Scalable Real-time Transport of Baseband Traffic

Krishna C. Garikipati, Kang G. Shin
University Of Michigan

Abstract—Upcoming wireless deployments, such as C-RAN and Massive-MIMO, rely on real-time transport of baseband samples. The radios (or radio heads) in such deployments generate baseband packets once every period, which are then transported to a backend processing cluster. Therefore, to meet the real-time processing constraints, the baseband transport network must deliver the packets to the backend within a fixed end-to-end delay bound. Hence, computing the queuing delays in a large-scale packet-switched network is intractable, making it difficult to provide end-to-end delay guarantees.

In this paper, we present a novel Fat-Tree-based design, called DISTRO, for transporting baseband traffic. DISTRO’s design supports real-time transport as it allows us to bound the maximum transport delay of each packet. The network switches can therefore implement well-known real-time scheduling policies to achieve end-to-end delay guarantees. DISTRO further partitions the transport network into separate aggregation- and edge-switch networks such that all scheduling policy changes occur only at the edge-switch network. We also characterize the maximum wireless capacity using DISTRO while meeting the real-time constraints of baseband processing.

I. INTRODUCTION

In upcoming wireless architectures such as C-RAN [1], [2], and Massive-MIMO [3], [4], a baseband transport network connects radios (front-ends) to a backend processing infrastructure. For instance, in a C-RAN deployment, a fronthaul network carries the baseband samples between the remote radio heads (RRHs) and the baseband processors located in a datacenter. Such architectures provide numerous cost advantages: lesser power consumption from resource pooling, quicker upgrade and replacement cycles, support for advanced signal processing, and flexibility in the management of radio infrastructure.

Since the baseband transport network is the key enabler of such architectures, its design must meet the following requirements to maximize the cost benefits.

Guarantees. The real-time nature of the wireless sample processing imposes stringent constraints on the transport network; the network must provide end-to-end (e2e) guarantees for delay and jitter. For instance, the transport delay bound can be as low as few microseconds for WiFi samples [5], to few hundred microseconds for LTE samples [6]. In addition, the transport behavior of the radio samples must be predictable, i.e., given a network topology and the traffic sources, one must be able to model the delivery of the baseband traffic. This model is necessary for an e2e schedulability analysis — determining whether the given network can meet the requested delay bounds. In the real-time systems literature [7], [8], however, modeling packet traffic in the network core is known to be intractable due to the non-periodic nature of arrivals.

Therefore, in a general baseband transport network consisting of multiple switches, it is not entirely clear what packet scheduling policies (implemented by switches) will achieve e2e schedulability.

Aggregation. The baseband transport design should support traffic aggregation from the radios, for scenarios such as MIMO processing, or, for resource pooling in C-RAN [9], where a common compute platform decodes multiple base-stations. Inherently, this can be achieved from a tree-based design using aggregation switches. However, traffic aggregation without proper scheduling introduces variable queuing delays, and moreover, cannot differentiate between traffic flows based on their delay bounds.

Scalable. Considering the size of future radio deployments, the baseband transport design should be scalable. It must be extensible to any number of radios while preserving its predictable behavior. Also, it must require only few additional resources (e.g., cables, switches) to add a large number of radios to the network. The scalability of baseband transport is desirable, if not necessary, for massive MIMO systems which are equipped with tens or hundreds of antennas/radios. Existing datacenter designs [10] that are optimized for scalability, are likely candidates for baseband transport. However, they are primarily designed to handle bisection traffic between the compute clusters, whereas traffic flows in a baseband network are exclusively in the north–south direction.

Optimality. Whether the baseband traffic is schedulable or not depends on the transport rate, which, in turn, depends on the sample quantization widths used at the radios. As the selected quantization widths affect the wireless capacity through quantization noise [11], [12], there is an indirect dependence between schedulability of the traffic and the wireless capacity of the network. Ideally, the network should operate at a point that ensures schedulability but also maximizes wireless capacity.

