Wireless Network Compression Via Memory-Enabled Overhearing Helpers

Ahmad Beirami, *Student Member, IEEE*, Mohsen Sardari, and Faramarz Fekri, *Senior Member, IEEE*

***Abstract*—Traces derived from real-world traffic show that significant redundancy exists at the packet level in mobile network traffic. This has inspired new solutions to suppress the redundancy present in the packet data to manage the explosive traffic. In this paper, we propose a novel approach to performing redundancy elimination by employing universal compression using memory-enabled overhearing helpers without backhaul connectivity, referred to as wireless network compression. The helpers overhear the data packets previously sent by the wireless gateway to various mobile clients within their coverage and use them as side information to reduce the overall communication cost. We study wireless network compression via overhearing helpers from an information-theoretic point of view and conclude that this approach potentially offers a threefold benefit: 1) offloading the wireless gateway and hence increasing the maximum number of mobile nodes the gateway can reliably serve; 2) reducing the average packet delay; and 3) improving the overall throughput in the network.**

***Index Terms*—Cooperative communication, network compression, overhearing helper nodes, redundancy elimination, two-part codes, wireless network.**

I. Introduction

MOBILE data efficiency is an important feature of wireless communication. It increasingly draws attention as providers face the difficulty of handling the growing demand and look for solutions to reduce the data delivery costs in wireless networks. One potential solution is to eliminate the redundant data that is being transmitted to clients through the bottleneck of the network, the most important being the last hop: the wireless link from the wireless gateway to the mobile client. IP-layer redundancy elimination (RE), in the form of repetition suppression for a single client, has been successful in traffic reduction [1]–[8]. In particular, it was found that RE can suppress as much as 20-50% of the data traffic when 50% redundancy is present in the data [1], [5], [6]. In another study [8], based on data traces collected from both laptop and smartphone users over a span of three months, authors show that an average of 20% traffic reduction is achieved within each user's data trace using redundancy elimination techniques. On the other hand, these redundancy elimination solutions are confined to de-duplication of repeated patterns. While de-duplication is effective in removing long repeated bit sequences, it ignores the sub-packet level statistical dependencies in the data that are not mere repetitions (see [9]).

In [9], we established that redundancy in network traffic exists in the form of statistical dependencies beyond exact duplicates. As such, in [9]–[11], we took the first steps towards characterizing the achievable benefits of compression-based redundancy elimination. Data compression (source coding) is a natural candidate for statistical redundancy elimination. However, traditional compression techniques would not be very effective when applied to network packets. The reasons are the following: 1) redundancy within a packet cannot be effectively removed due to small size of the packets [12], [13]; and 2) traditional compression methods cannot leverage the redundancy across clients as they compress each packet independently from other packets [9]. In [9], [10], we formulated the redundancy elimination as *network compression* and introduced a new framework for compression of network data called *memory-assisted compression*. It was shown that universal compression-based methods can complement de-duplication-based redundancy elimination techniques to suppress an even more substantial amount of redundancy in the network. This was also experimentally confirmed on real data gathered from network traffic [9]. Note that a combination of memory-assisted compression and parallel compression techniques that achieve high compression rate as well as high compression speed make compression-based redundancy elimination feasible on high rate links as well [14], [15].

In this paper, we propose *wireless network compression via memory-enabled overhearing helpers*, which is inspired from compression-based redundancy elimination but differs significantly in the network architecture by using passive overhearing helpers without backhaul connectivity. This was first presented in [16]. Fig. 1 demonstrates the most basic scenario involving a single wireless gateway S, a mobile client C and a helper M. The memory-enabled helpers are small, possibly cooperative nodes with sufficiently large storage space, that are used to memorize the overheard packets previously transmitted from the wireless gateway to mobile clients. The overhearing capability of helper nodes comes at no extra cost (in terms of bandwidth usage) due to the broadcast nature of

Manuscript received February 15, 2015; revised May 26, 2015; accepted August 3, 2015. Date of publication August 14, 2015; date of current version January 7, 2016. This work was supported in part by NSF under Grant No. CNS-1017234. This work has been presented in part at the 2013 Information Theory and Applications Workshop and the 2014 IEEE International Symposium on Information Theory. The associate editor coordinating the review of this paper and approving it for publication was Prof. Emre Koksal.

A. Beirami was with Georgia Institute of Technology, Atlanta, GA 30332 USA. He is now with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA, and also with the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ahmad.beirami@duke.edu; beirami@mit.edu).

M. Sardari was with Georgia Institute of Technology, Atlanta, GA 30332. He is now with the Electronic Arts, Inc., Redwood City, CA 94065 USA (e-mail: sardari@gatech.edu).

F. Fekri is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: fekri@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2015.2468729

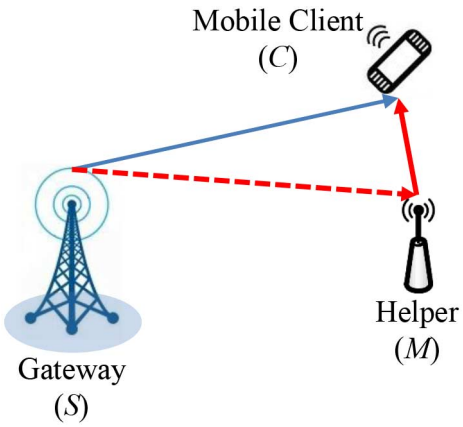

Fig. 1. An illustrative example of a wireless network with a single helper (deployed memory-enabled helper). A short-lived connection to the source by a mobile client is shown by a solid arrow. Overhearing is shown by a dashed arrow. The link supplementing side-information is shown by a thick solid arrow.

the wireless communication. The overhearing nature of the helper also eliminates the need for backhaul connectivity while offering throughput enhancement by theing the gateway.[1] The motivating assumption behind this work is that the communication in the helper-client link is less costly than that of the gateway-client link.

We analyze the achievable average codeword length region defined as the pair of codeword lengths communicated by the server and the helper nodes to the client node; and we derive converse bounds that quantify what codeword length pairs are hopeless to achieve. We further propose a coding strategy based on two-part codes that can achieve close to optimal performance in certain scenarios, and show that more than 50% cost reduction may be achieved by offloading the wireless gateway. The error-prone wireless environment makes it difficult to guarantee that the sender node and the helper node (that has overheard the previous communication of the source with other mobile clients) share the same model for the information source. This is because the error recovery mechanism is implemented between the source and the client to which the packet is destined to, and no error recovery mechanism is assumed in the gateway-helper (S-M) link. This can potentially result in a mismatch between the source model at the encoder and the decoder, which in turn makes the memory-assisted compression challenging. The impact of mismatched memory is also addressed in this paper.

The rest of the paper is organized as follows. In Section II, we present the background on universal compression. In Section III, we present the abstraction of the wireless network compression via memory-enabled overhearing helpers. In Section IV, we define the achievable codeword length region for the problem and present the converse bounds. In Section V, we present a two-part coding scheme whose performance is close to optimal in certain scenarios. In Section VI, we present simulation results performed in ns-2 simulator. In Section VII, we study the impact of mismatched memory. Finally, Section VIII concludes the paper.

[1]The focus of this paper is to increase bandwidth efficiency and we ignore the deployment cost and power requirements of the helpers.

II. Background and Related Work

In this section, we first review the background on universal compression that is needed for the technical presentation of our results and then review the related work.

A. Background on Universal Compression

Let S be an i.i.d. (memoryless) source over alphabet $\mathcal{X}$, with a $(|\mathcal{X}|-1)$-dimensional parametric vector θ which takes values in the d-dimensional simplex $\Theta \subset \mathbb{R}^d$ of all d-dimensional probability vectors, where $d = (|\mathcal{X}|-1)$ denotes the dimension of the unknown source parameter vector. One may extend this model to a more realistic setup for real-world sources by considering a mixture of parametric sources (with finite memory) as described in [17], [18]. Note that the side information (through memorized packets) primarily helps to remove the universal compression overhead, which is already significant for short memoryless sources. See Appendix A for a more detailed discussion on parametric source models.

Let μ_θ be the probability density function of the source parametrized by the d-dimensional parametric vector θ. Let $x^n = (x_1, \ldots, x_n)$ be a sequence generated by the source with probability $\mu_\theta(x^n)$. By this setup, for example, for a Bernoulli (binary memoryless) source which is represented with a single parameter $\theta = P[X = 1]$, the probability that the source would output the sequence x^n with k ones and $(n-k)$ zeroes is given by $\mu_\theta(x^n) = \theta^k(1-\theta)^{n-k}$.

If the parameter vector $\theta \in \Theta$ was known, the ideal code length of a packet x^n, obtained from the Shannon code [19] (ignoring the integer code length requirement), would be $\log(1/\mu_\theta(x^n))$.[2] On the other hand, since in practice the parameter θ is not known a priori, we wish to encode the packet using a universal probability distribution $P(x^n)$ that does not depend on the true θ, while it is "close" to the true unknown probability distribution $\mu_\theta(x^n)$ for all $\theta \in \Theta$. Although there is an extensive literature in the source coding community to address this problem, due to reasons that will be revealed in the sequel, we are interested in the *universal two-part codes* that provide a practical solution to this problem with close to optimal code lengths (see [13], [20], [21]).

In the absence of side information, x^n is universally coded using $c : \mathcal{X}^n \to \{0,1\}^*$ with the length function denoted by $l(x^n)$ that is prefix-free (no codeword is the prefix of any other codeword). $l(x^n)$ is simply the length of the codeword associated with x^n. It is well known that a necessary and sufficient condition for the existence of a prefix-free codeword is that $l(x^n)$ satisfies Kraft's inequality: $\sum_{x^n \in \mathcal{X}^n} \exp(-l(x^n)) \leq 1$.

Let $H_n(\theta)$ be the entropy of the parametric source induced by μ_θ as given by

$$H_n(\theta) = E\left[\log \frac{1}{\mu_\theta(x^n)}\right] = \sum_{x^n} \mu_\theta(x^n) \log \frac{1}{\mu_\theta(x^n)}.^{3} \quad (1)$$

The performance of the employed compression is measured in terms of the average code redundancy, which is given by

[2]In this paper, all logarithms and exponentiations are performed at base 2.

[3]In this paper, E denotes the expectation operation using the probability measure μ_θ.

