# SPIRO: Turning Elephants into Mice with Efficient RF Transport

Eugene Chai & Kang G. Shin
The University of Michigan
{zontar, kgshin}@umich.edu

Sung-Ju Lee
KAIST
sjlee@cs.kaist.ac.kr

Jeongkeun Lee
HP Labs
jklee@hp.com

Raul Etkin
Samsung Electronics America
raul.etkin@ieee.org

***Abstract*—Cloud-RANs (Radio Access Networks) assume the existence of a high-capacity, low-delay/latency fronthaul to support cooperative transmission schemes such as CoMP (Coordinated Multi-Point) and coordinated beamforming. However, building such hierarchical wired fronthauls is challenging as the typical I/Q data stream is non-elastic — I/Q data over the wired fronthaul has little tolerance for delay jitters and zero tolerance for losses. Any distortion to the I/Q data stream will make the resulting wireless transmission completely unintelligible. We propose SPIRO, a mechanism that efficiently transports RF signals over a wired fronthaul network. The primary goal of SPIRO is to make I/Q data streams elastic and resilient to unexpected network condition changes. This is accomplished through a novel combination of compression and data prioritization of I/Q data on the wired fronthaul. For a given wireless throughput, SPIRO can reduce the bandwidth demand of the fronthaul data stream by up to 50% without any noticeable degradation in the wireless reception quality. Further bandwidth reduction via compression and frame losses only have a limited impact on the wireless throughput.**

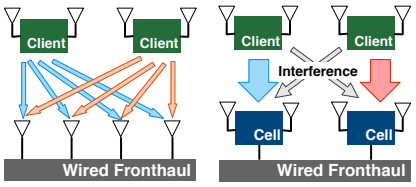
## I. Introduction

The Cloud Radio Access Network (C-RAN) design transitions cellular networks from the current centralized design to a distributed architecture that supports new network technologies such as inter-cell interference alignment and cancellation and network MIMO [1]. A typical C-RAN includes multiple baseband units (BBUs) co-located in a datacenter, multiple remote radio units (RRUs) distributed across multiple cell sites, and a high-speed fronthaul network that transports digital I/Q data between BBUs and RRUs. This high-speed network poses the greatest challenge to successfully constructing C-RANs. In current cellular networks, the BBU and RRU components already exist — existing macro basestations consist of a BBU and an RRU that are co-located within each cell. However, the difficulties in building a massive fronthaul network that can transport large amounts of latency- and delay-sensitive I/Q data reliably between RRUs and BBUs has so far prevented the deployment of any C-RANs.

### A. Why Is It Difficult to Build a C-RAN Fronthaul?

Building a modern fronthaul that can accomodate a large C-RAN is challenging due to the high bandwidth demanded by the I/Q data stream, and its inherent bandwidth inelasticity.

*1) Digital I/Q Data Is Not Elastic: Elasticity* refers to the ability for the bandwidth demanded by a data stream to be dynamically adjusted to match the available bandwidth on the network. Networks such as enterprise wired LANs rely on this property to effectively support a large number of users — data packets can be dropped with little consequence if the network is congested. This is made possible by the fact that TCP/UDP is the dominant protocol on wired LANs, and the Internet at large. Each TCP stream uses the quick start, exponential backoff and ARQ algorithms to adjust its effective bandwidth to the maximum that can be supported by the network, in the presence of competing flows.




**(a)** CoMP **(b)** Non-CoMP

**Fig. 1:** Uplink transmission in CoMP and non-CoMP networks.

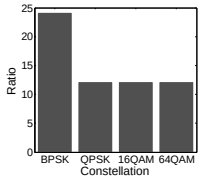



**Fig. 2:** Ratio of CoMP to non-CoMP bandwidth.

Unfortunately, digital I/Q data from RRUs and BBUs are not elastic. For a given wireless channel spectral bandwidth (e.g., a 20MHz LTE channel), the fronthaul bandwidth needed for I/Q data is fixed and constant. Furthermore, this I/Q data stream needs to be delivered over the fronthaul under strict latency and delay constraints. As an example, a 20MHz wireless transmission requires the RRU to transmit 20 million I/Q data samples a second, or one sample every 50 nanoseconds. The RRU has zero-tolerance for I/Q data that arrives slower than this rate. If the fronthaul bandwidth is abruptly reduced or I/Q data is lost due to congestion, the resulting wireless transmission will be unintelligible at the receiver.

*2) Cooperation Increases Fronthaul Demands:* One of the key benefits of C-RANs is the capability for coordination between RRUs, such as Coordinated Multipoint (CoMP) transmissions. However, such coordination increases the demands on the fronthaul network.

CoMP networks achieve greater wireless capacity, at the cost of greater complexity due to cooperative demodulation of sampled RF signals. A CoMP network, shown in Fig. 1a, can utilize four concurrent spatial streams with full coordination between all antennas. On the other hand, a non-CoMP network (Fig. 1b) with the same number of transmit and receive antennas, can only use two spatial streams, one per client,

for data transmission. The remaining stream from each client is needed for interference nullification [2].

However, the fronthaul bandwidth required by CoMP is significantly greater than the non-CoMP network. The number of bits generated by the four CoMP antennas that is sent to a centralized DSP server can be expressed as

$$N_{\text{CoMP}} = \frac{2N_{\text{ant}}N_bR}{\log_2 N_{\text{const}}} \quad (1)$$

where $N_{\text{ant}}$ is the number of receive CoMP antennas, $N_b$ is the number of bits transmitted by each client, $R$ is the number of bits used by the Analog-to-Digital (ADC) quantizer, and $N_{\text{const}}$ is the modulation constellation size. The factor of 2 is needed as we transmit both the I and Q samples. We ignore additional bits that may be received due to oversampling, channel probing and synchronization overheads as they can be trivially removed by the AP before transmission over the fronthaul.

The non-CoMP network with the same number of transmit and receive antennas but without cooperative demodulation, requires a maximum of $2N_b$ bits on the fronthaul network to represent the same transmission by the two clients. Fig. 2 shows the ratio of the fronthaul bandwidth demands of CoMP to that without cooperative demodulation. With BPSK, CoMP incurs $24\times$ the enterprise traffic bandwidth while this ratio falls to $12\times$ at higher modulation rates.

*3) Hierarchical Fronthaul Networks:* A large C-RAN with numerous BBUs and RRUs cannot rely on a single hop network between the RRUs and the BBUs (i.e., a completely-connected graph). Instead, a hierarchical network must be used. This is a challenge also seen in datacenter networks. Multi-level routing architectures [3] are used there as well to route data packets between processors in different racks or domains.

Unfortunately, as is the case with datacenter networks, the combination of hierarchical networks and data streams with variable bandwidth demands result in spurious network congestion. This is true even in a well provisioned network. The typical response to such congestion is packet drops. However, this approach will destroy the decodability of wireless transmissions in C-RANs.

*4) Over-Provisioning the Fronthaul Is Not Practical:* It is well known that the demand for wireless cellular bandwidth varies over time and space [4]. For example, peak weekday demands occur during regular office hours within the commercial districts. The ratio between the peak to off-peak bandwidth demands can be larger than 400% [4]. C-RANs take advantage of this variability by adjusting the wireless bandwidth to match the throughput demand. Hence, the fronthaul bandwidth demands will vary significantly.