Based on the above requirements, we introduce DISTRO, a design for real-time baseband transport (such as fronthaul) networks that can potentially scale to a large number of radios. DISTRO utilizes a logical tree structure of radio front-ends and network switches; the radios represent the leaf nodes of the tree, the network switches represent the internal nodes, and the root of the tree is a common aggregation point that connects to a pool of baseband processors. DISTRO’s design supports real-time transport as it allows us to bound the maximum transport delay of each baseband packet. Specifically, using a constrained tree design, the upper bound on the waiting time at any switch can be obtained irrespective of the input arrival sequence, from which one can get the maximum total delay of
a baseband flow. As a result, the network switches can utilize schedulability results from the real-time systems literature and implement scheduling policies (such as EDF [13] and fixed-priority [14]) to achieve e2e delay guarantees.

Since scheduling policies are subject to the baseband traffic parameters, any addition of radios or changes inthem requires policy changes in the entire network, making the design unscalable. Therefore, the tree structure in DISTRO is partitioned into aggregation- and edge-switch networks; the schedulability analysis is done only at the edge-switch network whereas the default packet scheduling (FIFO) is implemented in the aggregation-switch network. This logical division along with the tree-based design enables transport scaling to a large number of radios.

DISTRO selects quantization widths to maximize the wireless capacity while ensuring e2e schedulability. A brute-force search of optimal quantization widths, however, has exponential complexity in the number of radios. By using the monotonic dependence of the wireless capacity and the schedulability on the quantization widths, we propose a greedy-based approach that has much lower runtime complexity.

In summary, we make the following contributions:

- Proposal of a Fat-Tree architecture for scalable deployment of baseband radios;
- Calculation of the maximum delay bound of a baseband packet in the network and its use to achieve e2e schedulability; and
- Characterization of the wireless capacity under the schedulability constraint, and development of an efficient search algorithm to maximize the capacity.

The rest of this paper proceeds as follows. Sec. II provides the background on the baseband transport while Sec. III presents our proposed transport design and its evaluation results in a simulated network scenario. In Sec. IV we obtain the maximum wireless capacity with end-to-end schedulability through our proposed algorithm. Finally, Sec. V presents the related work and Sec. VI concludes the paper.

II. BACKGROUND

This section provides the background on baseband transport and processing, and describes how real-time scheduling is applicable to baseband transport.

A. Wireless Baseband Transport

The radio front-ends in a baseband network (also known as RRHs in C-RAN) act as converters between RF samples and complex (I and Q) baseband samples. Fig. 1 shows the components of such a radio. In the receive mode, the RF signal is down-converted, filtered and passed through an analog-to-digital converter (ADC) that gives out a digitized (e.g., 16-bit) stream of baseband samples. This stream is broken into fixed-size blocks, which are then transported as payloads in special-purpose packets generated with appropriate headers and tags.

Suppose there are \( n \) radios in the network, where each radio is denoted by index \( i \in \{1, \ldots, n\} \). Let \( f_i \) denote the desired sampling frequency from the ADC (achieved through decimation by digital-down converters), and let \( Q_i \) be the number of bits used to represent each I (and Q) baseband sample. Then, the transport data rate (in bits/s) of radio \( i \) can be expressed as:

\[
R_i = 2Q_if_i.
\] (1)

Further, let \( B \) denote the fixed payload size (in bits) of a transport packet. The inter-packet arrival time (in seconds) at radio \( i \), assuming negligible packet overhead, is given by:

\[
T_i = \frac{B}{R_i}.
\] (2)

Since the ADC operates at a fixed frequency, and fixed-size blocks are used, the packet inter-arrival time is a constant at each radio, which we refer to as the period of the arrival process.

Example: USRP [15] is a common software-radio platform that uses the UDP protocol for baseband transport. Fig. 2 shows the timestamps from the transport log of a USRP2 running at 25MHz sampling rate and payload size of 1492 Bytes. Using Eqs. (1) and (2), the packet inter-arrival time with 8-bit quantization is calculated to be 2.98\( \mu \)s. This is indeed close to inter-packet arrival time \( \in [3, 4]\mu \)s observed from the USRP2 logs (Fig. 2). Note that the logged arrival period has a minimum 1\( \mu \)s resolution.

Wireless protocols have a fixed e2e processing deadline for PHY-layer primitives such as channel sensing and decoding. Assuming a fixed (or worst-case) processing time at the baseband processors, the e2e PHY deadlines impose a maximum transport delay. Therefore, in order to support real-time processing, the generated baseband packets from the radios must be transported to the baseband processors within a fixed amount of time. That is, each radio \( i \) has an e2e transport delay bound, \( D_i \), that the transport network must satisfy.