TABLE I
SUMMARY OF THE FREQUENTLY USED NOTATIONS USED THROUGHOUT THIS PAPER

θ	true d-dimensional source parameter vector
Θ	set in which θ lives (which is the d-dimensional simplex of probability vectors)
x^n	new sequence to be sent from S to C
y^m	side information sequence that is available to S
z^m	side information sequence with erased symbols that is available to M
$\mathcal{E}$	fraction of the symbols that have been erased in z^m
n	length of the sequence to be compressed
m	length of the side information sequence
$\mu_\theta(x^n)$	parametric probability distribution over sequence x^n
$H_n(\theta)$	entropy of a n-tuple generated by the parametric source μ_θ
$\overline{R}(n,\Theta)$	the average minimax redundancy of compression of a sequence of length n
$l_S(x^n, y^m)$	the length of the code $c_S(x^n, y^m)$ transmitted on S-C link
$L_S(n,m)$	average codeword length transmitted on S-C link
$l_M(n, z^m)$	the length of the code $c_M(z^m, n)$ transmitted on M-C link
$L_M(n,m)$	average codeword length transmitted on M-C link
$\hat{\theta}(y^m)$	maximum likelihood estimate of θ given y^m
$l_S^{2p}(x^n, y^m)$	the length of the proposed two-part code transmitted on S-C link
$L_S^{2p}(n,m)$	average length of the proposed two-part code transmitted on S-C link
$l_M^{2p}(y^m)$	the length of the proposed two-part code transmitted on S-C link
$L_M^{2p}(n,m)$	average length of the proposed two-part code transmitted on M-C link
κ	ratio of the communication cost in M-C link to that of S-C link
$l^{2p}(x^n, y^m, \kappa)$	total communication cost of sending x^n where y^m is available to helper with link cost ratio κ
$L(\kappa)$	total average communication cost normalized by S-C link cost

$R(l,\theta) = E[l(X^n)] - H_n(\theta)$. Redundancy is the penalty term associated with the universality of the coding scheme. The average minimax redundancy, defined as

$$\overline{R}(n,\Theta) = \min_l \max_\theta R(l,\theta),$$

is a performance measure for universal lossless coding schemes. It is shown in [22], [23] that for a memoryless source with d unknown parameters, we have

$$\overline{R}(n,\Theta) \approx \frac{d}{2}\log n + C_\Theta,^4 \tag{2}$$

where C_Θ is an absolute constant with respect to n that depends on the geometry of the set in which the parameters of the source live. Roughly speaking, (2) states that the cost of universality is linear in the unknown parameters of the parametric model and is logarithmic in the length of the sequence that needs to be compressed. Therefore, the model cost per source symbol is asymptotically $O(\log n/n)$ and uniformly vanishes for all $\theta \in \Theta$. See Appendix A for a more detailed discussion. A summary of the frequently used notations used in this paper is presented in Table I.

B. Related Work

In [9], we studied memory-assisted compression, which refers to universal compression where side information (in the form of previously communicated packets) is available to both the encoder and the decoder. In this case, the memorization and learning from traffic takes place at the network layer because the routers (or the intermediate relays) are observing the packets at the network layer. Therefore, if the client has been in contact with the server for a long time and its physical constraints allow forming a common memory with the server, techniques similar to [9] could be applied to eliminate the redundancy on the hop between the wireless gateway and the client. Furthermore, with sufficiently large side information (4MB or more), there is no need for the use of a helper as the source can compress the packet close to its entropy and transmit to the client.

On the other hand, we assume that the client often lacks memory with the source. This is because the client is not connected to the source as often, and hence, even if it has obtained some packets from the source in the past, they may be outdated to carry information about source contents. Finally, physical constraints on the mobile client may prevent storing previous communication. Hence, due to lack of memory at the client, the memory-assisted compression is not applicable end-to-end; from the source all the way to the client. This work considers the case where the shared memory between the encoder and the decoder is no longer present as we are concerned with redundancy elimination in the last hop between a wireless gateway and a client who is assumed to share no memory with the wireless gateway.

Parallel to this work, opportunistic routing ideas have shown to be very effective in increasing throughput by avoiding the transmission of redundant data chunks destined to different clients sharing part of their path in multi-hop wireless networks using network coding (see ExOR [24] and COPE [25]). The nature of the redundancy that is tackled in the network coding is due to the same (or correlated) contents traversing the same edges in the network in order to reach spatially-separated clients in the network. On the other hand, the redundancy that is addressed in this work is due to the statistical dependencies and duplicates that exist even within a unicast session. Therefore, the two approaches (network coding and compression) are

[4] Throughout this paper, we have used the following asymptotic notations:
- $f(n) = o(g(n))$ iff $|f(n)| \le |g(n)|\epsilon$, $\forall \epsilon$,
- $f(n) = O(g(n))$ iff $|f(n)| \le |g(n)|k$, $\exists k$,
- $f(n) = \omega(g(n))$ iff $g(n) = o(f(n))$,
- $f(n) = \Omega(g(n))$ iff $g(n) = O(f(n))$,
- $f(n) \lesssim g(n)$ iff $f(n) \le g(n) + o(1)$,
- $f(n) \gtrsim g(n)$ iff $f(n) \ge g(n) + o(1)$, and
- $f(n) \approx g(n)$ iff $f(n) = g(n) + o(1)$.

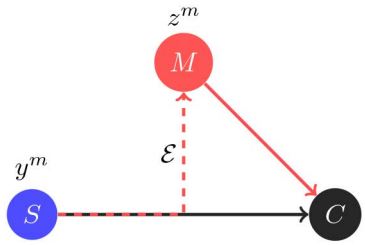

Fig. 2. The abstract illustration of the traffic reduction problem via network compression. The memorized sequence y^m represents the total past data overheard by M from S to the clients.

orthogonal and one can expect to benefit from both when redundant data are being transmitted in a multi-hop wireless network scenario.

Finally, the use of helper nodes in wireless networks is studied in various contexts from femto-cell network architectures [26] to device-to-device collaboration in wireless networks [27], [28]. Our use of helper nodes in the network is inspired by these designs. We also note that the network compression solution developed in [9], [10] is not applicable to the last hop in wireless networks. This is mainly because the broadcast nature of the wireless networks and also the asymmetric cost of transmission in different links that need to be incorporated in the network compression framework while guaranteeing that all packets are recoverable in a strictly lossless manner at the clients.

III. Wireless Network Compression via Memory-Enabled Overhearing Helpers

The idea of wireless network compression via overhearing helpers is to deploy memory-enabled (non-mobile) helpers that are capable of overhearing communication from the wireless gateway to all the mobile clients inside the coverage area of the wireless gateway. The overhearing comes at no extra cost due to the broadcast nature of the wireless communication. Although this can be applied to every cellular or WiFi access network, one realization of such memory-enabled helpers can be in femto-cell network design combined with traditional macro-cell networks, as in [26]. Note that the backbone connectivity of the helpers are not included in the problem setup, first because the learning process of helper nodes is performed only based on the overheard data which is available for free and secondly, solutions that rely on helper connectivity should include provisions to deal with intermittent connectivity and the impact of the extra load imposed on the backbone.

The abstract model of the problem is shown in Fig. 2, for a single helper and a single client. We consider the traffic reduction (compression) over the down-link, where the data are delivered from the wireless gateway S, which is the source in our abstraction, to the mobile client C in unicast sessions. The memory-enabled helper M is assumed to overhear the communication from S to C. The overhearing memory-enabled helper is also assumed to be capable of transmitting information to those mobile nodes in its vicinity.

In our abstraction of the problem, node S may be viewed as an i.i.d. memoryless source (see Section II) that sends independent sequences of length n to the clients in the cell. Note that the source parameters are a priori unknown to the clients requiring the compression procedure to be *universal*. Extension to non-stationary scenario is outside of the scope of this paper and can be found in [18].

Assume that several sequences (packets) have already been destined to some other clients via unicast from S, but due to the broadcast nature of the wireless environment, the helper M also overheard a subset of these sequences. Let y^m denote a sequence of length m, which is formed by the concatenation of all previously sent sequences to the other clients by S. Note that since the source is assumed to be i.i.d., X^n and Y^m are independent of each other given the source parameter vector θ. However, since θ is unknown they are correlated through the information they carry about the unknown source parameter vector. Throughout this paper, we assume that $m = \omega(1)$, i.e., the length of the memorized sequence is sufficiently large for our analysis to hold. It was demonstrated in [9] that a few megabytes of memory is sufficient for the benefits of network compression to be applicable.

Note that one can potentially think of a more sophisticated information structure, in which X^n and Y^m are correlated given the parameter vector θ. This is a viable model where the sequences are noisy observations of the same phenomenon (e.g., readings of two sensors of the temperature of the same room, or two different images of the same object). On the other hand, it is extremely hard to determine whether the packets are related in such fashion a priori. If one does not have prior knowledge on how X^n and Y^m are correlated then the simplest hypothesis is to assume they are independent, which falls back onto the model considered in this paper. We emphasize that in practice one would need to consider the packet generating source to be a mixture of not necessarily i.i.d. models in order to achieve the best compression results. This is beyond the scope of the present paper and is the subject of the study in [18].

The correlation structure between the overheard packets and the new information packet to be sent to client breaks down if the information source is encrypted. In such scenario, no gain would be expected from the overheard traffic as the goal of encryption is to remove all correlations and make the encrypted message appear random. Therefore, the analysis in this paper applies only if either the traffic is unencrypted or the helper and the gateway share the encryption/decryption key. The details of such design is beyond the scope of this paper. Note that the issue with encrypted traffic is not specific to wireless network compression and the same applies to any other redundancy elimination mechanism that uses helper nodes.

Although the link between S and C is lossy due to the wireless channel, we assume a proper feedback for packet retransmission would take care of packet losses on the S-C link. On the other hand, the loss on the S-M link would not be taken care of (as there is no feedback in place). In the absence of erasure caused by overhearing, S and M would share a common side information y^m. We assume that a fraction $\mathcal{E}$ of the symbols in y^m are erased before they reach the helper ($0 \leq \mathcal{E} \leq 1$). Let $e^m \in \{0, 1\}^m$ be a binary sequence that is the indicator of

symbol loss, where $e_i = 1$ would mean that y_i is lost on S-M link and hence is not available to the helper. We also assume that e^m is selected uniformly at random from the $\binom{m}{m\mathcal{E}}$ possible sequences for which $\sum_{i=1}^{m} e_i = m\mathcal{E}$. Let z^m denote the side information sequence that is available to M, which for all $1 \leq i \leq m$ is defined as

$$z_i = \begin{cases} y_i & \text{if } e_i = 0 \\ \varepsilon & \text{otherwise} \end{cases}, \tag{3}$$

where ε denotes an erasure. Further let $\mathcal{Z} \triangleq \mathcal{X} \cup \{\varepsilon\}$ denote the alphabet of the side information sequences at M, and hence, $z^m \in \mathcal{Z}^m$.