While brute-force over-provisioning can be used to deal with traffic variability, this results in very low network utilization (e.g. during off-peak periods). Realistically, the fronthaul capacity will not always satisfy the demands from the BBUs and RRUs. Unfortunately, current digital I/Q data streams are not elastic and cannot adjust their bandwidth in response to wired network conditions.

### B. Our Objective: An Efficient C-RAN Fronthaul

Current I/Q data streams cannot be easily transmitted over a C-RAN fronthaul as they are modeled after the data flow in typical RF processing chains. In a standard hardware baseband processor, I/Q samples are processed systematically by the different DSP blocks (e.g., FFT, de/modulator, equalizer) under the control of a global system clock. The regularity enforced by the hardware system clock removes the need for any data elasticity.

Our objective is to design and evaluate a novel mechanism for *elastic* digital I/Q data streams in a coordinated C-RAN. Specifically, our contributions are:

***(a) Cooperative compression with little-to-none wireless capacity reduction.*** We demonstrate that by cooperatively compressing RF signals from coordinated RRUs, we reduce overall fronthaul bandwidth demands *without any loss of wireless capacity*. This result is particularly surprising and important since at the PHY layer, a critically sampled (i.e., not oversampled) OFDM cellular signal is not sparse and thus, not losslessly compressible. Hence, typical approaches such as sub-Nyquist sampling [5] and compressed sensing [6] cannot be used to reduce the RF bandwidth.

***(b) Loss-resilient PHY transport.*** SPIRO employs a loss-resilient PHY transport protocol that allows fronthaul switches to rapidly and randomly discard I/Q samples in the event of wired congestion with minimal impact on the wireless capacity. This is in stark contrast to typical Software Defined Radios (SDR) DSP operations where the loss of even a small number of I/Q samples due to frame drops (as seen in USRP and WARP) can result in the loss of the entire wireless data frame.

***(c) Real-world evaluation on an SDR testbed.*** We implement and evaluate our bandwidth reduction and PHY transport on a large SDR testbed of 16 WARP devices.

We discuss background in §II and describe the design of SPIRO in §III and its algorithms in §IV. We evaluate our design in §V. We discuss related work in §VI and conclude in §VII.

### C. Target Deployment Scenario

SPIRO is designed for indoor commercial and enterprise networks, such as office buildings, stadiums, and shopping centers. Such environments are the focus of current commercial C-RAN/DAS deployments as they are centrally managed environments with high user densities. Our experimental setup represents one such indoor environment. While the SPIRO design is also applicable to metropolitan-scale networks, we leave such extensions to future work.

## II. BACKGROUND

### A. Network Model

We consider a centralized CoMP/network-MIMO architecture, as shown in Fig. 3. The PHY/MAC protocol is run from a BBU pool, and the generated I/Q samples are sent to the RRUs for transmission. This architecture parallels the typical C-RAN proposal for cellular networks, where BBUs on the backend handle protocol processing while RRUs directly attached to

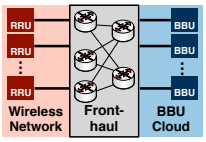



**Fig. 3:** CoMP architecture used by SPIRO.

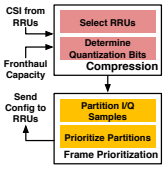



**Fig. 4:** SPIRO-FH controller on the wired fronthaul network.

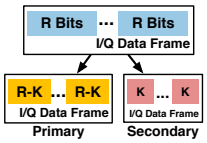



**Fig. 5:** A single frame is split into two frames carrying $R-K$ and $K$-bit samples.

the antennas are connected to the BBUs via long Common Public Radio Interface (CPRI) links.

### B. Uplink vs. Downlink Traffic

In many real-world networks, the downlink traffic is typically larger than the uplink traffic. However, the reverse is true when considering I/Q traffic in CoMP networks: *uplink I/Q bandwidth is significantly greater than downlink bandwidth demands.* This is because in cooperative transmissions, the I/Q symbols, rather than the individual data bits, are transmitted over the fronthaul.

*Downlink* traffic can be generated locally at each BBU, using techniques such as that employed by OpenRF [7]. *Uplink* traffic, on the other hand, consists of I/Q data and requires *up to 24*$\times$ more (i.e., $>$ an order-of-magnitude) fronthaul bandwidth than downlink traffic. Given that the number of downloaded *bits* exceeds that of uploaded bits by only a factor of six [8], uplink CoMP fronthaul traffic will easily saturate the fronthaul network. Hence, we focus on addressing the CoMP challenges for the uplink traffic.

### C. I/Q Data Framing

I/Q data that is exchanged between the BBUs and RRUs are grouped into *frames*. Data framing in this manner is compatible with both stream-based protocols such as CPRI, or packet-based ones such as Ethernet. This enables SPIRO to operate over a wide variety of fronthaul network protocols.

## III. SPIRO DESIGN

### A. Design Objectives

SPIRO is designed with the following salient properties.

**Property 1: Bandwidth-Awareness.** It is difficult to accurately and efficiently track the rapidly changing fronthaul capacity. Hence, the compressed RF streams must be *shapable*—in the event of network congestion, the fronthaul switches must be able to randomly drop the specially-constructed frames carrying RF information without significantly affecting the wireless capacity of the CoMP network.

**Property 2: Bandwidth-Reduction.** A CoMP system relies on both spatial diversity and multiplexing gain from multiple RRUs for cooperative demodulation. SPIRO coordinates the real-time compression of RF signal from each RRU by reducing the number of bits used to quantize I/Q samples, so that the fronthaul bandwidth demand of the CoMP system is reduced. The challenge in this distributed compression approach comes from the fact that it must be coordinated using only the Channel State Information (CSI) of the channel from each RRU, and without detailed knowledge of the statistics of the received data signal.

**TABLE I:** Variables and parameters used in SPIRO.

| | |
|---|---|
| $T_{\text{config}}$ | Configuration interval |
| $R$, $R_n$ | ADC quantization width, indexed by the $n^{\text{th}}$ RRU |
| $\mathbf{R}_{\text{supp}}$ | All supported quantization widths |
| $R_{\max}$ | Maximum ADC quantization width |
| $R_{\min}$ | Minimum ADC quantization width |
| $K$ | Number of low priority bits in sampled signal |
| $N_R$ | Number of RRUs in the CoMP network |
| $\mathbf{S}$ | Set of all RRUs within the CoMP network, with $\lvert\mathbf{S}\rvert = N_R$ |
| $\mathbf{S_R}$ | Set of active RRUs within an interval $T_{\text{config}}$ |
| $N_T$ | Number of concurrent mobile transmitters |
| $N_Q$ | Number of priority queues |
| $x^{(R)}$ | ADC output quantized with $R$ bits |
| $C$ | Current available fronthaul capacity |
| $C_{\text{res}}$ | Reserved fronthaul capacity |
| $C_m$ | Measured available fronthaul capacity |
| $C_{\max}$ | Maximum fronthaul capacity required by SPIRO |

**Property 3: Minimal RRU Usage.** Multiple operators typically share the same CoMP deployment to reduce installation costs. Hence, CoMP network deployments must share the set of RRUs across multiple wireless protocols. SPIRO aims to minimize the number of RRUs required to meet a pre-specified wireless channel capacity. The selection of RRUs must consider the compression ratio at each RRU, and vice versa [9].