The traffic specification of radio \( i \) is thus given by a 2-tuple \( \tau_i = (T_i, D_i) \), which represents the inter-arrival time, and the e2e delay bound, respectively. The radio traffic is said to be schedulable if for all \( 1 \leq i \leq n \), the maximum delay experienced by packet of radio \( i \) is not greater than the requested delay bound \( D_i \).

The transport network in large deployments of C-RAN runs over a fiber infrastructure such as dark fiber and WDM [2]. While in indoor environments, the radios may be deployed using high-capacity Ethernet or Infiniband links [4]. In both scenarios, baseband samples are exchanged through packet transmissions (most optical networks now offer packet switching for increased flexibility). Thus, the baseband transport network can be modeled as a packet-switch network with one
or more network switches. Every packet in the network passes through multiple links, switches, and routers before reaching its destination. While many routes can exist for a packet, we assume fixed routing in the network, which is necessary for e2e packet delay guarantees [16].

Despite its advantages, packet-switching introduces various delays at each link in a selected route. The e2e packet delay is the summation of delays over links and switches along the selected route, which is composed of:

- Propagation delay \( t_p \): the time taken for the packet to reach the next switch;
- Switching delay \( t_s \): the time taken for the packet to move from the ingress to egress port of a switch;
- Transmission delay: the time needed to transmit the packet, which is a function of the link capacity; and
- Queuing delay: the waiting time of packet in a switch’s egress queue.

Among the different delays, the queuing delay is the only unknown that can be different for each packet in the network. In general, it is a function of the switch’s scheduling policy and the input arrival sequence. One needs to model these delays at each switching stage, which becomes intractable for large networks as the output sequence from a switch is non-periodic even though packets arrive periodically. For example, consider a simple baseband network (Fig. 3) with periodic baseband traffic from 2 radios having inter-arrival times of 2µs and 4µs, respectively. Assume the processing link capacity is 10Gbps and the packet size is 1000 bytes. That is, the output transmission time of both flows is 1µs. Assume packets from the two radios arrive in the queue at the same time instant in the beginning. Fig. 3 shows the timeline of the queue output. As one can see, the inter-arrival time of the 3.3Gbps flow at the output is non-periodic (inter-arrival times of 2µs and 4µs) that is induced from the waiting in the queue. This non-linearity of queuing makes it difficult to model e2e packet arrivals, a fact well-known in the literature [7], [8]. Among them, the deadline scheduling is a natural approach where each arriving packet is assigned a deadline according to the requested delay bound. The packet with the earliest deadline is transmitted first. This scheduler is optimal in the sense that if packets meet their deadlines using any scheduling policy, so will they using deadline scheduling. While the original deadline scheduling considered implicit deadlines (deadline is the same as the period) and preemption, one can generalize it further to obtain both necessary and sufficient conditions for schedulability with arbitrary deadlines. Theorem A.1 (Appendix) formally states these conditions for both cases – with and without preemption.

Deadline scheduling is based on dynamic prioritization and thus difficult to realize in practice. A more feasible approach is the fixed-priority scheduling where each traffic flow is assigned a static priority [14] where incoming packets are transmitted in the order of their priority. As we show in Theorem A.2 (Appendix), there exists a schedulability test to determine whether the set of traffic flows with given priorities meet their delay bounds. Furthermore, one can do an iterative or offline search and use the schedulability criteria to arrive at the feasible priority-assignment policy if one can be found [17]. The necessary conditions for schedulability, however, are known only under special circumstances (e.g., when the deadline is less than or equal to the period).

### B. Transport Delay Bound

The wireless processing design depends on the wireless protocol and the target architecture. For WiFi signals, where slots are 9µs long, baseband samples are streamed and decoded on the fly [5]. In contrast, LTE has 1µs-long subframes and decoding is carried out on an accumulated buffer of baseband samples [6]. The time required to decode the baseband samples (or frames) depends on the capability and the optimizations of the platform. In this paper, we assume the processing time is fixed, and focus on the transport delays.