We assume an erasure channel where the helper would be able to infer if an erasure has happened. The inference is obtained via the private sequence numbers designed into application layer, sequence numbers (transport layer), time stamps (network layer), and hints from lower layers, such as the PHY. Thus, we assume that the helper knows the sequence e^m. We also assume that S knows the total number of erasures that have occurred in the S-M link, i.e., $\mathcal{E}$ is known to S. This can be done by a single one-time feedback sent from M to S every time that the side information between the two nodes needs to be synchronized. In particular, if the source is "stationary." then the statistics of the source do not change with time. Hence, M can wait until its packet store is filled up and then M can send a one-time feedback to S and identify the packet numbers of the missing packets. This is a very insignificant overhead compared with the size of the packets as the feedback size grows logarithmically with the size of the packet store. However, in reality, the statistics of the packets is not stationary requiring a packet store update with time. The frequency at which such synchronization should take place depends on the nature of the traffic and how quickly the statistics of the source are changing.

After the transient period in which the packet stores at S and M are populated, S wishes to send a new sequence x^n to C. Recall that the traffic (i.e., the packets) destined to different mobile clients from the gateway S are highly correlated as observed in [9]. Therefore, the memory-enabled overhearing helper M can learn about the source model by using the overheard packets from the past communication between the cell tower (or the WiFi access-point) and mobile nodes. This extracted source model can then be used as a side information (if provided to the client) improving the compression performance on the future traffic from the gateway S to any new mobile client C. In other words, the memory-enabled helpers can possibly help to reduce the transmission load of the wireless gateway by transmitting the side-information about the data traffic to the clients using a less costly memory-client M-C link. Note that the compression based redundancy elimination technique misses the opportunity to suppress exact large duplicates. As such, wireless network compression and deduplication based redundancy elimination can be employed together to complement each other in traffic reduction.

As mentioned before, since we wish to reduce the load of the gateway, we have an asymmetric situation where a higher cost is associated with the channel from the source to the client than from the helper to the client. This asymmetry between the channel costs is motivated by real-world cellular networks where a single base-station serves a large number of clients. Hence, if the load of the base-station by each client is reduced, it can potentially serve a larger number of clients. For example, the S-C link from the base-station to the client (and hence the overhearing link S-M) can operate in a frequency different from the M-C link. Whenever the base-station hands off the connection to the mobile client (and the overhearing memory-enabled helper), its frequency slot frees up and a new client can be served. Further, due to a lower communication radius, the frequency slot allocated to the M-C link can be reused within a cell for the link between some other memory-enabled helpers with nearby clients. This architecture together with the proposed network compression offers a novel opportunity for traffic reduction so as to deliver x^n by exploiting the side-information shared between S and M.

IV. Achievable Codeword Length Region

In this section, we will analyze the degree at which we can offload the gateway when the helper transmits a certain number of bits to the client. Let x^n be a packet of length n to be delivered from the source to C. The problem of interest, in its general form, is how would the encoder of S encode x^n knowing that a side information would be transmitted from M to C such that the aggregate communication cost on the link S-C together with the cost of supplemented bits on the link M-C would be minimized. The other important requirement is that x^n should be recovered at the client error-free. Following the notations in Section II, let l_S and l_M denote the (prefix-free) length functions of the codewords transmitted to the client C from the source S and the helper M, respectively. The length functions are also a function of the side information sequence, i.e., $l_S(x^n, y^m)$ and $l_M(n, z^m)$. Recall that the summary of the notations used in this paper can be found in Table I.

Definition 1 (Achievable average codeword length pair): The pair (L_S, L_M) is called an achievable average codeword length pair (in short achievable pair) if there exist codes c_S and c_M with length functions l_S and l_M such that $E[l_S(X^n, Y^m)] = L_S$ and $E[l_M(n, Z^m)] = L_M$ and there exists a decoder d_C (available to the client) such that for all $x^n \in \mathcal{X}^n$, $y^m \in \mathcal{X}^m$, and $z^m \in \mathcal{Z}^m$, we have

$$d_C(c_S(x^n, y^m), c_M(n, z^m)) = x^n.$$

Observe that from Definition 1 it is clear that the decoder only has access to $c_M(n, z^m)$, which is a function of z^m whereas the encoder has encoded x^n using the side information y^m. Despite this discrepancy, the decoder has to decode x^n correctly for any possible values of $x^n \in \mathcal{X}^n$, $y^m \in \mathcal{X}^m$, and $z^m \in \mathcal{Z}^m$. In the rest of this paper, we use $L_S = L_S(n, m)$ and $L_M(n, m)$ to denote the average codeword lengths transmitted by S and M, respectively, where the expectation is performed with respect to the true distribution μ_θ over all realizations of $(x^n, y^m, z^m) \in \mathcal{X}^n \times \mathcal{X}^m \times \mathcal{Z}^m$.

Definition 2 (Achievable average codeword length region): For a given sequence length n and memory size m, the

achievable average codeword length region (in short achievable region) is defined as the union of all achievable average codeword length pairs (L_S, L_M).

Lemma 1: The achievable codeword length region is convex, i.e., if (L_S^1, L_M^1) and (L_S^2, L_M^2) are achievable codeword length pairs then $(\lambda L_S^1 + (1-\lambda)L_S^2, \lambda L_M^1 + (1-\lambda)L_M^2)$ is also an achievable codeword length pair for any $0 < \lambda < 1$.

Proof: Let the code be constructed as follows. Use the pair (L_S^1, L_M^1) for communication with probability λ and the pair (L_S^2, L_M^2) for communication with probability $(1-\lambda)$. Hence, the average codeword length sent from S is going to be $\lambda L_S^1 + (1-\lambda)L_S^2$ as desired. ■

Lemma 2: Let (L_S^1, L_M^1) and (L_S^2, L_M^2) be two achievable pairs such that $(L_S^1, L_M^1 - \epsilon)$ and $(L_S^2, L_M^2 - \epsilon)$ are not achievable for any $\epsilon > 0$. In other words, (L_S^1, L_M^1) and (L_S^2, L_M^2) belong to the boundary of the achievable rate region. Then,

$$\left|L_S^2 - L_S^1\right| \leq \left|L_M^2 - L_M^1\right|. \tag{4}$$

Proof: Note that any bits that are sent by the helper to the client could have also be sent by the wireless gateway as well (since the wireless gateway has access to the same memory and is able to generate the same bits). Therefore, if (L_S, L_M) is achievable then $(L_S + dL_S, L_M - dL_S)$ is also achievable for any $dL_S > 0$. Now assume that (L_S^2, L_M^2) is an achievable pair such that $L_S^2 < L_S^1$. Then, $(L_S^1, L_M^2 + L_S^1 - L_S^2)$ is also achievable. On the other hand, we know that $(L_S^1, L_M^1 - \epsilon)$ is not achievable for any $\epsilon > 0$. Hence,

$$L_M^2 + L_S^1 - L_S^2 \geq L_M^1. \tag{5}$$

The proof goes through similarly for $L_S^2 > L_S^1$. ■

Lemma 2 states that in order to reduce any single bit that needs to be sent on the *S-C* link, we need to send at least one bit on the *M-C* link. In other word, the bits sent by S are at least as useful as the bits sent by M which was intuitively expected. Next, we will state a converse for the achievable codeword length region.

Theorem 3: If $\mathcal{E} < 1$ is fixed, then all pairs (L_S, L_M) in the achievable codeword length region would need to asymptotically satisfy either P1 or P2.[5]

P1. For some $t = \omega(1)$:

$$\begin{cases} L_S \gtrsim H_n(\theta) + \frac{d}{2}\log\left(1 + \frac{n}{dt}\right) \\ L_M \eqsim \frac{d}{2}\log t \end{cases}, \tag{6}$$

P2. For some $t = O(1)$:

$$\begin{cases} L_S \gtrsim H_n(\theta) + \overline{R}(n, \Theta) - \frac{d}{2}\log t \\ L_M \eqsim \frac{d}{2}\log t \end{cases}. \tag{7}$$

See Appendix B for the proof.

Theorem 3 provides a converse bound on what can be hoped to achieve using wireless network compression via overhearing helpers. In other words, it is hopeless to try to do better than the bounds stated in the theorem. We will study the code design to achieve good codewords that perform close to the stated bounds in Section V.

In the case where $\mathcal{E} = 1$, i.e., no side information is available to M, we can state a tighter converse.

Proposition 4: If $\mathcal{E} = 1$, then for all L_M

$$L_S \gtrsim H_n(\theta) + \overline{R}(n, \Theta). \tag{8}$$

Proof: In this case since there is no memory available, everything falls back onto the traditional setup of universal compression and the result is then straightforward. ■

One thing to note here is that we conjecture that the converse that is provided in Theorem 3 is not tight for $\mathcal{E} > 0$.

Corollary 5: If (L_S, L_M) is an achievable pair, then we have

$$L_S \gtrsim H_n(\theta). \tag{9}$$

This is readily deduced from Theorem 3. It is also intuitive as there is no way to drive the average codeword length required to encode a sequence below its entropy.

In this paper, we are interested in evaluating the communication cost. We assume that the total cost function is linear in L_S and L_M, i.e., communicating each bit on the *S-C* and *M-C* links have constant costs that do not vary with time. Extension to the time-varying costs is left as an open future direction. Let κ denote the ratio of the cost of communicating one bit in the *M-C* link to that of the *S-C* link. As described in the introduction, it is expected that $\kappa < 1$, i.e., offloading the gateway by transmitting through the helper is beneficial, which is the main motivation of this paper. This is because S serves several femto-cells but a helper node only serves the clients within a single femto-cell. Let $L(\kappa)$ denote the total communication cost (normalized to the communication cost in the *S-C* link) as given by

$$L(\kappa) = L_S + \kappa L_M. \tag{10}$$

Remark: Thus far, we ignored the cost associated with the client. It is evident that the client would need to receive both sequences and hence an average codeword length of $L_S + L_M$, which is linear in L_S and L_M. Hence, the client-side cost can also be incorporated in the overall total cost by modifying κ.

As a corollary to Theorem 3, we state the following result on the total cost of communication.

