehicular networks because it is highly mobile, has no backbone to exchange data, and it is hard to achieve coordination without any arbitration node.
- **Self-interference cancellation:** Our work is also related to full duplex communication [3, 8, 9, 20, 21, 26, 34] which has gained much attention in recent years. However, these works are confined to point-to-point communications in static environment, while our paper uses self-interference cancellation in a mobile broadcast network. In broadcast, nodes experience interference from other transmitters along with its self-interference. To the best of our knowledge, we made the first attempt to apply self-interference cancellation technique in an adhoc broadcast network.
- **CS for preamble detection:** CS based preamble detection has been used by others in different scenarios. In [39], an AP assigns randomly generated preambles to the clients associated with it and uses compressive sensing to estimate the channel from those preambles in a single slot. Similarly in Buzz [60], an RFID reader assigns temporary preambles to the RFID tags and use these preambles to estimate channels from the tags. In both cases, the network is static, coordination is required for the assignment of the preambles and the coordinating node already knows the set of possible preambles which are comparatively small. In vehicular networks it is non-trivial to know exactly which vehicles are present in the network and assign them unique preambles. In [33], a preamble detection algorithm is proposed that can detect a small number of preambles out of a very large space, in particular it requires a preamble of length 2^m to create a set of $2^{\frac{m(m+1)}{2}}$ unique preambles. However, these preambles cannot be divided into sub-preambles, which is needed to reduce space requirements.

8 CONCLUSION

A collision embracing MAC protocol called CoReCast, is proposed for vehicular networks. CoReCast exploits the abundance of power for communication in vehicular networks and availability of GPS for time/frequency synchronization. A novel preamble detection scheme is proposed that can estimate the channels from transmitters without any prior information about them. CoReCast combines the collided packets from multiple slots to decode them faster than OCP. It provides 15× and 2× lower delay than 802.11p and OCP, respectively. The reliability of CoReCast is 8× and 2× better than 802.11p and OCP for high packet generation rate (200packets/s).

9 ACKNOWLEDGEMENT

This work was partially supported by NSF grant # CNS-1302620, NSF grant # CNS-1618520 and ONR grant # N00014-17-1-2412.

REFERENCES

[1] 2018. 5G Automotive Vision. https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf. (2018).

[2] M. Alsabaan, K. Naik, and T. Khalifa. 2013. Optimization of Fuel Cost and Emissions Using V2V Communications. *IEEE Transactions on Intelligent Transportation Systems* (2013).

[3] Ehsan Aryafar, Mohammad Amir Khojastepour, Karthikeyan Sundaresan, Sampath Rangarajan, and Mung Chiang. 2012. MIDU: Enabling MIMO full duplex. In *In Proc. of ACM MOBICOM*. 257–268.

[4] autoconnectedcar.com. 2018. Self-Driving Autonomous Car Riders Will Look Out Windows, Relax and Be Stress Free. http://www.autoconnectedcar.com/2017/08/self-driving-autonomous-car-riders-will-look-out-windows-relax-and-be-stress-free/. (2018).

[5] Fan Bai, Daniel D. Stancil, and Hariharan Krishnan. 2010. Toward Understanding Characteristics of Dedicated Short Range Communications (DSRC) from a Perspective of Vehicular Network Engineers. In *In Proc. of ACM MOBICOM*. 329–340.

[6] Tarun Bansal, Bo Chen, Prasun Sinha, and Kannan Srinivasan. 2003. Symphony: Cooperative Packet Recovery over the Wired Backbone in Enterprise WLANs. In *In Proc. of ACM MOBICOM*.

[7] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. 2008. "A Simple Proof of the Restricted Isometry Property for Random Matrices". *Constructive Approximation* 28, 3 (2008), 253–263.

[8] Dinesh Bharadia and Sachin Katti. 2014. Full duplex MIMO radios. In *In Proc. of NSDI*. 359–372.

[9] Dinesh Bharadia, Emily McMilin, and Sachin Katti. 2013. Full duplex radios. In *In Proc. of ACM SIGCOMM*, Vol. 43. 375–386.

[10] S. Bharati and W. Zhuang. 2013. CAH-MAC: Cooperative ADHOC MAC for Vehicular Networks. *IEEE Journal on Selected Areas in Communications* (2013).

[11] K. Sjoberg Bilstrup, E. Uhlemann, and E. G. Strom. 2010. Scalability Issues of the MAC Methods STDMA and CSMA of IEEE 802.11p When Used in VANETs. In *IEEE ICC Workshops*.

[12] M. J. Booysen, S. Zeadally, and G. J. van Rooyen. 2011. Survey of media access control protocols for vehicular ad hoc networks. *IET Communications* (2011).

[13] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta. 2003. ADHOC MAC: new MAC architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services. (2003).

[14] Flaminio Borgonovo, Antonio Capone, Matteo Cesana, and Luigi Fratta. 2004. ADHOC MAC: new MAC architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services. *Wireless Networks* 10, 4 (2004), 359–366.