**Property 4: Low Complexity.** Signals must be compressed at the RRUs before they are transmitted to the backend for further processing. Hence, to minimize the computational resources at the RRUs, the compression algorithm must have low complexity and be executed quickly. In SPIRO, signal quantization is the primary means of compression. A quantized version of a sampled signal can be either obtained directly from the ADC, or via a simple table lookup. This ensures that the implementation overhead remains low.

### B. Design Overview

SPIRO is designed to operate within a CoMP/C-RAN infrastructure as shown in Fig. 3. SPIRO consists of two key components: SPIRO-FH and SPIRO-RRU. Table I lists the variables and parameters used by SPIRO .

**SPIRO-FH** is a controller module that executes in the hierarchical fronthaul. It monitors the I/Q traffic in the fronthaul and determines (a) which RRUs to enable or disable, (b) the compression level employed by each active RRU, and (c) the priority of each I/Q data frame.

**SPIRO-RRU**, which runs continuously on each RRU, receives configuration information from SPIRO-FH. If the RRU is active (i.e., it is in $\mathbf{S_R}$), it compresses the uplink I/Q samples from the ADC according to its pre-computed quantization width. It then transmits the I/Q data frames back to the BBUs for processing. Note that the bandwidth overhead of control signaling is only a small fraction of the bandwidth of the I/Q data.

### C. SPIRO-FH

Fig. 4 illustrates the operation of SPIRO-FH. At the start of each configuration interval $T_{\text{config}}$, SPIRO-FH receives the

CSI from all CoMP RRUs in the network and the measured available fronthaul bandwidth $C_m$. It then executes the *compression* and *frame prioritization* stages.

*1) RF Compression Stage:* The amount of fronthaul bandwidth required by the CoMP system can be reduced by compression. SPIRO compresses the I/Q samples primarily using quantization.

**Lossy Compression via Quantization.** The ADCs in RRUs map the analog input signal into a complex-valued fixed-point numbers with each of the I and Q components spanning $R$ bits. Let $x^{(R)}$ be a sampled value (either I or Q) that is quantized using $R$ bits. ADCs typically use $R = 12$ or $14$ to minimize the distortion that will be introduced into a wide variety of signals.

We compress these sampled signals lossily by using $r < R$ bits to represent them. The I and Q components are rounded to the nearest $r$-bit fixed-point number using

$$\Delta(x^{(r)}) = round\left(x^{(R)} \cdot 2^{r-1}\right)/2^{r-1}. \quad (2)$$

Since actual value of each I/Q component is between $\pm 2^{-(r-1)}$, the total signal-to-quantization noise ratio (SQNR) is given by

$$\text{SQNR(dB)} = 20\log_{10}(2^r). \quad (3)$$

Hence, every one-bit reduction in the number of quantization bits results in a 6.02 dB reduction in SQNR. Our evaluation will show that a decrease in SQNR does not necessarily decrease the wireless throughput.

**Selecting the Appropriate *Lossy* Compression Level.** The fronthaul bandwidth demand depends on the number of active RRUs, $|\mathbf{S_R}|$, and the ADC quantization width used by the active RRUs, $\mathbf{R}_{\text{opt}} = \{R_n | n \in \mathbf{S_R}\}$. Given a CoMP transmission with $N_T$ transmitters and $|\mathbf{S_R}|$ receiving RRUs, the achievable wireless capacity is given by [10]

$$C_{\text{wl}}(\mathbf{S_R}, \{R_n | n \in \mathbf{S_R}\}) = \log_2 \det\left(\mathbf{I} + \mathbf{H}^*\mathbf{Q}^{-1}\mathbf{H}\right) \quad (4)$$

where $\mathbf{H}$ is the $|\mathbf{S_R}| \times N_T$ CSI of the system and $\mathbf{Q}$ is the SNR of the system given by

$$\mathbf{Q} = \text{diag}\left([\rho_1 + \gamma(R_1), \ldots, \rho_{|\mathbf{S_R}|} + \gamma(R_{|\mathbf{S_R}|})]\right).$$

$\rho_n$ and $\gamma(R_n)$ are, respectively, the channel and quantization noises for the $n^{\text{th}}$ RRU, $n \in \mathbf{S_R}$. The corresponding fronthaul capacity demand is proportional to

$$C_{\text{fronthaul}}(\mathbf{S_R}, \{R_n | n \in \mathbf{S_R}\}) \propto \sum_{n \in \mathbf{S_R}} R_n. \quad (5)$$

If SPIRO determines that the fronthaul bandwidth demand can be increased, it can achieve a corresponding increase in wireless capacity by increasing either the number of active RRUs in $\mathbf{S_R}$, or the number of quantization bits used by each RRU, or both. However, the actual wireless bandwidth gain due to each of these options depends on (a) the channel state and (b) the noise seen at each RRU. Unfortunately, the optimal choice of active RRUs and quantization widths that gives the greatest overall wireless capacity can only be found via an exponential-time 2D search over the space defined by $\mathbf{S_R}$ and $R_n$. In SPIRO, instead of using such an expensive approach, we adopt the heuristic in §IV to obtain $\mathbf{S_R}$ and $R_n$.

*2) Frame Partitioning & Prioritization Stage:* SPIRO prioritizes the I/Q data frames to achieve bandwidth-aware I/Q transmission so that the fronthaul switches can drop frames, according to priority, during a congestion event without significant impact on the wireless channel quality.

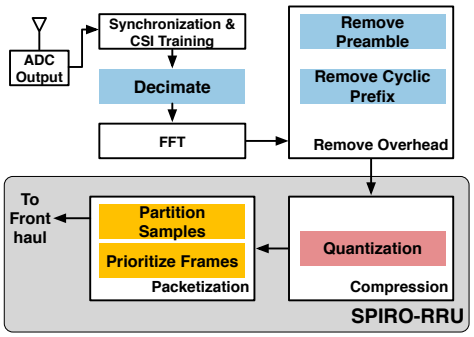



**Fig. 6:** SPIRO-RRU controller on the RRU.

At each RRU, let $x^{(R)}$ be the ADC output of an I/Q component that is quantized using $R \leq R_{\text{max}}$ bits. As an example, consider the case where SPIRO partitions $x^{(R)}$ into two components $x^{(R-K)}$ and $y^{(K)}$. $x^{(R-K)}$ is simply the value of $x^{(R)}$ further quantized using only $R - K$ bits and

$$y^{(K)} = x^{(R)} - x^{(R-K)}$$

is encoded using $K$ bits. Each RRU then creates two different data frames, one that contains only $x^{R-K}$ samples and the other only $y^{(K)}$ samples, as shown in Fig. 5. We refer to these frames as the *primary* and *secondary* I/Q frames, respectively.

SPIRO partitions each $x^{(R)}$ sample into one primary frame and one or more secondary frames. To ensure decodability at the DSP cloud, the primary frame always has a higher priority than the secondary frames.