Corollary 6: For all $\kappa \geq 0$, we have

$$L(\kappa) \gtrsim H_n(\theta). \tag{11}$$

Note that we have assumed that X^n and Y^m are independent given the parameter vector θ. If Y^m and X^n were further correlated given θ, then one would expect to push the cost further down and asymptotically achieve a cost $H(X^n|Y^m)$ which could be smaller than the entropy of the sequence X^n. See [29] for further discussion on this point. This is where there is opportunity for taking advantage of de-duplication based redundancy elimination techniques to complement wireless network compression by suppressing the big repeated chunks.

Proposition 7: If $\kappa \geq 1$, then the communication cost $L(\kappa)$ is minimized by the pair (L_S, L_M):

$$\begin{cases} L_S \eqsim H_n(\theta) + \overline{R}(n, \Theta) \\ L_M \eqsim 0 \end{cases}. \tag{12}$$

[5] See Footnote 4 for a summary of the asymptotic notations used in this paper.

Proof: For any pair (L'_S, L'_M) with cost $L_1 = L'_S + \kappa L'_M$, according to Lemma 2, $(L'_S + L'_M, 0)$ is also achievable. On the other hand, the cost associated with the latter pair is $L_2 = L'_S + L'_M \leq L_1$ as $\kappa \geq 1$. Now, observe that this falls back to the problem of universal compression for which it is known that the optimal codeword length satisfies the average minimax redundancy in (2), which completes the proof. ■

V. Two-Part Code Design for Matched Memory Case ($\mathcal{E} = 0$)

In this section, we present a code design based on the adaptation of the two-part coding (see [13], [30], [31] and the references therein) for the matched memory case (i.e., $\mathcal{E} = 0$). In this case, y^m is available to both the encoder and the decoder. In other words, $z^m = y^m$. We defer the impact of the mismatched side information to Section VII. The adaptation of two-part codes is natural for this problem since the compressed codeword describing x^n is consisted of two parts that can be separately sent to the end-user, i.e., the client node C; one part from the source S and the other from the memory-enabled helper M. See [13], [30], [31] for a detailed review of the two-part coding scheme.

In this case both M and S share the memory y^m after the memorization phase. We use a two-part statistical compression method for coding. A model of the unknown parametric source μ_θ is created at S and M using the (shared) side information y^m. In the compression of a new packet x^n, the server S would only send (to the client C) the output of the arithmetic encoder which compresses x^n using the model. To complement the compressed sequence sent by S, the memory-enabled helper M forwards (to the mobile client C) the corresponding source model used by the arithmetic encoder. Hence, the mobile client would be able to decode x^n although the client did not have access to the side information (i.e., the source model).

First, let us discuss what is transmitted on the M-C link. The source model is equivalent to an estimate for the unknown source parameter vector θ. Let the maximum likelihood (ML) estimate of the unknown parameter vector from the side information sequence be denoted by $\hat{\theta}(y^m)$. The ML estimate is formally defined as $\hat{\theta}(y^m) = \arg\max_\theta \mu_\theta(y^m)$. Since the ML estimate is a sufficient statistics for the source, we assume that the helper simply encodes a *truncated* ML estimate of the unknown parameter vector, denoted by $\lfloor \hat{\theta}(y^m) \rfloor_{\hat{l}(y^m)}$ where $\hat{l}(y^m)$ denotes the number of bits used in the truncated estimate and is a design parameter to be exploited. Hence, $l_M^{2p}(y^m) = \hat{l}(y^m)$. Note that the same truncated parameter vector $\lfloor \hat{\theta}(y^m) \rfloor_{\hat{l}(y^m)}$ is also available to the encoder. For simplicity of notation, we often use $\lfloor \hat{\theta}(y^m) \rfloor$ to denote the truncated estimate when it is clear from the context that it is a function of y^m.

Then, on the S-C link, the gateway simply transmits the best code associated with the truncated parameter vector to the client. The length of such codeword (using an arithmetic code) is

$$l_S^{2p}(x^n, y^m) = \left\lceil \log \frac{1}{\mu_{\lfloor \hat{\theta}(y^m) \rfloor}(x^n)} \right\rceil + 1.$$

Note that there is an inherent tradeoff between the two parts of the code. By increasing $l_M^{2p}(y^m)$, the truncated ML estimate gets closer to the true parameter θ, and hence, the description length $l_S^{2p}(x^n, y^m)$ of the packet to be compressed becomes smaller. Our goal is to derive the optimal operation point in the tradeoff that minimizes the overall cost.

There are two main differences between the two-part coding scheme for wireless network compression and traditionally used two-part codes. First, the existing two-part codes build the truncated estimate from x^n itself and not y^m. Second, the existing two-part codes equally weigh the two parts of the code whereas in our case, the truncated estimate sent from M needs to be weighed with a factor of κ which will affect the sweet spot in the tradeoff between the two parts of the code. The optimization of the traditional two-part codes is extensively studied in the literature and performance of two-part codes has been characterized (see [13], [22], [23], [30]). In short, one can design two-part codes that are minimax optimal whose average length is equal to the average minimax redundancy given in (2). On the other hand, with a slight compromise in the performance, one can design two-part codes that perform close to optimal but are much simpler to implement. In this paper, we take the latter approach and optimize the code design to achieve the sweet spot associated with the asymmetric cost.

A. Code Design

Let the total cost of delivering a sequence x^n using side information y^m on links with cost ratio κ be denoted by $l^{2p}(x^n, y^m, \kappa)$. We have

$$\begin{aligned} l^{2p}(x^n, y^m, \kappa) &= l_S^{2p}(x^n, y^m) + \kappa l_M^{2p}(y^m) \\ &= \left\lceil \log \frac{1}{\mu_{\lfloor \hat{\theta}(y^m) \rfloor}(x^n)} \right\rceil + 1 + \kappa \hat{l}(y^m) \end{aligned} \quad (13)$$

where κ is the ratio of the cost of transmission on the M-C link to that of the S-C link. Note that the total average cost (of the two-part code) denoted by $L^{2p}(\kappa)$ in Section IV is then given by $L^{2p}(\kappa) = E[l^{2p}(X^n, Y^m, \kappa)]$.

The key to constructing a two-part code achieving the minimum communication cost is to discretize Θ to a countable set of points $\Phi \subset \Theta$ such that the maximum likelihood (ML) estimator restricted to Φ achieves almost the same codelength as the unrestricted ML estimator [30]. The summary of the construction of two-part codes for wireless network compression via helpers is presented in Table II.

Let L_S^{2p} and L_M^{2p} be the average codeword length on the S-C and M-C links achieved by the described two-part coding scheme, respectively. The following theorem determines the communication cost in the case of network compression via overhearing helper. We stress that our construction is for $\mathcal{E} = 0$ where y^m is available at both S and M.

Theorem 8: Given a memory of size m such that $m = \omega(1)$, and for any $0 < \lambda < 1$, we have

$$\begin{cases} L_S^{2p} \lesssim H_n(\theta) + \frac{d}{2} \log\left(1 + \frac{n}{m\lambda}\right) + O(1) \\ L_M^{2p} \lesssim \frac{d}{2} \log m\lambda + O(1) \end{cases}.$$

TABLE II
SUMMARY OF WIRELESS NETWORK COMPRESSION VIA TWO-PART CODES FOR MATCHED MEMORY CASE ($\mathcal{E} = 0$)

Initialization (S and M)
The helper node M overhears the communication between S with past clients and accumulates previous packets. Node S also keeps those packets. Therefore, S and M have access to a shared memory y^m, which they use to obtain a truncated maximum likelihood estimate of the unknown source parameters and build a statistical model for the source.
Operation (S)
For every new packet (sequence) x^n, S uses the statistical model to estimate the probability (likelihood) of the symbols in x^n. Then, the estimates of the likelihoods along with x^n are sent to an arithmetic encoder. The output of the encoder (NOT the estimated source model) is then sent to C.
Operation (M)
Once the helper M finds out that the compressed packet $c(x^n)$ is sent to a client within its coverage, then M encodes and transmits the truncated maximum likelihood estimate of the source parameters to C.
Operation (C)
The client C receives the output of the (arithmetic) encoder from S and the truncated maximum likelihood estimate of the source parameters from M and feeds them to an arithmetic decoder to reconstruct x^n free of error.

The proof is provided in Appendix C.

A close look at Theorem 8 reveals that for sufficiently large m, we can actually achieve the converse bound derived in Theorem 3 up to a constant term by using the described two-part coding.

Proposition 9: If $m = \omega(1)$, then for any $\kappa < 1$, the following is achievable:

$$L(\kappa) \lesssim H_n(\theta) + \frac{d}{2}\kappa \log \frac{(1-\kappa)n}{\kappa} + O(1), \qquad (14)$$

and for $\kappa = 1$, then

$$L(1) \lesssim H_n(\theta) + \overline{R}(n, \Theta) + O(1).$$

Proof: To prove the achievability of $L(\kappa)$, we use the achievability pairs (L_S, L_M) using two-part codes in Theorem 8 and optimize over λ. It turns out that the optimum value is achieved by $\lambda^\star = \frac{1-\kappa}{\kappa}\frac{n}{m}$ for which the cost is obtained in (14). For $\kappa = 1$, it suffices to use the traditional two-part codes on the S-C link and ignore the memory content at M. ■

Corollary 10: If $m = \omega(1)$, we have

$$L(0) \asymp H_n(\theta) + O(1). \qquad (15)$$

Remark: Observe that when m is sufficiently large and $\mathcal{E} = 0$, the number of bits transmitted by the wireless gateway (L_S) is close to the entropy of the sequence which is the information-theoretic lower bound on the average number of bits needed to be sent by S. Further, the aggregate expected communication cost ($L = L_S + \kappa L_M$) is close to the entropy, which is the absolute lower bound on the cost. Hence, we observe that when $\kappa < 1$ for sufficiently large m (asymmetric communication cost), very large memorized sequences can be employed to form a maximum likelihood estimate that is communicated on the M-C link while the total communication cost is still dominated by the bits transmitted from the server to the client on the S-C link, which is close to the entropy. Note that the total number of communicated bits is strictly larger than that of what would have been sent by the gateway if the helper was not present but we have saved in the communication cost as the bits sent from the helper are cheaper.