The transport delay bound of radio \( i \) is computed by subtracting the maximum processing time, \( T_{\text{proc}} \), from its end-to-end protocol deadline, \( T_{\text{prot}} \), as:

\[
D_i = T_{\text{prot}} - T_{\text{proc}}.
\]

In case of WiFi signals, the protocol deadline, \( T_{\text{prot}} \), can be 4µs (since CCA assert should occur within 4µs during energy sensing [18]). Assuming \( T_{\text{proc}} = 2 \mu s \) to perform sample summation, this results in a 2µs delay bound for the transport network. On the other hand, for LTE signals, \( T_{\text{prot}} = 2 \text{ms} \), as there is no channel sensing, and the protocol deadline is governed by the HARQ process. Consequently, the delay bound is much larger (0.5–0.7 ms) than the WiFi case.

It is worth noting that the transport delay bound, \( D_i \), is not always fixed but can vary with the number of radios, \( n \), since the processing time, \( T_{\text{proc}} \), typically increases with \( n \). For instance, \( T_{\text{proc}} \propto O(n^3) \), when decoding \( n \) spatially multiplexed signals [19].
III. DISTRO

We now describe the construction, requirements, and analysis of the proposed real-time transport design for baseband traffic.

A. Design Philosophy

The design of a baseband network is driven by two key observations. First, the baseband traffic flows exclusively between the radios and the processor pool (cross-traffic between radios is negligible). Second, depending on the application, baseband traffic is aggregated into one or more links for processing. In addition, the baseband transport operates in real time: given the traffic flows and delay bounds, we must be able to give a sufficient, if not necessary, condition to check their e2e schedulability.

We propose DISTRO, a baseband network design that combines the tree structure with real-time scheduling. Fig. 4 illustrates the proposed design in a deployment of heterogeneous radios. The radios are connected to edge switches, and the links from the edge switches are aggregated at multiple levels till the root switch. The destination is assumed to be located at the root switch, which is the common aggregation point for the baseband packets. The traffic destination is assumed to be a physical point that is connected to a pool of baseband processors.

DISTRO’s design accommodates an increasing number of radios without severely affecting their delay performance, and makes the schedulability flexible to the addition and removal of radios. This is achieved by partitioning the baseband network into two components: edge- and aggregation-switch network. The edge switch network contains the edge switches that form the first entry point of a baseband flow. From a deployment standpoint, each edge switch could connect a group of radios that are in physical proximity of each other, for instance, a basestation site in a cellular network. The aggregation network contains all the remaining switches except the edge switches, and its purpose is to aggregate traffic from edge towards the destination.

B. Design Requirements

The aggregation network in DISTRO is a logical tree of links and switches. To simplify the schedulability analysis, we place the following restrictions on its design.

1) Fat-Tree. A tree is a Fat-Tree if for every switch in the tree, the switch’s uplink capacity is greater than or equal to the sum capacity of the incoming links.

2) Symmetric. A tree is symmetric if for every switch, interchanging the incoming links yields the same tree.

3) Non-preemptive. The network switches always use a non-preemptive scheduling policy, i.e., an ongoing packet transmission is never preempted.

4) Equal packet sizes. The baseband packet sizes in the network are always equal irrespective of their source.

C. End-to-End Guarantees

As noted earlier, a packet can experience queuing delay at any of the switches in the network. Beyond the edge switch, characterizing the queuing behavior becomes difficult as the packet arrivals are no longer periodic [7]. However, under our symmetric Fat-Tree construction, we can easily bound the maximum queuing time.

Assume $K$ edge switches, and let $S_i, i = 1, 2, \ldots, K$, denote the set of radios connected to edge switch $k$ where $S_k \subseteq \{1, 2, \ldots, n\}$. For simplicity, assume at time $t = 0$ packets from radios of each edge switch are queued at the switch, and the packets arrive periodically at the switch thereafter. Further, assume no transmission or propagation delay from the radio to the edge switch. Let $C_1$ denote the transmission time of a packet on the link connecting the edge switch and the next aggregation switch. Since packet size, $B$, is fixed for the network, every baseband packet has the same transmission time going through the edge switch.

For simplicity, assume a binary aggregation tree. Let the transmission time sequence as a packet traverses from the edge to the destination be represented as $C_1, C_2, \ldots, C_{h+1}$, where $h$ is the height of the aggregation tree. We denote $h = 1$ if there is only one aggregation switch.