We reconstruct $x^{(R)}$ from the primary and secondary frames according to

$$x^{(R)} = x^{(R-K)} + y^{(K)}.$$

If the secondary frame is dropped, I/Q sample information is still preserved in $x^{(R-K)}$, albeit with higher quantization noise. However, we cannot recover any information from the secondary frame alone. Hence, SPIRO assigns the primary frame a higher priority than the secondary frame.

Let $N_Q$ be the number of priority queues available in the fronthaul network. SPIRO-FH sorts the primary and secondary frames from all RRUs in decreasing order of their priorities. The sorted frames are then divided equally amongst the $N_Q$ priority frames in order of priority. For example, if $N_Q = 2$, SPIRO-FH maps the first half of the sorted frames to the high-priority queue and the bottom half of the frames to the low-priority queue.

Note that the frame partitioning step is a form of multiple-description coding (MDC) [11]. However, we emphasize that due to the short I/Q frames that are needed to maintain low latency, the length of the MDC is short. SPIRO enhances the performance of MDC with frame prioritization to minimize the probability of the loss of all coded data from the same I/Q frame.

### *D. SPIRO-RRU*

Fig. 6 shows the operation of SPIRO-RRU that executes continuously on each RRU. The SPIRO-RRU first locally processes all parts of a frame that does not require cooperative

decoding. This reduces the number of I/Q samples that need to be sent to the DSP cloud, which in turn reduces the demand for fronthaul bandwidth.

**Redundancy Elimination before Compression.** PHY layer transmissions include redundant information due to the OFDM cyclic prefix, oversampling, preamble and pilot tones that are used for time and frequency synchronization, and channel state measurements. These redundancies can be trivially eliminated at the RRUs and are not transmitted over the fronthaul network. We emphasize that **SPIRO** ***only operates on critically-sampled (i.e., not oversampled) I/Q signals that have all redundancies eliminated. Hence, all reductions in fronthaul bandwidth demands by*** **SPIRO** ***are achieved with respect to critically-sampled I/Q signals.***

**Compression.** The I/Q samples are quantized using $R_n$ bits, as specified by SPIRO-FH.

**Packetization.** SPIRO-RRU then partitions the remaining I/Q data samples into primary and secondary components, and constructs the corresponding frames from them. These frames are sent over the fronthaul to the BBUs.

**Supported Range of Quantization Widths, $\mathbf{R}_{\text{supp}}$.** For clarity, we show the quantization step in Fig. 6 to be after the FFT operation. However, in a hardware implementation, the quantization of data symbols can occur *before* the finite-precision FFT without incurring any additional loss of precision. For example, quantization can be carried out by using multi-resolution ADCs [12] to improve efficiency. To address this possibility, we also evaluate the performance of lossy compression under a finite set of quantization widths, $\mathbf{R}_{\text{supp}}$.

## IV. ALGORITHMS IN SPIRO

### A. Bandwidth Compression

I/Q quantization in SPIRO involves a trade-off between spatial diversity and quantization noise. SPIRO-FH can adopt two different approaches to lossy compression: uniform and non-uniform quantization.

**Non-Uniform Quantization.** We search over all combinations of supported ADC quantization widths, $\mathbf{R}_{\text{supp}}$, and RRU subsets to find the optimal solution pair, $(\mathbf{S_R}, \mathbf{R}_{\text{opt}} = \{R_n | n \in \mathbf{S_R}\})$, of quantization rates and RRUs. Unfortunately, the optimal solution is found via a complicated combinatorial integer optimization, which severely limits its applicability to real-time environments. Thus, we relax the integer constraints to obtain a convex optimization formulation that can be solved quickly.

**Uniform Quantization.** Our compression algorithm is simplified even further by using only the same quantization for all RRUs and a sub-optimal antenna selection algorithm [9]. Our evaluation results indicate that given the same fronthaul capacity constraints, it achieves similar wireless channel throughput to the non-uniform algorithm. However, the uniform quantization approach uses more RRUs than the non-uniform algorithm.

*1) Uniform Quantization:* Algorithm 1 describes the uniform quantization. For each supported quantization width $R \leq R_{\max}$, we determine the optimal set of RRUs, $\mathbf{S_R}$, using the `FindActiveRRUs` function in Algorithm 2. We then select the optimum $(R, \mathbf{S_R})$ pair that achieves the highest wireless bandwidth, under the constraint that the fronthaul bandwidth demand does not exceed the measured available bandwidth.

**Algorithm 1:** Uniform quantization

**Input**: $\mathbf{H} = [\mathbf{H}_f, f = 1, \ldots, N_{\text{FFT}}]$ is a vector of $N_R \times N_T$ CSI matrices, one for each OFDM subcarrier; $C_m$ is the measured available fronthaul capacity
**Output**: $(\mathbf{S}_{\text{opt}}, R_{\text{opt}})$
**Data**: $\mathbf{S}$ = Set of all RRUs in a CoMP network

1 **begin**
2  $b_{\max} \leftarrow 0$;
3  **for** $R \in \mathbf{R}_{\text{supp}}$ **do**
4  $\mathbf{S_R} \leftarrow$ `FindActiveRRUs`$(\mathbf{S}, \mathbf{H}, R, C_m)$;
5  $\mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})}) + \mathbf{I}_{|\mathbf{S_R}|} \cdot 2^{-2R}$;
6  $b \leftarrow \sum_{f=1}^{N_{\text{FFT}}} \log_2 \det\left(\mathbf{I}_{N_T} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{Q}^{-1} \mathbf{H}_f^{(\mathbf{S_R})}\right)$;
7  **if** $b > b_{\max}$ **then**
8  $b_{\max} \leftarrow b$; $R_{\text{opt}} \leftarrow R$; $\mathrm{S}_{\text{opt}} \leftarrow \mathrm{S_R}$;
9  **end**
10  **end**
11 **end**

**Algorithm 2:** `FindActiveRRUs`

**Input**: $\mathbf{S}$ is the set of all RRUs in CoMP network;
$\mathbf{H} = [\mathbf{H}_f, f = 1, \ldots, N_{\text{FFT}}]$ is a vector of $N_R \times N_T$ CSI matrices, one for each OFDM subcarrier; $R$ is the ADC quantization width; $C_m$ is the measured available fronthaul capacity
**Output**: $\mathbf{S_R}$ = Set of selected RRUs
**Data**: $N_{\text{FFT}}$ = number of OFDM subcarriers

1 **begin**
2  $\mathbf{S_R} \leftarrow \mathbf{S}$;
3  $V \leftarrow$ compute bits per I/Q sample from $C_m$;
4  $v \leftarrow |\mathbf{S_R}| \times R$;
5  **while** $v > V$ **do**
6  $\mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})}) + \mathbf{I}_{|\mathbf{S_R}|} \cdot 2^{-2R}$;
7  **foreach** $1 \leq f \leq N_{\text{FFT}}$ **do**
8  $\mathbf{B}_f \leftarrow \left(\mathbf{I}_{N_T} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{Q}^{-1} \mathbf{H}_f^{(\mathbf{S_R})}\right)^{-1}$;
9  **end**
10  $k_{\min} \leftarrow \arg\min_{k \in \mathbf{S_R}} \sum_{f=1}^{N_{\text{FFT}}} \left|\mathbf{H}_f^{(k)*} \mathbf{B}_f \mathbf{H}_f^{(k)}\right|$;
11  $\mathbf{S_R} \leftarrow \mathbf{S_R} \setminus \{k_{\min}\}$;
12  $v \leftarrow |\mathbf{S_R}| \times R$;
13  **end**
14 **end**

In these algorithms, $\mathbf{n}$ is the vector of channel noise at each RRU and $\mathbf{n}^{(\mathbf{S_R})}$ is a subvector consisting only of the elements indexed by $\mathbf{S_R}$. $\mathbf{H}_f$ denotes an $N_R \times N_T$ CSI matrix of the $f^{\text{th}}$ subcarrier and $\mathbf{H}_f^{(\mathbf{S_R})}$ denotes a submatrix using rows from $\mathbf{H}_f$.