B. Complexity of Two-Part Codes

We briefly discuss the complexity of the proposed two-part coding strategy. As described earlier, the first stage (i.e., forming the information source model) involves determining the best description of the information source using the memorized sequence of length m. This stage has a complexity linear in size of m and it only needs to be performed once to form the model for the compression of all new packets. The second stage, which is the actual compression of a new packet, involves entropy coding of a packet of size n which has linear complexity in the packet size (i.e., n). With regard to the cost of communication, in this paper we assumed that the cost of transmitting one bit in the S-C channel is unity and that of the M-C channel is κ times as costly (where $\kappa < 1$). In a practical setting, these costs can be assigned through examination of power and bandwidth constraints and our framework can be employed accordingly.

C. Example

To illustrate the trade-offs in Theorem 8, we consider a memoryless source model with alphabet size 256, i.e., each symbol of the source is 1 byte. For small packet lengths, a memoryless source model suffices for modeling of the underlying source in practice so as to avoid overfitting (see [17], [18]). For simplicity, we uniformly discretize the parameter space although better results would be achieved if the discretization points were optimized [30]. The memorized sequence is used to choose the best source parameter from the discretized space that minimizes the description length. The discretization is done such that the sum of L_S and L_M is minimized. The length n packet to be compressed together with the estimated parameter is then fed to a standard arithmetic coder [32].

Fig. 3 shows the ratio (L_M/L_S) achieved using the two-part coding scheme with arithmetic encoding as a function of the sequence length n for a shared memory of 4MB between the encoder and the decoder ($m = 4$MB) for the memoryless source with alphabet size 256. The ratio has been derived for the optimal two-part coding scheme for encoding a sequence of length n (without the asymmetric cost), i.e., the ratio is optimized for $\kappa = 1$. Further, in Fig. 3, L_M is the model cost and L_S is the cost of encoding the sequence using that model.

As an example, for a packet length of 1kB, the optimal size of the parameter estimate L_M is roughly the same size as the compressed packet using the aforementioned arithmetic coder, i.e., $\frac{L_M}{L_S} \approx 1$. To quantify the improvement obtained using wireless network compression via memory-enabled helpers over the traditional compression, we define

$$g = \frac{L(1)}{L(\kappa)},$$

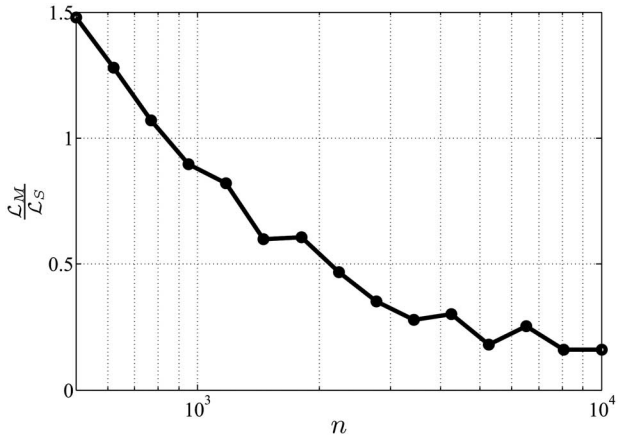

Fig. 3. Ratio of the output size of the helper L_M to the source output size L_S vs. the packet size n for a memoryless source with an alphabet size 256.

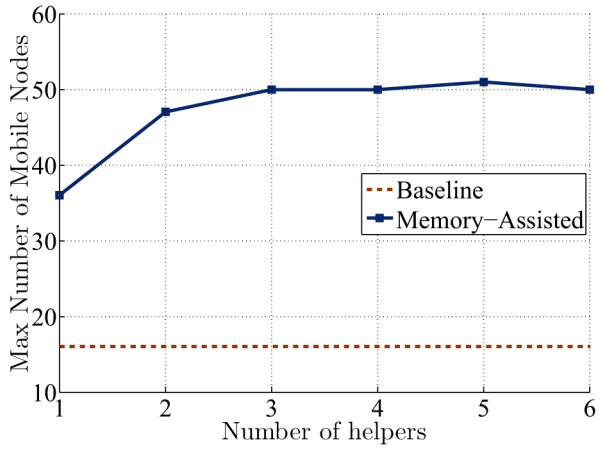

Fig. 4. Maximum number of mobile nodes supported by S vs. the number of helpers in the network. The packet drop rate threshold is fixed at 10% and the traffic generator is CBR over UDP, as in Table III.

TABLE III
SIMULATION PARAMETERS AND VALUES

Parameter	Value
Number of helpers (M)	$0-10$
Comm. Radius of S	250m
Comm. Radius of helper	20m
CBR over UDP rate	64 kbps
UDP baseline packet size	8000 bits
Packet Drop Rate Threshold	10%
FTP file size	20kbit
TCP window size	128

where the numerator denotes the minimum codeword length needed with an asymmetric cost to encode the sequence and the denominator entails asymmetric encoding with cost κ. In light of Proposition 9, the arithmetic encoding scheme achieves the optimal cost up to an additive constant, i.e., $L(1) = H_n(\theta) + \overline{R}(n, \Theta) + O(1)$. For packet sizes of length 1kB where $\frac{L_M}{L_S} \approx 1$, when $\kappa < 0.1$, we observe that $g > 1.5$, as the communication cost of M-C link is much smaller.

VI. SIMULATION

In this section, we simulate an example to evaluate the performance of the proposed wireless network compression via overhearing helpers using ns-2 simulator [33]. The details of the example are provided in Table III. Note that this simulation is just one example to demonstrate the potential benefits of wireless network compression. In reality one should validate the benefits using the specific design constraints of any particular problem. We employed a flat grid topography with a wireless base-station (S) at the origin. Further, multiple memory-enabled helpers (M) are deployed within the coverage of S. The helpers are uniformly distributed in the coverage of S, which is assumed to be a circle of radius 250 m. The communication range of the helpers is 20 m and they are placed such that they are outside of the communication range of each other. All the mobile clients are within the communication range of S, but only a subset is covered by helpers at any given time.

We simulate both Constant Bit Rate (CBR) traffic generator over User Datagram Protocol (UDP) and File Transfer Protocol (FTP) which is running over Transport Control Protocol (TCP). We considered the case where S shares a common memory with each of the helpers and that memory is used for compression of packets sent to mobile nodes within the coverage of the corresponding helper. Further, each mobile client (if covered by a helper) only communicates with a unique helper node. Obviously, if a node is not in the range of any helper, it receives its packets directly from S (via compression without memory). For the baseline simulation scenario, we consider the case where no helper is deployed and all the communication is conducted by S. Hence, packets are compressed individually without using any memory, i.e., end-to-end compression. For FTP simulations, we consider files of size 20kbits for which a memory packet of size 2kbits is sent from helper to the client. The details of simulation parameters are given in Table III.

To examine the effectiveness of the memory-assisted compression, with respect to the baseline scheme, we have considered three performance quantities and evaluated them for both UDP and TCP scenarios. The first quantity is the maximum number of nodes that can be supported for the traffic. To obtain the maximum number of supported nodes in Fig. 4, we have increased the number of mobile nodes in the environment until the packet drop rate exceeds a 10% threshold. We observe that using memory-assisted compression the maximum number of supported nodes increases from 15 to almost 50, as shown in Fig. 4. Since the bottleneck of the network is the output bandwidth of S, we observe from Fig. 4 that adding helpers beyond a certain number does not increase the maximum number of client nodes supported.

In Fig. 5, we have depicted the maximum total throughput/goodput versus the fraction of the nodes covered by helpers. For both of the plots in Fig. 5, the number of nodes is chosen similar to the setup for Fig. 4, that is, the nodes are added to the network (while keeping the helper's coverage constant) until the packet drop rate reaches 10%. The total throughput for UDP traffic and the goodput for TCP traffic is then measured as the sum over all the clients in the network. As expected, as helpers cover more mobile nodes in the network, higher total throughput is achieved. Since the traffic generation for UDP and TCP scenarios is different, we observe different amount of increase

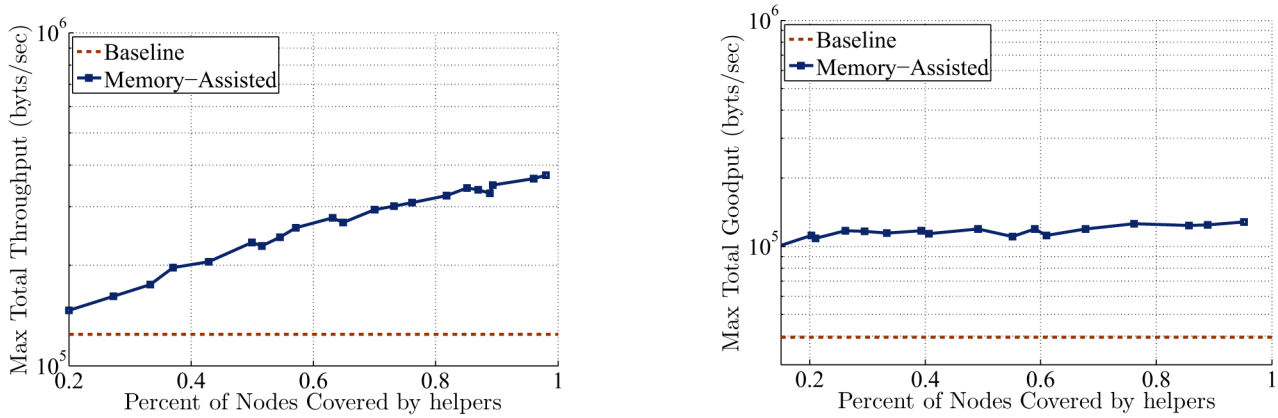

Fig. 5. Maximum total throughput/goodput in the network vs. the fraction of mobile nodes covered by helpers for UDP (left) and TCP (right).