From the symmetric Fat-Tree definition, it holds that if $C_j$ is the transmission time on the incoming link, then the transmission time on the aggregation link, $C_{j+1}$, for all $j, 1 \leq j \leq h$, satisfies:

$$C_{j+1} \leq C_j/2.$$  \hspace{1cm} (4)

Let us assume the edge-switch uses a non-preemptive scheduling policy whereas the aggregation switches use FIFO scheduling. Under non-preemptive scheduling, the inter-arrival time of the incoming packets from the edge is equal to or larger than $C_1$. Consider the first aggregation switch: the outgoing transmission time is $C_2(C_2 \leq C_1/2)$. The packets in each of the two incoming links have at least $C_1$ time separation. Therefore, any packet will wait for at most $C_2$ time in the

![Fig. 4: Fat-Tree architecture of DISTRO with heterogeneous radios.](image)

![Fig. 5: Path of baseband packet from the source to the destination.](image)
queue, which occurs when two packets arrive at the same time. This is illustrated in Fig. 6(a) where packet \( c \) arrives at the same instant as packet \( b \), and is blocked by the transmission time of packet \( b \).

Similarly, for every \( j \), packets in the two incoming links with transmission time \( C_j \) and inter-arrival time greater than \( C_j \), will have a maximum queuing delay of \( C_{j+1} \). Therefore, for the path as shown in Fig. 5, the maximum total delay of any baseband packet across the aggregation network is bounded by:

\[
T_a = \sum_{j=2}^{h+1} (t_a + C_j + C_{j+1} + t_p) \tag{5}
\]

\[
= h(t_a + t_p) + \sum_{j=2}^{h+1} 2C_j \tag{6}
\]

\[
\leq h(t_a + t_p) + \sum_{j=2}^{h+1} \frac{C_1}{2^{j-2}} \tag{7}
\]

\[
= h(t_a + t_p) + 2(1 - 2^{-b}C_1) \tag{8}
\]

where we use the inequality \( C_j \leq C_{j-1}/2 \leq \ldots \leq C_1/2^{j-1} \) using Eq. (4).

Therefore, the aggregation delay bound obtained is independent of the arrivals at in the network. Now, in order for a baseband packet of radio \( i \) to meet its e2e delay bound \( D_i \), the total packet delay at its edge switch must be less than \( D_i \) less the aggregation delay bound. Adding the switching and propagation delays of the edge switch, we arrive at the e2e schedulability of all baseband flows in the network by checking the schedulability of baseband traffic at each edge switch.

**Theorem 1.** In a q-ary symmetric fat-aggregation-tree of height \( h \), the radio traffic \( \tau_i = (T_i, D_i), i = 1, 2, \ldots, n \), is schedulable, if for every \( k, 1 \leq k \leq K \), the set of traffic flows \( \tau_i = (T_i, d'_i), \forall i \in S_k \), with transmission time \( C_1 \) and no preemption, is schedulable at edge switch \( k \), where \( d'_i = D_i - \frac{1}{q^k}C_1 - (h+1)(t_a + t_p) \).

**Proof.** By extension of the binary tree analysis to q-ary tree and noting that the maximum queuing delay of a packet on an uplink link of a switch with \( q \) incoming links, each with transmission time \( C_j \), is \((q - 1)C_j\).

Theorem 1 provides only a sufficient condition for schedulability, but does not affirmatively tell us if a given set of traffic flows are schedulable. Nevertheless, the result is still useful as it allows us to construct a transport network that guarantees to meet the requested e2e delay bounds. Furthermore, it gives the scheduling policies that the network must implement: non-preemptive scheduling at edge switches, and regular FIFO scheduling at aggregation switches. Each edge switch implements a non-preemptive scheduler which can either be EDF or fixed-priority. The fixed-priority scheduler is easy to realize with multiple outbound queues at a switch. Each incoming packet is first classified and placed in its corresponding queue, and the queues are then dequeued in the order of their priority.

The reason for restricting our construction to a symmetric Fat-Tree and non-preemptive scheduling is simple: the packet transmission (tx) times are equal for the incoming links, and therefore, the maximum queuing delay in the aggregation network is easy to obtain. This is not the case, however, if we assume unequal transmission times, which happens if unequal links are used or when preemption is allowed. For instance, Fig. 6(b) shows the blocking of a smaller packet, \( c \), due to the transmission of earlier packets. This example is akin to the head-of-line (HOL) blocking problem that occurs when subsequent transmissions are held up by the first transmission [20]. Then, to arrive at the maximum queuing time in the network, one must examine different arrival sequences and their transmission times at each switch, which can quickly become intractable.