[15] C. Borgs, J. T. Chayes, A. Frieze, Jeong Han Kim, P. Tetali, E. Vigoda, and Van Ha Vu. 1999. Torpid mixing of some Monte Carlo Markov chain algorithms in statistical physics. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*. 218–229.

[16] Viveck R Cadambe and Syed A Jafar. 2008. Interference Alignment and the Degrees of Freedom for the K User Interference Channel. *IEEE Transactions on Information Theory* (2008).

[17] Emmanuel J Candes, Yonina C Eldar, Deanna Needell, and Paige Randall. 2011. Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis* 31, 1 (2011), 59–73.

[18] Emmanuel J. Candés, Justin K. Romberg, and Terence Tao. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics* 59, 8 (2006), 1207–1223.

[19] CMU-Penn University Transportation Center. 2018. Using V2V and V2I to Reduce Road Congestion via Parking Management. https://goo.gl/rq1Cqo. (2018).

[20] Lu Chen, Fei Wu, Jiaqi Xu, Kannan Srinivasan, and Ness Shroff. 2017. BiPass: Enabling End-to-End Full Duplex. In *In Proc of MobiCom*.

[21] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. 2010. Achieving single channel, full duplex wireless communication. In *In Proc. of ACM MOBICOM*. 1–12.

[22] Tanmoy das. 2018. CoReCast: Technical Report. https://goo.gl/GPKAdR. (2018).

[23] R. Ding and Qing-An Zeng. 2009. A clustering-based multi-channel Vehicle-to-Vehicle (V2V) communication system. In *2009 First International Conference on Ubiquitous and Future Networks*.

[24] D.L. Donoho. 2006. Compressed sensing. *Information Theory, IEEE Transactions on* 52, 4 (April 2006), 1289–1306.

[25] M. Duarte and A. Sabharwal. 2010. Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results. In *In Proc of Asilomar*.

[26] Melissa Duarte and Ashutosh Sabharwal. 2010. Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*. IEEE, 1558–1562.

[27] Shane B. Eisenman and Andrew T. Campbell. 2007. E-CSMA: Supporting Enhanced CSMA Performance in Experimental Sensor Networks using Per-neighbor Transmission Probability Thresholds. In *In Proc. of IEEE INFOCOM*. 1208–1216.

[28] R. Gold. 1967. Optimal binary sequences for spread spectrum multiplexing (Corresp.). *IEEE Transactions on Information Theory* 13, 4 (October 1967), 619–621.

[29] Shyamnath Gollakota, Samuel David Perli, and Dina Katabi. 2009. Interference Alignment and Cancellation. In *In Proc. of ACM SIGCOMM*.

[30] Michael Grant and Stephen Boyd. 2014. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. http://cvxr.com/cvx. (March 2014).

[31] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane. 2015. TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues. *IEEE Communications Surveys Tutorials* (2015).

[32] Daniel Halperin, Thomas Anderson, and David Wetherall. 2008. Taking the Sting out of Carrier Sense: Interference Cancellation for Wireless LANs. In *In Proc. of ACM MOBICOM*. 339–350.

[33] S. D. Howard, A. R. Calderbank, and S. J. Searle. 2008. A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes. In *In Proc. of Information Sciences and Systems,*. 11–15.

[34] Mayank Jain, Jung Il Choi, Taemin Kim, Dinesh Bharadia, Siddharth Seth, Kannan Srinivasan, Philip Levis, Sachin Katti, and Prasun Sinha. 2011. Practical, real-time, full duplex wireless. In *In Proc. of ACM MOBICOM*. 301–312.

[35] L. Jiang and J. Walrand. 2010. A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks. *IEEE/ACM Transactions on Networking* 18, 3 (June 2010), 960–972.

[36] Jackson Labs. 2018. Jackson Labs, Fury GPSDO. http://jackson-labs.com. (2018).

[37] Jackson Labs. 2018. M12M Receiver. https://goo.gl/CGyLpP. (2018).

[38] J. Lee, J. Lee, Y. Yi, S. Chong, A. Proutiére, and M. Chiang. 2009. Implementing utility-optimal CSMA. In *In Proc. of Allerton Conference on Communication, Control, and Computing, 2009. Allerton 2009*. 102–111.

[39] T. H. Lin and H. T. Kung. 2013. Concurrent channel access and estimation for scalable multiuser MIMO networking. In *In Proc. of IEEE INFOCOM,*. 140–144.

[40] Jia Liu, Fengyuan Ren, Limin Miao, and Chuang Lin. 2011. A-ADHOC: An adaptive real-time distributed MAC protocol for Vehicular Ad Hoc Networks. *Mobile Networks and Applications* 16, 5 (2011), 576–585.

[41] NHTSA. 2018. U.S. DOT advances deployment of Connected Vehicle Technology to prevent hundreds of thousands of crashes. https://www.nhtsa.gov/press-releases/us-dot-advances-deployment-connected-vehicle-technology-prevent-hundreds-thousands. (2018).

[42] NSNAM. 2018. NS-3. https://www.nsnam.org/. (2018).

[43] Institute of Transportation Systems. 2018. Simulation of Urban MObility. https://goo.gl/ES3ZS8. (2018).