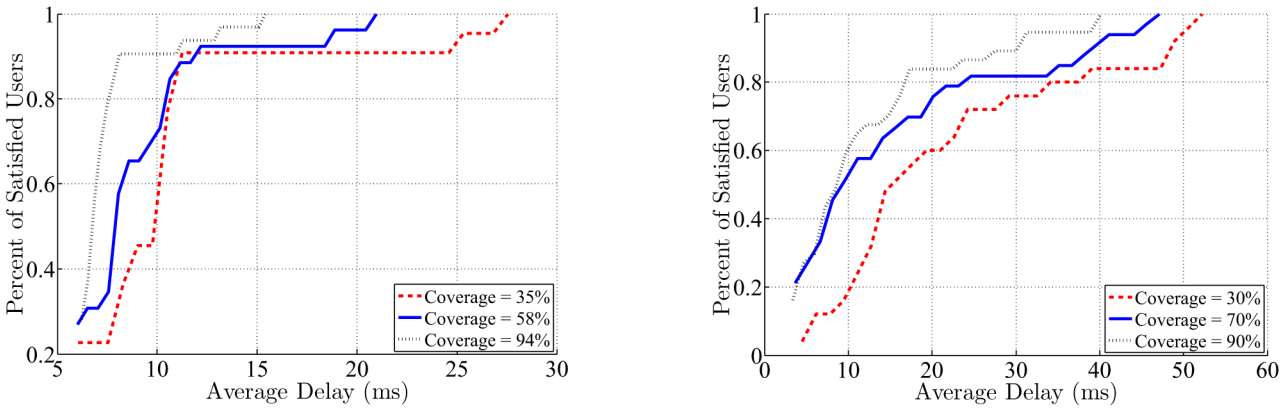

Fig. 6. Fraction of satisfied users in the network vs. maximum allowed average delay of packets for UDP (left) and TCP (right) traffic for different helper coverage percentages.

in the total throughput/goodput but the trend is increasing for both scenarios as we increase the density of the helpers.

The third quantity of interest is the Quality of Service (QoS). To demonstrate the benefit of memory-assisted compression on QoS, we have considered a simulation scenario with fixed number of clients and measured the average delay of packets for each client. The number of helper nodes is changed to obtain the plots for different helper coverage ratios. The number of clients per helper is fixed and we have added more helpers to serve more number of clients. Fig. 6 depicts the fraction of satisfied clients for a given maximum allowable average delay. As we see, users experience less amount of delay as the fraction of nodes covered by helpers increases.

VII. Impact of Packet Loss

Wireless networks are prone to errors. Packets are lost due to errors, the wireless channel, limited buffers and congestion. The effect of the packet loss on redundancy elimination in wireless networks is first studied in [6], where the authors show that high loss rate can be detrimental to the redundancy elimination. However, the data gathered from gateways in North American and European wireless service providers show that the average loss rate in the downlink of UMTS providers is around 3% which is well below the loss rate that causes harm to redundancy elimination. We should also point that the average loss rate on the uplink can be as high as 14%, which if not corrected can render redundancy elimination effectively useless. In [6], authors introduced loss recovery schemes which eliminate the adverse effect of loss on redundancy elimination when the loss rate is high. In particular, the "informed marking" scheme, where each receiver signals the sender whenever it cannot decode a packet due to a missing packet from its cache memory. The receiver sends a control packet of the missing packet and the sender blacklists the corresponding packet in its own cache; in future encoding, any blacklisted packet will be ignored.

Since the focus of this paper is the compression of the downlink traffic and the average loss rate in the downlink is minimal, the coding techniques discussed in previous sections can be applied in real-world scenarios with little modification if informed marking is employed to counter the loss. The cost would be very little communication overhead between S and M.

In high loss rate scenarios, more complicated compression schemes should be developed where one might require the coding to perform under mismatched side information instead of trying to make the memory matched [34]. In the following, we provide a constructive approach which leads to a non-trivial upper bound on the average minimax redundancy for

$m = \omega(1)$. Instead of a joint encoding we choose to encode each of the source parameters separately. Consider a sequence $y^m \in \mathcal{X}^m$ at S with $m_S^{(a)}$ being the number of times $a \in \mathcal{X}$ appears in y^m. Likewise, let $m_M^{(a)}$ be the number of times a appears in $z^m \in \mathcal{Z}^m$. Let $\hat{\theta}_S$ denote the maximum likelihood (ML) estimate of θ at S and $\hat{\theta}_M$ be the ML estimate of θ at M, such that $\hat{\theta}$ is a d-dimensional vector and $\hat{\theta}^{(a)}$ denotes its component that is associated with $a \in \mathcal{X}$. We have

$$\hat{\theta}_S^{(a)} = \frac{m_S^{(a)}}{m},$$
$$\frac{m_S^{(a)} - r}{m} \leq \hat{\theta}_M^{(a)} = \frac{m_M^{(a)}}{m} \leq \frac{m_S^{(a)}}{m}. \tag{16}$$

A strictly lossless compression scheme requires both the encoder and the decoder to use the same parameter estimate for encoding each new symbol. To overcome the mismatch in (16) between $\hat{\theta}_S$ and $\hat{\theta}_M$, we consider the following scheme: both the encoder and the decoder divide the interval $(0, 1)$ in which $\hat{\theta}_S^{(a)}$ and $\hat{\theta}_M^{(a)}$ live into bins of size $\mathcal{E}$. Since $\hat{\theta}_M \leq \hat{\theta}_S$ and $|\hat{\theta}_M - \hat{\theta}_S| < \mathcal{E}$, then $\hat{\theta}_S$ and $\hat{\theta}_M$ are either in the same bin or in two adjacent bins. This discrepancy can be resolved with one extra bit sent by either S to M or vice versa.

Theorem 11: If $m = \omega(1)$, then if $\mathcal{E} = \omega\left(\frac{1}{\sqrt{n}}\right)$, the following pair is achievable

$$\begin{cases} L_S^{2p} \lesssim H_n(\theta) + \frac{d}{2}\log\frac{2n}{\pi e} + d\log\sin^{-1}\mathcal{E} \\ L_M^{2p} \approx \frac{d}{2}\log m + O(1) \end{cases},$$

and if $\mathcal{E} = o\left(\frac{1}{\sqrt{n}}\right)$, we have

$$\begin{cases} L_S^{2p} \lesssim H_n(\theta) + \frac{d}{2}\log\left(1 + \frac{n}{m\lambda}\right) + O(1) \\ L_M^{2p} \lesssim \frac{d}{2}\log m\lambda + O(1) \end{cases}.$$

See Appendix D for the proof.

Also, let us consider the case where $\kappa \approx 0$ to see what would be the maximum achievable benefit from the helper.

Corollary 12: If $m = \omega(1)$, then if $\mathcal{E} = \omega\left(\frac{1}{\sqrt{n}}\right)$, we have

$$L(0) \lesssim H_n(\theta) + \frac{d}{2}\log\frac{2n}{\pi e} + d\log\sin^{-1}\mathcal{E},$$

and if $\mathcal{E} = o\left(\frac{1}{\sqrt{n}}\right)$, we have

$$L(0) \lesssim H_n(\theta) + O(1).$$

It is straightforward to verify that the bound provided by Corollary 12 reduces to the trivial average minimax redundancy bound stated in (2) when $d = 1$ and $\mathcal{E} \to 1$. Further, the bound derived here is strictly larger than (2) for $d > 1$ and $\mathcal{E} \to 1$ due to losing the benefits of vector quantization by encoding each parameter separately [13]. On the other hand, the bound and the achieved rate is still in the form of $H_n(\theta) + \frac{d}{2}\log n + O(1)$. By comparing with Corollary 10, we conclude that when $\mathcal{E} = o\left(\frac{1}{\sqrt{n}}\right)$ the scheme can work as good as if there was no packet loss with possibly constant overhead. In other words, when there is little packet loss, this scheme can automatically achieve the optimal performance without the need to setup a mechanism to match the memory between the source and helper nodes.

VIII. Conclusion

In this paper, we introduced wireless network compression via memory-enabled overhearing helpers, which is a new framework for decreasing the output flow of the wireless gateway in a wireless network by eliminating redundancy from the traffic. The key idea is to deploy a number of memory-enabled helpers in the coverage area of the wireless gateway that are capable of overhearing and memorizing previous communications on the down-link from the wireless gateway to mobile nodes. These helpers then provide side-information to mobile clients using which the wireless gateway may send fewer bits to the client. We adapted two-part codes with the asymmetric cost of communication from the wireless gateway to the client (S-C) versus the memory-enabled helper to the client (M-C), and arrived at optimal two-part codes for the problem. The ns-2 simulation results show that wireless network compression holds a great promise for improving the data transmission efficiency in wireless networks. We observe that network compression increases the maximum throughput significantly while reducing the average delay of packets (hence better QoS) for both UDP and TCP traffic.

Acknowledgements

The authors are grateful to the anonymous reviewers for useful comments that led to a significant improvement of the paper.

Appendix A
Parametric Source Models

In this section, we discuss the parametric source models in more details. Consider a d-dimensional parameter space Θ where the true source parameter θ is chosen from. Further, let the parameter θ be chosen according to a prior density $w(\theta)$ defined over Θ. Define $\underline{R}(n, \Theta)$ as the average maximin redundancy of the parametric source, i.e.,

$$\underline{R}(n, \Theta) = \max_{w(\cdot)} \min_{l} \int_{\Theta} R(l, \theta) w(\theta) d\theta. \tag{17}$$

The average maximin redundancy is associated with the best code under the worst prior on the set of all parameter vectors (i.e., the capacity achieving Jeffreys' prior). Let $\overline{R}(n, \Theta)$ denote the average minimax redundancy, which is defined as

$$\overline{R}(n, \Theta) = \min_{l} \max_{\theta} R(l, \theta). \tag{18}$$

Gallager showed that the average minimax redundancy and the average maximin redundancy (as defined above) are both equal to the capacity of the channel defined between the source parameters and the samples drawn from the source [35]. Let $\mathcal{I}(\theta)$ be the Fisher information matrix associated with the parameter vector θ, i.e.,

$$\mathcal{I}(\theta) \triangleq \left\{ \lim_{n\to\infty} \frac{1}{n\log e} E\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log\left(\frac{1}{\mu_\theta(X^n)}\right)\right]\right\}. \tag{19}$$

Roughly speaking, Fisher information matrix quantifies the amount of information, on average, that each symbol in a sample sequence x^n from the source conveys about the source parameter vector. Let Jeffreys' prior on the parameter vector θ be denoted by

$$p_\Theta(\theta) \triangleq \frac{|\mathcal{J}(\theta)|^{\frac{1}{2}}}{\int_\Theta |\mathcal{J}(\lambda)|^{\frac{1}{2}} d\lambda}. \tag{20}$$

Jeffreys' prior is optimal in the sense that the average minimax redundancy is asymptotically achieved (up to a constant) when the parameter vector θ is assumed to follow Jeffreys' prior [35]–[37]. Jeffreys' prior is particularly interesting because it is also maximin optimal, which corresponds to the worst-case prior for the best compression scheme (called the capacity achieving prior) [35].

We need some regularity conditions to hold for the parametric model so that our results can be derived.