**D. Scalability**

The design of DISTRO is scalable on three fronts. First, as the number of radios in the network, \( n \), grows, and the number of edge switches, \( K \), increases, the maximum total delay of a packet increases only logarithmically. To see this, one can upper bound the total aggregation delay in Eq. (8) for a binary tree as follows:

\[
T_a < h(t_a + t_p) + 2C_1. \tag{9}
\]

The delay bound grows linearly with the height, \( h \), which is always less than or equal to \( \lceil \log_2 K \rceil \). For large packets (e.g., Jumbo Ethernet frames), the transmission time is much larger than the switching and propagation time. In this case, the delay scaling factor, \( t_a + t_p \), becomes even less significant.

Second, DISTRO supports heterogeneous baseband radios with differing periods and deadlines. There is no restriction on the type of radio protocol. This aids the deployment of a radio access network over multiple wireless standards, such as LTE and WiFi, at the same physical location.

Finally, each edge switch, \( k \), implements a schedulability test locally that is dependent only on the traffic generated from the set of radios, \( S_k \), connected to it. Any addition or removal of radios requires one to check only local schedulability, without disturbing the performance of other flows. This feature enables incremental deployment of the baseband network.

**E. Run-time Scheduling**

The aggregation network in DISTRO implements the default FIFO scheduling. On the other hand, each edge switch implements a non-preemptive scheduler which can either be
We implement and evaluate DISTRO’s Fat-Tree architecture using NS-3 [21]. NS–3 is a discrete-event network simulator that accurately simulates network traffic in large deployments. Table I shows the simulation parameters used in our setup. Each edge switch connects 4 heterogeneous radios that have different flow rates (1, 1.5, 2, and 2.5G) but fixed packet size of 1000 bytes. We simulate a fat-aggregation-tree with three levels: 1) a core switch that is connected to the destination through 200Gbps link; 2) \( q \) aggregation switches connected to core switch with 40Gbps links; 3) \( q \) edge switches connected to each aggregation switch with 10Gbps links. Thus, the total number of radios in our setup is \( 4q^2 \).

| Index (1G Flow) | Index (1.5G Flow) | Index (2G Flow) | Index (2.5G Flow) |
|-----------------|-------------------|-----------------|-------------------|
| 0               | 1                 | 2               | 3                 |

**F. Evaluation**

We use the NS–3 packet tagging mechanism to tag each baseband packet with a priority level. The priority levels are chosen to achieve e2e schedulability (Sec. III-C). We then implement a packet-classifier at the edge switch that classifies the incoming packet and places it in one of the 4 egress queues. For dequeuing, the queues are searched in decreasing order of their priority. For simplicity, we assume the transport delay-bound is same as the inter-arrival period of each flow. Under this assumption, the priorities are determined by the inverse of the period (equivalent to a rate-monotonic scheduler).

Fig. 7 shows our system model where samples received by radios over a wireless channel are transported using a single aggregation switch. Let \( x \in \mathbb{C}^{m \times 1} \) be the signal vector sent...
by the transmitter, and let \( y = C^n x^1 \) be the received signal vector, where \( m \) is the number of transmit antennas, and \( n \) is the number of radios. Let \( H \in \mathbb{C}^{n \times m} \) represent the wireless channel between the transmitter and the radios.

**Quantization.** Assume radio \( i \) has ADC quantization width, \( Q_i \), that takes an integer value from the set \( \mathcal{L} = \{ L_1, \ldots, L_d \} \), where \( L_1 \) and \( L_d \) are the minimum and the maximum quantization widths, respectively. Further, let \( Q = [Q_1, \ldots, Q_n] \), and let \( \gamma(Q) \) be the average quantization noise power injected into the baseband samples, where \( \gamma(\cdot) \) is the quantization noise function.

**Transport.** Assume a fixed size \( B \) of baseband packets. From Eq. (1) and Eq. (2), the inter-arrival time at radio \( i \) is \( T_i = \frac{2x_i}{\gamma_i} \), which is a function of the radio quantization width, \( Q_i \).