[44] H. A. Omar, W. Zhuang, and L. Li. 2013. VeMAC: A TDMA-Based MAC Protocol for Reliable Broadcast in VANETs. *IEEE Transactions on Mobile Computing* (2013).

[45] Hariharan Shankar Rahul, Swarun Kumar, and Dina Katabi. 2012. JMB: scaling wireless capacity with user demands. In *In Proc. of ACM SIGCOMM*. 235–246.

[46] Shreevatsa Rajagopalan, Devavrat Shah, and Jinwoo Shin. 2009. Network Adiabatic Theorem: An Efficient Randomized Protocol for Contention Resolution. In *In Proc. of Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '09)*. 133–144.

[47] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. 2010. CSMA/CN: Carrier Sense Multiple Access with Collision Notification. In *In Proc. of ACM MOBICOM*. 25–36.

[48] Essam Sourour, Hussein El-Ghoroury, and Dale McNeill. 2004. Frequency Offset Estimation and Correction in the IEEE 802.11 a WLAN. In *In Proc. of Vehicular Technology Conference (VTC),*, Vol. 7. 4923–4927.

[49] H. Su and X. Zhang. 2007. Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology* (2007).

[50] Sundar Subramanian, Marc Werner, Shihuan Liu, Jubin Jose, Radu Lupoaie, and Xinzhou Wu. 2012. Congestion Control for Vehicular Safety: Synchronous and Asynchronous MAC Algorithms. In *In Proc. of ACM International Workshop on Vehicular Inter-networking, Systems, and Applications (VANET '12)*. 63–72.

[51] SXBlue. 2018. Rugged, Bluetooth Sub-Meter Mapping Receiver. https://goo.gl/1UXrKD. (2018).

[52] Kun Tan, He Liu, Ji Fang, Wei Wang, Jiansong Zhang, Mi Chen, and Geoffrey M Voelker. 2009. SAM: enabling practical spatial multiple access in wireless LAN. In *In Proc. of ACM MOBICOM*. 49–60.

[53] Gopi Krishna Tummala, Derrick Cobb, Prasun Sinha, and Rajiv Ramnath. 2016. Soft-swipe: Enabling High-accuracy Pairing of Vehicles to Lanes Using COTS Technology. In *In Proc. of CarSys*. 62–63.

[54] Gopi Krishna Tummala, Dong Li, and Prasun Sinha. 2016. RoadMap: Mapping Vehicles to IP Addresses Using Motion Signatures. In *In Proc. of CarSys*. 30–37.

[55] G. K. Tummala, D. Li, and P. Sinha. 2017. Roadview: Live View of On-Road Vehicular Information. In *In Proc. of SECON*. 1–9.

[56] USDOT. 2018. Connected Vehicle Basics. https://goo.gl/nMwSFc. (2018).

[57] USDOT. 2018. DSRC: The Future of Safer Driving. https://goo.gl/g2LRsq. (2018).

[58] USDOT. 2018. Improving Safety and Mobility Through Connected Vehicle Technology. https://goo.gl/9W4Gkx. (2018).

[59] A. Vinel, E. Belyaev, O. Lamotte, M. Gabbouj, Y. Koucheryavy, and K. Egiazarian. 2013. Video transmission over IEEE 802.11p: Real-world measurements. In *2013 In Proc of ICC)*.

[60] Jue Wang, Haitham Hassanieh, Dina Katabi, and Piotr Indyk. 2012. Efficient and Reliable Low-power Backscatter Networks. In *In Proc. of ACM SIGCOMM*. 61–72.

[61] Wikipedia. 2018. Soft-Swipe: Enabling High-Accuracy Pairing of Vehicles to Lanes using COTS Technology. https://goo.gl/oetpFT. (2018).

[62] Wikipedia. 2018. Walsh- Hadamard. https://goo.gl/9A8UaP. (2018).

[63] Su Yang, Hazem H Refai, and Xiaomin Ma. 2005. CSMA based inter-vehicle communication using distributed and polling coordination. In *In Proc. of Intelligent Transportation Systems,*. IEEE, 167–171.

[64] Jijun Yin, Tamer ElBatt, Gavin Yeung, Bo Ryu, Stephen Habermas, Hariharan Krishnan, and Timothy Talty. 2004. Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks. In *In Proc. of ACM VANET*.

[65] Kyu young Whang, Brad T. Vander-zanden, and Howard M. Taylor. 1990. A linear-time probabilistic counting algorithm for database applications. *ACM Transactions on Database Systems* 15 (1990), 208–229.

[66] Wenjie Zhou, Tarun Bansal, Prasun Sinha, and Kannan Srinivasan. 2014. BBN: Throughput Scaling in Dense Enterprise WLANs with Bind Beamforming and Nulling. In *In Proc. of ACM MOBICOM*. 165–176.

[67] Wenjie Zhou, Tanmoy Das, Lu Chen, Kannan Srinivasan, and Prasun Sinha. 2016. BASIC: Backbone-assisted Successive Interference Cancellation. In *In Proc. of ACM MOBICOM (MobiCom '16)*.