1) The parametric model is smooth, i.e., twice differentiable with respect to θ in the interior of Λ so that the Fisher information matrix can be defined. Further, the limit in (19) exists.
2) The determinant of the Fisher information matrix is finite for all θ in the interior of Λ and the normalization constant in the denominator of (20) is finite.
3) The parametric model has a minimal d-dimensional representation, i.e., $\mathcal{J}(\theta)$ is full-rank. Hence, $\mathcal{J}^{-1}(\theta)$ exists.
4) We require that the central limit theorem holds for the maximum likelihood estimator $\hat{\theta}(x^n)$ of each θ in the interior of Λ so that $(\hat{\theta}(X^n) - \theta)\sqrt{n}$ converges to a normal distribution with zero mean and covariance matrix $\mathcal{J}^{-1}(\theta)$.

Denote by $\overline{R}(w, \theta)$ the expected redundancy of a universal compression scheme with the prior $w(\theta)$ on $\Theta_0 \subset \Theta$. A result related to (2) is the following [30], [36]:

$$\overline{R}(w, \theta) = \frac{d}{2} \log\left(\frac{n}{2\pi e}\right) - \log w(\theta) + \log |\mathcal{J}(\theta)|^{\frac{1}{2}} + o(1), \tag{21}$$

where the convergence is uniform in $\theta \in \Theta_0$. Accordingly, Jeffreys' prior on Θ_0 as defined in (20) (i.e., $p_{\Theta_0}(\theta)$,) is maximin optimal. Note that Jeffreys' prior is also minimax optimal because it makes the above risk expression constant [12].

Finally, another important relationship that we use in this paper is the following result due to Gallager [35] which shows that if μ_θ is a measurable function of θ, then

$$\overline{R}(n, \Theta) = \sup_{w(\theta)} I(X^n; \theta), \tag{22}$$

where $I(X^n; \theta)$ is the mutual information between X^n and θ, and $w(\theta)$ is the prior distribution on θ.

The two-part coding with memory y^m is comprised of three steps. First, the ML estimate of θ is obtained from the memorized sequence; this estimate is denoted by $\hat{\theta}(y^m)$. In the second step, to find a codeword describing the ML estimate the space Θ is split into a set of regions $\mathbf{R}$; the center point of each region $R \subset \mathbf{R}$, denoted by ϕ_R, is used to discretize Θ. Let

$$\Phi = \bigcup_R \{\phi_R\}$$

be the discretized space and denote the corresponding ML estimate in Φ closest to $\hat{\theta}(y^m) \in R$ by $\hat{\phi}_R(y^m)$.[6] Let w_{ϕ_R} denote the probability density corresponding to $w(\theta)$ in the discretized space Φ. We have

$$w_{\phi_R} = \int_R w(\theta)\, \mathrm{d}\theta.$$

Finally, a sequence x^n is compressed using a Shannon code (which is the optimal code when the source parameter vector is known) with parameter vector $\hat{\phi}(y^m)$. The description length of x^n using the Shannon code is given by $-\log P_{\hat{\phi}(y^m)}(x^n)$. The description length of the parameter $\hat{\phi}(y^m)$ is $-\log w_{\hat{\phi}(y^m)}$. We note that the Shannon code is sent by S whilst the parameter is transmitted from M which is less costly by a factor $\kappa < 1$. Henceforth, the communication cost of transmitting x^n from source to client with memorized sequence y^m available to the memory-enabled helper can be written as

$$l(x^n, y^m) = \log \frac{1}{P_{\hat{\phi}(y^m)}(x^n)} + \kappa \log \frac{1}{w_{\hat{\phi}(y^m)}}. \tag{23}$$

By adding and subtracting a $\log(1/\mu_\theta(x^n))$ term (which is the length of the Shannon code using the true source parameter) and also a $\log(1/P_{\hat{\theta}(y^m)}(x^n))$ term (which is the length of the Shannon code using the maximum likelihood parameter after y^m is observed) from (23) and then taking expectation, we can write the expected communication cost as in (29) for the purpose of the proof of Theorem 8.

Appendix B
Proof of Theorem 3

The proof of Theorem 3 is divided into two parts that are presented in the sequel.

Fort the first part, consider $t = \omega(1)$. Then, let $L_M \approx \frac{d}{2} \log t$ be the number of bits sent by M to C. Also notice that the ML estimate $\hat{\theta}$ is a sufficient statistic for the source parameter vector in this case (see [18]), which converges in distribution to a jointly Gaussian random vector with mean θ (See Appendix A). Hence, we only need to encode $\hat{\theta}$. On the other hand, also observe that the rate distortion function of a random variable gives us the minimum number of bits that are needed to encode the random variable such that it could be recovered subject to a given distortion level [19]. Since $\hat{\theta}$ is a Gaussian random vector, the rate distortion function is logarithmic in the number of bits that is available to describe. Note that both the covariance matrix of $\hat{\theta}$ and the redundancy (which induces the distortion measure) scale with $\mathcal{J}(\theta)^{-1}$. Therefore, we can think about the best code as having sent a memory of td previous symbols, where d appears because of the vector quantization gain. According to Theorem 2 of [29], in this case $\frac{d}{2} \log\left(1 + \frac{n}{2t}\right)$ is the minimum number of bits that need to be communicated to the client on top of entropy such that the codeword can be uniquely decoded. Putting all these pieces together proves the first part.

Considering the second part of the Theorem for $\frac{d}{2} \log t = O(1)$, we can invoke Lemma 2 to directly obtain the desired result.

[6] The subscript R is dropped when it is clear from the context.

APPENDIX C
PROOF OF THEOREM 8

Before we proceed with the proof, we need to state a lemma that is going to be used in the proof of the theorem. The lemma is a generalization of Theorem 2 of [29].

Lemma 13: If $m = \omega(1)$ and $\hat{l}(y^m) = \frac{d}{2}\log m\lambda + O(1)$, then

$$E\left[\log \frac{\mu_\theta(X^n)}{\mu_{\lfloor\hat{\theta}(Y^m)\rfloor}(X^n)}\right] \lesssim \frac{d}{2}\log\left(1+\frac{n}{m\lambda}\right) + O\left(\frac{1}{m^{\frac{3}{2}}}\right).$$

Proof: Note that

$$E\left[\log \frac{\mu_\theta(X^n)}{\mu_{\lfloor\hat{\theta}(Y^m)\rfloor}(X^n)}\right] = E\left[\log \frac{\mu_\theta(X^n)}{\mu_{\hat{\theta}(T^{m\lambda})}(X^n)}\right] + E\left[\log \frac{\mu_{\hat{\theta}(T^{m\lambda})}(X^n)}{\mu_{\lfloor\hat{\theta}(Y^m)\rfloor}(X^n)}\right], \quad (24)$$

where $T^{m\lambda}$ is a sequence of length $m\lambda$ generated independently from X^n by the same source μ_θ. The first term on the right hand side is studied in Theorem 2 of [29] and is known to be

$$E\left[\log \frac{\mu_\theta(X^n)}{\mu_{\hat{\theta}(T^{m\lambda})}(X^n)}\right] = \frac{d}{2}\log\left(1+\frac{n}{m\lambda}\right). \quad (25)$$

We finish the proof of the lemma by showing that the second term in (24) is $o(1)$. This intuitively makes sense because $\lfloor\hat{\theta}(Y^m)\rfloor$ contains about $\frac{1}{2}\log m\lambda$ *accurate* bits about each of the unknown source parameters, which is similar to that of $\hat{\theta}(T^{m\lambda})$. To pursue a formal proof, consider the Taylor's expansion of the term $-\log\mu_\theta(X^n)$ around the ML estimate obtained from y^m (which exists and converges to the true parameter because of our assumptions on the parametric source model discussed in Appendix A). We have

$$\begin{aligned} E\left[-\log\mu_\theta(X^n)\right] = {} & E\left[-\log\mu_{\hat{\theta}(Y^m)}(X^n)\right] \\ & + E\left[\left(\nabla\log\frac{1}{\mu_\theta(X^n)}\right)(\theta-\hat{\theta}(Y^m))\right] \\ & + \frac{n}{2}E\left[(\theta-\hat{\theta}(Y^m))^{\mathrm{T}}\mathcal{I}(\hat{\theta})(\theta-\hat{\theta}(Y^m))\right] \\ & + O\left(\frac{1}{m^{\frac{3}{2}}}\right), \end{aligned}$$

where $\mathcal{I}(\hat{\theta})$ is the expected Fisher information matrix evaluated at $\hat{\theta}$, defined as

$$\mathcal{I}_{ij}(\hat{\theta}) = E\left[-\frac{\partial^2}{\partial\theta_i\partial\theta_j}\mu_\theta(X^n)\right]_{\theta=\hat{\theta}}. \quad (26)$$

The second term in the Taylor's expansion is zero as the ML is the maximizer of the likelihood function. For probability densities from the exponential family, the Fisher information matrix is inversely proportional to the covariance matrix, i.e., $E_\theta\left[(\theta-\hat{\theta}(Y^m))^{\mathrm{T}}(\theta-\hat{\theta}(Y^m))\right] = \frac{1}{m}\mathcal{I}^{-1}$. Now, the catch is to realize that when we replace m with $m\lambda$, the error term becomes $\frac{1}{m\lambda}\mathcal{I}^{-1}$. On the other hand, when we truncate the estimate at $\frac{d}{2}\log m\lambda + O(1)$ bits, it means that we the estimate for each parameter at $\frac{1}{2}\log m\lambda + O(1)$ bits. This can generate an extra square error that is at most $\frac{1}{m\lambda}\mathcal{I}^{-1}$ and hence, the total square error is upper bounded by $\left(\frac{1}{m}+\frac{1}{m\lambda}\right)\mathcal{I}^{-1}$. Now, since $m = \omega(1)$, the difference between the two uniformly converges to zero completing the proof. ■

Now, we are ready to present the proof of Theorem 8.