The key operation in Algorithm 2 is found in lines 7-10. Here, `FindActiveRRUs` searches for the RRU that contributes the least to the wireless capacity. This RRU is dropped from the active set to reduce the fronthaul bandwidth demand. Let $\mathbf{S_R}^{(-k)} \triangleq \mathbf{S_R} \setminus \{k\}$ for some $k \in \mathbf{S_R}$. The capacity of $\mathbf{S_R}^{(-k)}$ RRUs is

$$C(\mathbf{S_R}^{(-k)}) = \log_2 \det\left(\mathbf{I}_{|\mathbf{S_R}^{(-k)}|} + \mathbf{H}_f^{(\mathbf{S_R}^{(-k)})*} \mathbf{Q}^{-1} \mathbf{H}_f^{(\mathbf{S_R}^{(-k)})}\right)$$
$$= C(\mathbf{S_R}) + \log_2\left(1 - \mathbf{H}_f^{(k)*} \mathbf{B} \mathbf{H}^{(k)}\right) \quad (6)$$

where $\mathbf{B}$ is defined in line 8. Removing the $k^{\text{th}}$ RRU reduces the wireless capacity by $\sum_{f=1}^{N_{\text{FFT}}} \log_2\left(1 - \mathbf{H}_f^{(k)*} \mathbf{B} \mathbf{H}^{(k)}\right)$. In each iteration, the RRU that incurs the smallest capacity reduction is dropped.

*2) Non-Uniform Quantization:* The set of non-uniform quantization values can be determined using the following steps.

1) For each $\mathbf{S_R} \subset \mathbf{S}$, find $\mathbf{R} = \{R_n | n \in \mathbf{S_R}\}$ using
$$\max \sum_{f=1}^{N_{\text{FFT}}} \left| \log_2 \det \left( \mathbf{I}_{|\mathbf{S_R}|} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{U}^{-1} \mathbf{H}_f^{(\mathbf{S_R})} \right) \right|$$
$$\text{s.t.} \sum_{n=1}^{N_R} 2R_n \leq V, \quad R_n \in \mathbf{R}_{\text{supp}}$$
where $\mathbf{U} = \text{diag}\left( \mathbf{n}^{(\mathbf{S_R})} + 2^{-2\mathbf{R}^{(\mathbf{S_R})}} \right)$.
2) Choose the $(\mathbf{S_R}, \mathbf{R})$ pair that achieves the highest wireless capacity, as according to Eq. (4).

However, actually performing this optimization is challenging because (a) it requires a combinatorial search over all subsets of RRUs and (b) the optimization problem is an NP-complete integer programming problem as $R_n$ only takes integer values. Instead, we solve a simplified problem
$$\max \sum_{f=1}^{N_{\text{FFT}}} \log_2 \det \left( \mathbf{I} + \mathbf{H}_f \mathbf{H}_f^* \mathbf{W}^{-1} \right)$$
$$\text{s.t.} \sum_{n=1}^{N_R} 2\overline{R}_n \leq V, \quad 0 \leq \overline{R}_n \leq R_{\max}$$
where $\mathbf{W} = \text{diag}\left( \mathbf{n} + 2^{(-2\overline{\mathbf{R}})} / \overline{\mathbf{R}} \right)$ and $\overline{\mathbf{R}} = [\overline{R}_n, \ldots, \overline{R}_{N_R}]$. Note that $\overline{R}_n$ are real, not integer, values. We then use the RRU-selection step, as shown in Algorithm 3, to obtain the final RRU selection and corresponding quantization width, $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$.

**Algorithm 3:** Non-uniform RRU selection

**Input**: $\overline{\mathbf{R}} = [\overline{R}_1, \ldots, \overline{R}_{N_T}]$
**Output**: $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$
1 **begin**
2 $\quad \mathbf{R} \leftarrow [R_1, \ldots, R_{N_T}]$ where $R_n = \max(\min(\lceil \overline{R}_n \rceil, R_{\max}), R_{\min})$ for $1 \leq n \leq N_T$;
3 $\quad$ **while** $\sum_{n=1}^{N_T} R_n > V$ **do**
4 $\quad\quad k \leftarrow \arg\min_{1 \leq n \leq N_T} \overline{R}_n$; $R_k \leftarrow 0$; $\overline{R}_k \leftarrow \infty$;
5 $\quad$ **end**
6 $\quad \mathbf{S_R} \leftarrow \{n | R_n > 0\}$;
7 $\quad \mathbf{R}_{\text{opt}} \leftarrow \{R_n | n \in \mathbf{S_R}\}$;
8 **end**

### *B. Frame Prioritization*

SPIRO uses Algorithm 4 to construct the quantization width used in the primary and secondary I/Q frames. We first compute the optimal $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ given the measured fronthaul capacity constraint, $C_m$, using either the uniform or non-uniform antenna selection. Also, let $\lambda$ be the smallest number of quantization bits used to represent each I/Q sample in the secondary frame. In our implementation, we find that $\lambda = 2$ bits offers the best results. The frame prioritization algorithm takes $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ and $\lambda$ as input, and computes the priority of primary and secondary frames from each active RRU.

In the first while-loop (lines 4-15), we partition the I/Q samples from each RRU into multiple groups of $\lambda$ bits, down to a minimum partition size of $R_{\min}$. These $\lambda$-bit partitions are enqueued into $\mathbf{P}$ in order of increasing priority. This is followed by the second while-loop (lines 17-24) where we prioritize the remaining $R_{\min}$-bit I/Q samples from all RRUs.