**Model.** Assuming a narrow-band channel, the received signal can be expressed as:

\[
y = Hx + z + z_Q,
\]
where \( E[xx^H] = \rho I_m \), \( \rho \) denotes the transmitted power that is equal across all symbols, \( z \sim \mathcal{CN}(0, \sigma^2 I_n) \) is the additive complex white Gaussian noise, and \( z_Q \) represents the quantization noise vector. We assume the effect of quantization manifests through the additive term \( z_Q \), which is approximated as a zero-mean complex Gaussian noise with covariance matrix \( \Sigma_Q = \text{diag}(\gamma(Q_1), \ldots, \gamma(Q_n)) \). Let \( \Sigma = \sigma^2 I_n + \Sigma_Q \) denote the equivalent noise covariance matrix.

The ergodic wireless capacity (in b/s/Hz) for a fixed \( Q \), with imperfect channel knowledge at the transmitter, is [23, Eq. 20]:

\[
R(\mathbf{Q}) = E[H]\log_2 \det(\mathbf{I} + \rho \Sigma^{-1} HH^H)]
\]
where the expectation \( E[H] \) is taken over all channel realizations of \( H \).

**A. Problem Formulation**

We want to select the quantization, \( \mathbf{Q} \), that ensures e2e schedulability, and also maximizes the wireless capacity. This is expressed through an optimization problem (OP):

**OP:** \[ \max_{\mathbf{Q} \in \mathcal{L}^n} R(\mathbf{Q}) \]

\[ \text{s.t. } \tau_i(\mathbf{Q}) = (T_i, D_i), i = 1, \ldots, n, \] is schedulable

where \( D_i \) is the transport delay bound of radio \( i \). Next, we show the following properties of our optimization problem.

**Proposition 1.** If \( \gamma(\cdot) \) is monotonically decreasing, \( R(\mathbf{Q}) \) is monotonically increasing in \( \mathbf{Q} \).

**Proof.** Let \( \mathbf{Z} = \mathbf{I} + \rho \Sigma^{-1} \mathbf{HH}^H \), \( \Sigma^{-1} = \text{diag}(\mu_1, \ldots, \mu_n) \), where \( \mu_i = 1/(\sigma^2 + \gamma_i) \) for \( i = 1, \ldots, n \), and let \( \lambda_1, \ldots, \lambda_n \) denote the eigen values of \( \mathbf{Z} \) such that \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \). We follow an approach similar to [23, Theorem 10.3] for the proof. For a fixed \( i \), we claim that \( \lambda_j \), for every \( j \), is monotonically increasing in \( \mu_i \). On the contrary, suppose this is not true and there exists an interval of \( \mu_i \), and \( k, k \in \{1, \ldots, n\} \), such that \( \lambda_k \) increases and decreases in that interval. Consequently there exists \( \lambda' \) such that \( (\lambda' - \lambda_k) \) vanishes for at least two values of \( \mu_i \). Since \( \det(\mathbf{Z} - \mathbf{Z}) = \prod_{j=1}^n (\lambda - \lambda_j) \), then, for \( \lambda = \lambda' \), \( \det(\mathbf{Z} - \mathbf{Z}) \) vanishes for more than one value of \( \mu_i \), which is impossible, since \( \det(\mathbf{Z} - \mathbf{Z}) \) is a linear polynomial in \( \mu_i \). Therefore if \( \gamma(Q_i) \) strictly decreases with \( Q_i \), \( \mu_i \) strictly increases with \( Q_i \), thus, \( \forall j, \lambda_j \) monotonically increases with \( Q_i \), for any \( i = 1, \ldots, n \). Since \( \log_2 \det(\mathbf{Z}) = \sum_{j=1}^n \log_2 \lambda_j \), and expectation is a monotonic operator, therefore, \( R(\mathbf{Q}) \) is monotonically increasing in \( \mathbf{Q} \).

**Proposition 2.** For quantization \( \mathbf{Q} \), if the set of traffic flows \( \tau_i = (T_i, D_i), i = 1, \ldots, n \), is schedulable with the sufficient conditions of scheduling given in Theorems A.1–A.2, then the traffic flows are also schedulable by decreasing \( \mathbf{Q} \).