Proof of Theorem 8: Fix $\hat{l}(y^m) = \frac{d}{2}\log m\lambda + O(1)$, which is only a function of m and not the particular y^m to be compressed. Hence,

$$L_M^{2p} = E[\hat{l}(Y^m)] = \frac{d}{2}\log m\lambda + O(1).$$

It suffices to show that the two-part coding scheme described in this paper achieves the desired bound. We have

$$\begin{aligned} l_S^{2p}(x^n, y^m) &= \left\lceil \log\frac{1}{\mu_{\lfloor\hat{\theta}(y^m)\rfloor}(x^n)}\right\rceil + 1 \\ &\le \log\frac{1}{\mu_{\lfloor\hat{\theta}(y^m)\rfloor}(x^n)} + 2, \end{aligned}$$

and hence

$$\begin{aligned} L_S^{2p} &= E[l_S^{2p}(X^n, Y^m)] && (27)\\ &\le E\left[\log\frac{1}{\mu_{\lfloor\hat{\theta}(Y^m)\rfloor}(X^n)}\right] + 2 && (28)\\ &= E\left[\log\frac{1}{\mu_\theta(X^n)}\right] + E\left[\log\frac{\mu_\theta(X^n)}{\mu_{\lfloor\hat{\theta}(Y^m)\rfloor}(X^n)}\right] + 2. && (29) \end{aligned}$$

The first term in (29) is by definition the entropy, i.e., $H_n(\theta)$. The second term is bounded by Lemma 13, which results in the desired result. ■

APPENDIX D
PROOF OF THEOREM 11

Consider a case where we form bins for each of the source parameters. Let $\Theta_i = ((i-1)\mathcal{E}, i\mathcal{E})$ be the i-th bin. If $\mathcal{E} = \omega(\frac{1}{n})$, then each bin will contain $\omega(1)$ estimate points for the two-part coding, and hence, according to (21), the redundancy of a compression scheme, with the side information that the source parameter is chosen from Θ_i, can be obtained as

$$\overline{R}(n, \Theta_i) = \frac{1}{2}\log\left(\frac{n}{2\pi e}\right) + \log\int_{\Theta_i}|\mathcal{I}(\theta)|^{\frac{1}{2}}d\theta + o(1),$$

where we have $\mathcal{I}^{-1}(\theta) = \theta(1-\theta)$ for a binary memoryless source parameter. On the other hand, it is evident that the redundancy is maximized if the parameter vector lives in the endpoint bins. Hence,

$$\int_{\Theta_i}|\mathcal{I}(\theta)|^{\frac{1}{2}}d\theta \le \frac{2}{\pi}\sin^{-1}\mathcal{E}. \quad (30)$$

Putting these together completes the proof of the first part.

Now, considering the case where $\mathcal{E} = o(\frac{1}{\sqrt{n}})$, then size of the bin Θ_i is also $o(\frac{1}{\sqrt{n}})$. Let $\theta^\star \in \Theta_i$, then,

$$\begin{aligned}\overline{R}(n, \Theta_i) &= E[\log \mu_\theta(X^n) - \log \mu_{\theta^\star}(X^n)] \\ &= n D(\mu_\theta || \mu_{\theta^\star}) \\ &\overset{(i)}{=} \frac{n}{2}(\theta - \theta^\star)^2 \mathcal{I}(\theta) + o(1) \\ &\overset{(ii)}{=} o(1), \end{aligned} \tag{31}$$

where $D(\cdot||\cdot)$ is the KL divergence between two probability measures. In (31), equality (i) follows from the second order approximation of the KL divergence term and (ii) follows from the fact that $(\theta - \theta^\star)^2 < \frac{1}{n}$.

REFERENCES

[1] N. T. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic," in *Proc. SIGCOMM*, 2000, vol. 30, pp. 87–95.

[2] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet caches on routers: The implications of universal redundant traffic elimination," in *Proc. SIGCOMM*, 2008, vol. 38, pp. 219–230.

[3] Z. Zhuang, C.-L. Tsao, and R. Sivakumar, "Curing the amnesia: Network memory for the internet," Tech. Rep., 2009 [Online]. Available: http://www.ece.gatech.edu/research/GNAN/archive/tr-nm.pdf

[4] A. Anand, V. Sekar, and A. Akella, "SmartRE: An architecture for coordinated network-wide redundancy elimination," in *Proc. SIGCOMM*, 2009, vol. 39, pp. 87–98.

[5] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: Findings and implications," in *SIGMETRICS'09: Proc. 11th Int. Joint Conf. Meas. Model. Comput. Syst.*, 2009, pp. 37–48.

[6] C. Lumezanu, K. Guo, N. Spring, and B. Bhattacharjee, "The effect of packet loss on redundancy elimination in cellular wireless networks," in *Proc. Internet Meas. Conf.*, 2010, pp. 294–300.

[7] S. Hsiang-Shen, A. Gember, A. Anand, and A. Akella, "Refactoring content overhearing to improve wireless performance," in *Proc. MobiCom*, Las Vegas, NV, USA, 2011.

[8] S. Sanadhya, R. Sivakumar, K.-H. Kim, P. Congdon, S. Lakshmanan, and J. Singh, "Asymmetric caching: Improved deduplication for mobile devices," in *Proc. ACM MOBICOM Conf.*, 2012, pp. 161–172.

[9] A. Beirami, M. Sardari, and F. Fekri, "Packet-level network compression: Realization and scaling of the network-wide benefits," *IEEE/ACM Trans. Netw.*, pp. 1–17, May 2015.

[10] M. Sardari, A. Beirami, and F. Fekri, "Memory-assisted universal compression of network flows," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 91–99.

[11] M. Sardari, A. Beirami, and F. Fekri, "On the network-wide gain of memory-assisted source coding," in *Proc. IEEE Inf. Theory Workshop*, Oct. 2011, pp. 476–480.

[12] N. Merhav and M. Feder, "A strong version of the redundancy-capacity theorem of universal coding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 714–722, May 1995.

[13] A. Beirami and F. Fekri, "Results on the redundancy of universal compression for finite-length sequences," in *Proc. IEEE Int. Symp. Info. Theory*, Jul. 31/Aug. 5, 2011, pp. 1504–1508.

[14] N. Krishnan and D. Baron, "A universal parallel two-pass MDL context tree compression algorithm," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 1–8, Jun. 2015.

[15] R. Lenhardt and J. Alakuijala, "Gipfeli—High speed compression algorithm," in *Proc. Data Compress. Conf.*, 2012, pp. 109–118.

[16] M. Sardari, A. Beirami, and F. Fekri, "Wireless network compression: Code design and trade offs," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, 2013, pp. 1–8.

[17] M. Sardari, A. Beirami, J. Zou, and F. Fekri, "Content-aware network data compression using joint memorization and clustering," in *Proc. IEEE Conf. Comput. Netw.*, Apr. 2013, pp. 255–259.

[18] A. Beirami, L. Huang, M. Sardari, and F. Fekri, "Universal compression of a mixture of parametric sources with side information," arXiv:1411.7607, 2014.

[19] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.

[20] A. R. Barron and T. M. Cover, "Minimum complexity density estimation," *IEEE Trans. Inf. Theory*, vol. 37, no. 4, pp. 1034–1054, Jul. 1991.

[21] J. Rissanen, "Strong optimality of the normalized ML models as universal codes and information in data," *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 1712–1717, Jul. 2001.

[22] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2743–2760, Oct. 1998.

[23] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 40–47, Jan. 1996.

[24] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2004, vol. 34, pp. 69–74.

[25] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2006, vol. 36, pp. 243–254.

[26] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: A survey," *IEEE Comm. Mag.*, vol. 46, no. 9, pp. 59–67, Sep. 2008.

[27] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[28] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 1107–1115.

[29] A. Beirami and F. Fekri, "On lossless universal compression of distributed identical sources," in *Proc. IEEE Int. Symp. Inf. Theory*, 2012, pp. 561–565.

[30] P. D. Grunwald, *The Minimum Description Length Principle*. Cambridge, MA, USA: MIT Press, 2007.

[31] D. Baron, "Fast parallel algorithms for universal lossless source coding," Ph.D. dissertation, Dept. ECE, Univ. Illinois at Urbana-Champaign, Champaign, IL, USA, 2003.

[32] G. G. Langdon Jr., "An introduction to arithmetic coding," *IBM J. Res. Develop.*, vol. 28, no. 2, pp. 135–149, Mar. 1984.

[33] Lawrence Berkeley National Laboratory. (1997). *The Network Simulator NS-2* [Online]. Available: http://www.isi.edu/nsnam/ns/

[34] M. Sardari, A. Beirami, and F. Fekri, "Mismatched side information in wireless network compression via overhearing helpers," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 2222–2226.

[35] R. G. Gallager, "Source coding with side information and universal coding," Sep. 1979 [Online]. Available: http://web.mit.edu/gallager/www/papers/paper5.pdf, to be published.

[36] B. Clarke and A. Barron, "Information-theoretic asymptotics of Bayes methods," *IEEE Trans. Inf. Theory*, vol. 36, no. 3, pp. 453–471, May 1990.

[37] K. Atteson, "The asymptotic redundancy of Bayes rules for Markov chains," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2104–2109, Sep. 1999.

Ahmad Beirami (S'07) received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2007, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2011 and 2014, respectively. Currently, he is a Postdoctoral Associate jointly affiliated with the Information Initiative at Duke (iiD) and the Research Laboratory of Electronics (RLE), MIT. His research interests include information theory, cyber security, machine learning, statistics, and networks. He is the coauthor of a paper that received the Best Student Paper nomination in IEEE Midwest Symposium on Circuits and Systems in 2008. His Ph.D. work received the Center for Signal and Information Processing Outstanding Research Award in 2014, the 2013–2014 School of ECE Graduate Research Excellence Award, and the 2015 Sigma Xi Best Ph.D. Thesis Award, all from Georgia Institute of Technology.

Mohsen Sardari received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2007, and the M.S.E.C.E. and Ph.D. degrees from the School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta, GA, USA, in 2010 and 2013, respectively. He is currently a Data Scientist with Electronic Arts, Inc., Redwood City, CA, USA. His research interests include information theory, signal processing, large-scale data analytics, and machine learning.

Faramarz Fekri (S'91–M'00–SM'03) received the Ph.D. degree from Georgia Institute of Technology, Atlanta, GA, USA, in 2000. Since 2000, he has been with the Faculty of the School of Electrical and Computer Engineering, Georgia Institute of Technology, where he currently holds a Professor position. His research interests include communications and signal processing, in particular, source and channel coding, information theory in biology, statistical inference in large data, information processing for wireless and sensor networks, and communication security. He serves on the technical program committees of several IEEE conferences. He is an Associate Editor of the IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL, AND MULTI-SCALE COMMUNICATIONS. In the past, he served on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS, and the *Elsevier Journal on PHYCOM*. He received the National Science Foundation CAREER Award in 2001, Southern Center for Electrical Engineering Education (SCEEE) Young Faculty Development Award in 2003, and Outstanding Young Faculty Award of the School of ECE in 2006.