**Algorithm 4:** Compute the priority of I/Q frame partitions

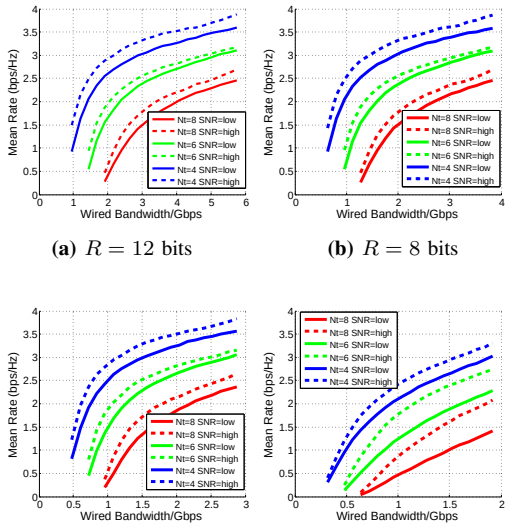
**Input**: $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$, $\lambda$
**Output**: $\mathbf{P}$ is the priority queue of I/Q frame partitions
1 **begin**
2 $\quad \mathbf{R} \leftarrow \mathbf{R}_{\text{opt}}$;
3 $\quad \mathbf{P} \leftarrow []$;
4 $\quad$ **while** $\exists R_n \geq R_{\min} + \lambda$, $n \in \mathbf{S_R}$, $R_n \in \mathbf{R}$ **do**
5 $\quad\quad b_{\max} \leftarrow 0$; $n_{\max} \leftarrow []$;
6 $\quad\quad$ **foreach** $n \in S_R$ **do**
7 $\quad\quad\quad$ **if** $R_n \leq \lambda$ **then continue**;
8 $\quad\quad\quad R'_n \leftarrow R_n - \lambda$;
9 $\quad\quad\quad \mathbf{R}' \leftarrow [R_1, \ldots, R_{n-1}, R'_n, \ldots, R_{|\mathbf{S_R}|}]$;
10 $\quad\quad\quad \mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})} + 2^{-2\mathbf{R}'})$;
11 $\quad\quad\quad b \leftarrow \sum_{f=1}^{N_{\text{FFT}}} \log_2 \det \left( \mathbf{I}_{|\mathbf{S_R}|} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{Q}^{-1} \mathbf{H}_f^{(\mathbf{S_R})} \right)$;
12 $\quad\quad\quad$ **if** $b > b_{\max}$ **then** $b_{\max} \leftarrow b$; $n_{\max} \leftarrow n$; ;
13 $\quad\quad$ **end**
14 $\quad\quad R_n \leftarrow R_n - \lambda$;
15 $\quad\quad \mathbf{P} \leftarrow \text{append}(\mathbf{P}, (n_{\max}, \lambda))$;
16 $\quad$ **end**
17 $\quad$ **while** $|\mathbf{S_R}| > 0$ **do**
18 $\quad\quad \mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})}) + \mathbf{I}_{|\mathbf{S_R}|} \cdot 2^{-2\mathbf{R}^{(\mathbf{S_R})}}$;
19 $\quad\quad$ **foreach** $1 \leq f \leq N_{\text{FFT}}$ **do**
20 $\quad\quad\quad \mathbf{B}_f \leftarrow \left( \mathbf{I}_{N_T} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{Q} \mathbf{H}_f^{(\mathbf{S_R})} \right)^{-1}$;
21 $\quad\quad$ **end**
22 $\quad\quad k_{\min} \leftarrow \arg\min_{k \in \mathbf{S_R}} \sum_{f=1}^{N_{\text{FFT}}} \left| \mathbf{H}_f^{(k)*} \mathbf{B}_f \mathbf{H}_f^{(k)} \right|$;
23 $\quad\quad \mathbf{P} \leftarrow \text{append}(\mathbf{P}, (k, R_k))$;
24 $\quad\quad \mathbf{S_R} \leftarrow \mathbf{S_R} \setminus \{k_{\min}\}$;
25 $\quad$ **end**
26 **end**

Each entry in the priority queue $\mathbf{P}$ is an $(n, r)$ pair where $n$ is the RRU identifier and $r$ is the number of quantization bits to be used at this priority. SPIRO maps $\mathbf{P}$ to $N_Q$ priority queues, similar to those found in Ethernet switches, by partitioning the entries in $\mathbf{P}$ equally among the $N_Q$ queues. If multiple $(n, r)$ entries from the same RRU are in the same switch priority queue, they are merged into one larger secondary frame.

## V. EVALUATION

We implement and evaluate SPIRO on a testbed of 16 WARP SDR platforms running WARPLab, each with two antennas [13]. The fronthaul network is constructed using a single HP 6600 48-port switch. The WARP boards are connected to this switch. All WARP platforms are globally time and frequency synchronized. The antennas are placed throughout a large server room environment. Obstructions throughout the testbed ensure existence of both line-of-sight and non-line-of-sight channels between different antenna pairs.

In each experiment, we randomly select $N_R = 24$ antennas as uplink RRUs and $N_T = 4, 6$ or $8$ antennas as concurrent transmitters. We transmit 500 OFDM frames from the $N_T$ transmitters. Each OFDM frame spans $800\mu s$ at a bandwidth of 20MHz, and uses symbols that have 256 subcarriers and 64-tap cyclic prefixes. SPIRO uses the preamble from all $N_R$ antennas to determine the optimal compression solution $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ and decode the transmitted frame from the active antennas at the corresponding quantization widths. The smallest number of RRUs is always constrained by $N_R = N_T$ to




**(a)** $R = 12$ bits **(b)** $R = 8$ bits

**(c)** $R = 6$ bits **(d)** $R = 4$ bits

**Fig. 7:** SPIRO with uniform quantization.

ensure MIMO decodability. If $N_R > N_T$, the wireless capacity benefits from additional spatial diversity.

Our results are obtained using two SNR ranges, high and low, with median SNRs of 14 and 3 dB respectively, as shown in Fig. 8. We evaluate (a) the uniform and non-uniform quantization algorithms, and (b) the performance of frame prioritization in the event of fronthaul bandwidth fluctuations.

### A. Quantization

*1) What Is the Baseline Evaluation of Our Uplink CoMP Testbed?:* Fig. 7a shows the wireless rate per user achieved by RRU selection under 12-bit uniform quantization as we increase the wired fronthaul capacity available to SPIRO. The I/Q samples here do not require any additional quantization since the WARP platforms already come equipped with 12-bit ADCSs. The achievable wireless rate depends on (a) the number of RRUs selected, (b) the number of concurrent uplink users and (c) the SNR distribution at the RRUs.

**Number of Active RRUs.** With uniform quantization, the fronthaul bandwidth demand is met by varying the number of active RRUs that send I/Q samples back to the DSP cloud. As we increase the number of active RRUs, the wireless rate per user increases due to the increased spatial diversity. For $N_T = 4, 6$ and $8$ transmitters, the wireless rate per user reaches a maximum of 3.8, 3 and 2.55 bits/s/Hz under high SNR when all 24 RRUs are active.

**Number of Concurrent Transmitters.** The achieveable mean wireless rate per user decreases as we increase the number of concurrent users. This is due to the increased interference encountered from the imperfections in time and frequency synchronization that is found in real-world uplink transmitters. Such imperfections lead to power leakage from the channel of one transmitter to another, thus reducing the SNR of each of the $N_T$ decoded frames.

**SNR.** The wireless rate per user is lower with the low SNR

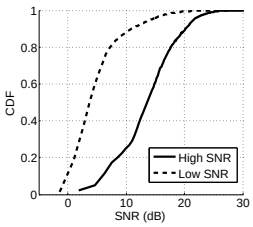



**Fig. 8:** Experiments in two separate SNR environments.

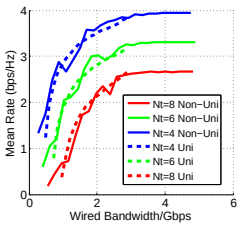



**Fig. 9:** Mean rate of non-uniform vs. uniform quantization under the same fronthaul capacity bound.

experiment as expected. However, the rates achieved by the low and high SNR experiments are within 10% of each other.

*2) How Much Fronthaul Bandwidth Can We Save by Reducing the Quantization Width of All RRUs?:* Figs. 7b and 7c show the wireless rate per user under increasing wired bandwidth constraints when we quantize the I/Q samples with 8 and 6 bits, respectively. Note that one can quantize I/Q samples from our testbed using 6 bits (down from the original 12-bit ADC output) without any loss of wireless performance. There are two key findings to observe.

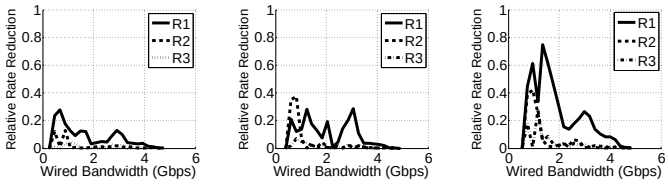
First, *given the same target rate per user, when we reduce the number of quantization bits from 12 to 6, the fronthaul bandwidth requirement is reduced by 50% from the original 12-bit I/Q samples and the number of RRUs required remains unchanged.*

Second, *under uniform quantization, the achievable wireless capacity is dominated by the degree of spatial diversity as we reduce the number of quantization bits to 6.*

However, we cannot quantize the I/Q samples with fewer than 6 bits without any loss in wireless capacity. As an example, compare the performance of $R = 6$ with that of $R = 4$. When we have a fronthaul capacity limit of 1 Gbps, we achieve 2.3 bits/s/Hz when using $R = 4$ and 2.8 bits/s/Hz with $R = 6$. This is despite the fact that 12 RRUs are active with $R = 4$ while only 8 are used with $R = 6$. This disparity is evident even at other fronthaul bandwidth constraints. Hence, when we use fewer than 6 quantization bits, the increase in quantization noise overwhelms any gains we obtain from increased spatial diversity.

*3) Can We Reduce the Number of Active RRUs?:* We can reduce the number of active RRUs with non-uniform quantization. We use $\mathbf{R}_{\text{supp}} = \{4, \ldots, 12\}$ to demonstrate this. Fig. 9 shows the rate per user of $N_T = 4, 6$ and $8$ with non-uniform quantization under high SNR conditions. We also plot the rate per user with uniform $R = 6$ quantization for comparision. Observe that *for a given fronthaul bandwidth constraint, non-uniform quantization can achieve the same wireless rate as the uniform quantization approach*. Furthermore, non-uniform quantization comes with an added benefit.

Fig. 11 compares the number of RRUs used by non-uniform quantization and $R = 6$ uniform quantization algorithms, for $N_T = 4, 6$ and $8$ transmitters. *Non-uniform quantization*




**(a)** $N_T = 4$ **(b)** $N_T = 6$ **(c)** $N_T = 8$

**Fig. 10:** Mean wireless bitrate deviation with a reduced set of supported quantization widths.

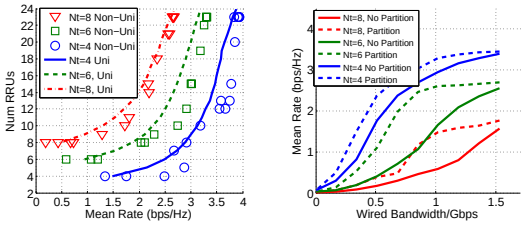



**Fig. 11:** Non-uniform quantization requires up to 43% fewer RRUs than uniform quantization.

**Fig. 12:** Rate per user with frame prioritization.

*requires up to 43% fewer active RRUs to attain the same wireless throughput as uniform quantization.*

Hence, when compared with a CoMP network that relies only on an RRU selection algorithm to manage the fronthaul bandwidth demands, the *non-uniform scheme requires 50% less fronthaul bandwidth and 43% fewer RRUs to maintain the same wireless channel rate per uplink user.*

*4) Can We Achieve the Same CoMP Performance with Fewer Number of Quantization Widths?:* If quantization is implemented using multiple ADCs or multi-resolution ADCs, a smaller number of required quantization widths translates into a more efficient hardware implementation. We consider three different quantization ranges in Fig. 10: $\mathbf{R}_1 = \{4, 12\}$, $\mathbf{R}_2 = \{4, 8, 12\}$ and $\mathbf{R}_3 = \{4, 6, 8, 10, 12\}$. When $N_T = 4$, the reduction in wireless rates under a 1 Gbps (and greater) fronthaul constraint is less than 5% when $\mathbf{R}_2$ and $\mathbf{R}_3$ are used. Such small reductions can also be seen with $N_T = 6$ and 8. However, we see more drastic reductions in throughput with $\mathbf{R}_1$. In particular, when $N_T = 8$, up to 75% relative reduction in wireless rate is seen under a 1 Gbps fronthaul constraint.

### *B. Frame Partitioning and Prioritization*

*1) How Much Benefit Do We Get from Frame Partitioning?:* Fig. 12 compares the wireless rate per user using frame prioritization with and without frame partitioning, under the high SNR scenario. To obtain these results, we compute the optimal $(\mathbf{S}_\mathbf{R}, \mathbf{R}_{\text{opt}})$ solution given a fronthaul capacity $C_m$ of 1.5 Gbps using non-uniform quantization. The partitioned and unpartitioned I/Q streams are generated using $\lambda = 2$ and $\lambda = 0$ in Algorithm 4, respectively. We then reduce the fronthaul bandwidth usage by discarding Ethernet frames carrying I/Q samples at the switch, in order of priority. To ensure optimal prioritization, we use $N_Q = 80$ priority queues—each primary or secondary frame is in its own queue and in the event of congestion, frames are dropped in a strict order of priority.

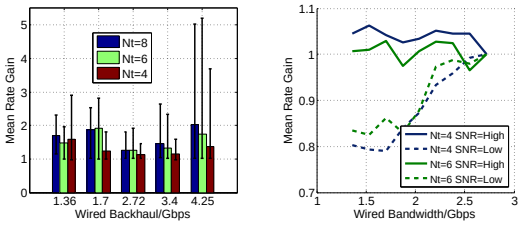
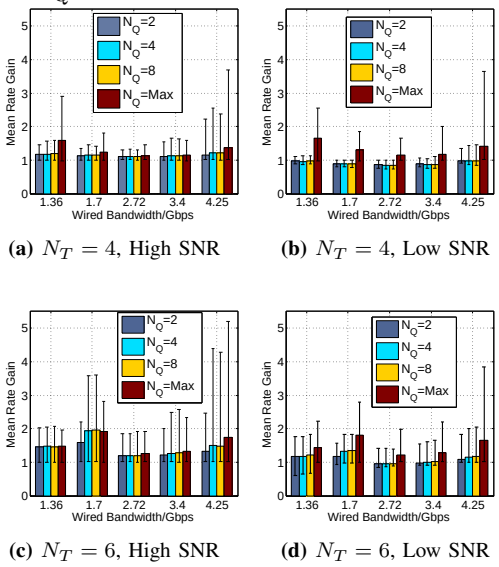



**Fig. 13:** Rate gain with frame partitioning vs without partitioning, under different fronthaul capacity constraints, $C_m$ $(\mathbf{S}_\mathbf{R}, \mathbf{R}_{\text{opt}})$. and $N_Q = 80$.

**Fig. 14:** Wireless rate gain of priority-based frame drops vs optimal compression using




**(a)** $N_T = 4$, High SNR **(b)** $N_T = 4$, Low SNR

**(c)** $N_T = 6$, High SNR **(d)** $N_T = 6$, Low SNR

**Fig. 15:** Gains in wireless rate per user from frame partitioning and prioritization. Each bar shows the mean gain, while the error bars denote the maximum and $5^{\text{th}}$ percentile gains.

By partitioning the I/Q samples into primary and secondary Ethernet frames, we ensure that frame losses primarily increase quantization noise, while maintaining spatial diversity for as long as possible. This has two primary consequences: *(a) frame partitioning and prioritization has greater benefits for transmissions with a larger number of concurrent users (i.e* $N_T = 6$ *and* 8*)* and *(b) in the event of frame losses at the switch, we retain up to 3× more wireless capacity with* SPIRO *frame partitioning and prioritization*. Fig. 13 shows that this observation holds at other fronthaul constraints $C_m$. Each bar shows the average gain in the wireless rate per user, while the error bars demarcate the maximum and $5^{\text{th}}$ percentile gains.

*2) How Does Frame Partitioning Perform with Fewer Priority Queues?:* Commercially available Ethernet switches have far fewer than 80 priority queues. However, we can still benefit from frame partitioning and prioritization with fewer queues. Fig. 15 shows the gains under $N_Q = 2, 4$ and 8 priority queues. Under high SNR situations, improvements in per-user

rates are achieved with fewer priority queues, with situations involving a larger number of concurrent users, $N_T = 6$, seeing larger gains than those with fewer concurrent users, $N_T = 4$. However, under low SNR conditions, frame partitioning and prioritization have a small negative impact on the per-user rates when $N_T = 4$ concurrent users are active. In such situations, a larger number of priority queues is necessary to obtain the benefits of frame prioritization in SPIRO.

*3) How Well Does Priority-based Frame Drops Compare with Optimal I/Q Compression?:* We compare the wireless rate achieved by using priority-based frame-drops with that obtained by our optimal bandwidth compression in Fig. 14. For the frame prioritization algorithm, we use $C_m = 2.4$ Gbps. We see that under high SNR, the wireless rate achieved by frame prioritization and drops, is similar to that obtained by optimal compression. However, at low SNR, optimal compression achieves up to 20% higher wireless rate than frame dropping at the switch.

## VI. Related Work

Practical network MIMO or CoMP schemes [2], [14] usually assume that the fronthaul is capable of transporting the I/Q samples necessary for centralized (de)modulation. However, this assumption may not hold in the presence of interfering cross traffic over a shared fronthaul. Quantization of RF data [15]–[17] has been proposed to reduce the fronthaul bandwidth demands of next-generation LTE networks. These proposals focus on compressing RF data from each RRU individually, and do not exploit the spatial diversity between antennas. To address this limitation, distributed Wyner-Ziv [18] encoding has been used to jointly compress signals from multiple antennas. Compressed sensing [19]–[21] takes a different approach where the the signal is compressed before sampling and digitization by the ADC. However, most WiFi and LTE data signals are not transmitted sparsely, thus limiting the applicability of compressed sensing to these scenarios.

Datacenters in Cloud-RAN deployments are known to have rapidly changing flow behaviors [22], [23] and congestion patterns. Incast TCP traffic [24] also leads to sporadic congestion and packet drops within datacenters. SPIRO accommodates such variability by supporting traffic shaping at the switch in the event of congestion.

## VII. Conclusion

Designing a fronthaul in a C-RAN CoMP system is challenging due to the inelasticity of the I/Q data stream and the high bandwidth demands of the I/Q samples. We designed and implemented SPIRO, a novel bandwidth-aware RF transport to (a) minimize the fronthaul bandwidth demands of indoor CoMP systems and (b) adapt to the capacity variability in the fronthaul network. Our evaluation results show that SPIRO reduces fronthaul bandwidth demands by more than 50% without any degradation of the achievable wireless capacity. At the same time, it minimizes the number of active RRUs necessary to improve the sharing of the CoMP system with multiple mobile operators.

## Acknowledgement

The work reported in this paper was supported in part by NSF under grants CNS-1160775 and CNS-1317411.

## References

[1] X. Zhang, K. Sundaresan, M. A. A. Khojastepour, S. Rangarajan, and K. G. Shin, "NEMOx: Scalable network MIMO for wireless networks," in *MobiCom*, 2013.
[2] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous MIMO networks," in *SIGCOMM*, 2011.
[3] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM*, 2008.
[4] C. Peng, S.-B. Lee, H. Luo, and S. Lu, "GreenBSN: Enabling energy-proportional cellular base station networks," *IEEE Trans on Mobile Computing*, 2014.
[5] X. Zhang and K. G. Shin, "E-MiLi: energy-minimizing idle listening in wireless networks," in *MobiCom*, 2011.
[6] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.
[7] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing cross-layer mimo to today's wireless lans," in *SIGCOMM*, 2013.
[8] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *IMC*, 2010.
[9] A. Gorokhov, "Antenna selection algorithms for mea transmission systems," in *ICASSP*, 2002.
[10] D. Tse and P. Viswanath, *Fundamentals of wireless communication*.
[11] V. K. Goyal, "Multiple description coding: Compression meets the network," *Signal Processing Magazine, IEEE*, vol. 18, no. 5, 2001.
[12] D. Hostetler and Y. Xie, "Adaptive power management in software radios using resolution adaptive analog to digital converters," in *IEEE Computer Society Annual Symposium on VLSI*, 2005.
[13] "WARP: Wireless open-access researc platform." http://warp.rice.edu.
[14] S. Gollakota, S. D. Perli, and D. Katabi, "Interference alignment and cancellation," in *SIGCOMM*, 2009.
[15] D. Samardzija, J. Pastalan, M. MacDonald, S. Walker, and R. Valenzuela, "Compressed transport of baseband signals in radio access networks," *Trans. Wireless Comms*, Sept 2012.
[16] P. Marsch and G. Fettweis, "Uplink CoMP under a constrained backhaul and imperfect channel knowledge," *Trans. on Wireless Comms*, 2011.
[17] P. Baracca, S. Tomasin, and N. Benvenuto, "Constellation quantization in constrained backhaul downlink network mimo," *Trans on Comms*, March 2012.
[18] A. del Coso and S. Simoens, "Distributed compression for MIMO coordinated networks with a backhaul constraint," *Wireless Communications, IEEE Transactions on*, Sept 2009.
[19] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Trans on Information Theory*, Feb 2006.
[20] J. Paredes, G. Arce, and Z. Wang, "Ultra-wideband compressed sensing: Channel estimation," *Selected Topics in Signal Proc*, 2007.
[21] P. Zhang, Z. Hu, R. Qiu, and B. Sadler, "A compressed sensing based ultra-wideband communication system," in *ICC*, 2009.
[22] S. Kandula, S. Sengupta, and A. Greenberg, "The nature of data center traffic: measurements & analysis," in *IMC*, 2009.
[23] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," in *SIGCOMM*, 2009.
[24] Y. Chen, R. Griffith, J. Liu, R. H. Katz, and A. D. Joseph, "Understanding TCP incast throughput collapse in datacenter networks," in *WREN*, 2009.