**Proof.** For simplicity, we consider a non-preemptive EDF scheduler and show that the proposition holds. For any \( i \), changing \( Q_i \) to \( Q_i' \) increases the inter-arrival period from \( T_i \) to \( T_i' \). Since \( T_i < T_i' \) implies \( [(t - d_i)/T_i]^+ \leq [(t - d_i)/T_i']^+ \), \( \forall t \), substituting in Eq. (12), it follows that \( \tau_i(\mathbf{Q}'') = (T_i', D_i') \), \( i = 1, \ldots, n \), is schedulable if \( \tau_i(\mathbf{Q}) = (T_i, D_i), i = 1, \ldots, n \), is schedulable.

**B. Search Algorithm**

The search space in OP is \( \mathcal{L}^n \), hence, finding the optimal \( \mathbf{Q} \) through a brute-force search has an exponential complexity. Note that we cannot relax the integral constraint because of the discreteness of the schedulability. However, from the monotonic dependence on \( \mathbf{Q} \) (Propositions 1, 2), we can construct a greedy search algorithm.

The idea is to use breadth-first search (BFS) on the enumeration of the search space. Starting from the highest quantization, \( [L_d, L_d, \ldots, L_d] \), we enumerate the next highest quantizations, \( [L_{d-1}, L_d, \ldots, L_d], \ldots, [L_{d-1}, L_d, \ldots, L_{d-1}] \), and then enumerate the next highest quantization for each of them, and so on. More generally, let us define the function \( \text{enum}(\cdot) \), for quantization, \( \mathbf{Q} = [L_{k_1}, L_{k_2}, \ldots, L_{k_n}] \), as follows:

\[
\text{enum}([L_{k_1}, L_{k_2}, \ldots, L_{k_n}]) = ([L_{k_1-1}, L_{k_2}, \ldots, L_{k_n}],
\]
\[
[L_{k_1}, L_{k_2-1}, \ldots, L_{k_n}],
\]
\[
\ldots, [L_{k_1}, L_{k_2}, \ldots, L_{k_{n-1}}]).
\]

As we enumerate each possible quantization, we check its schedulability (according to Theorems A.1–A.2). If it is schedulable, we calculate the corresponding capacity, but do not enumerate its further. Finally, from the calculated capac-
Algorithm 1 BFS search

1: Initialize: $Q^* \leftarrow \phi$, $C^* \leftarrow 0$, $U \leftarrow \text{Queue}()$
2: Returns: $C^*$, $Q^*$, optimal capacity and quantization
3: $U.push([L_2, \ldots, L_d])$
4: while $U$ is not empty do $\triangleright$ breadth-first traversal
5: $Q' \leftarrow U.pop()$
6: if $Q'$ is schedulable then $\triangleright$ check e2e schedulability
7: if $R(Q') > C^*$ then $\triangleright$
8: $C^* \leftarrow R(Q')$, $Q^* \leftarrow Q'$
9: end if
10: else
11: for $T$ in $\text{enum}(Q')$ do $\triangleright$ next highest quantization
12: $U.push(T)$
13: end for
14: end if
15: end while

TABLE II: Simulation parameters for capacity evaluation

| $B$ | 1 KB   | $L$ | $\{2, 4, 8\}$ |
|-----|--------|-----|--------------|
| $\rho$ | 1      | $\sigma^2$ | 1          |
| $m$ | 2      | $f_i$ channel rayleigh |
| $\gamma(x)$ | $10^{\frac{2\sigma^2}{x^2} - 2^x}$ | 20MHz |

Fig. 10: Wireless capacity as the aggregation link capacity is varied.

for all $i = 1, \ldots, n$. We use a rate monotonic scheduler as the fixed priority scheduler.

Fig. 10 shows the maximum sum wireless rate averaged over 1000 channel realizations. Observe that as the aggregation link capacity increases, the transmission times decreases, and thus the schedulability supports higher quantization rates resulting in increased capacity. For example, when $n = 8$, the rate increases (from 2-bit to 4-bit quantization) at 14Gbps. Also, as we increase the number of radios from 8 to 32, we require at least 30Gbps, up from 14Gbps, link rate for the the baseband traffic with 4-bit quantization to meet their delay bounds.

V. RELATED WORK

The concept of a Fat-Tree topology originated from the work on non-blocking switch networks. The topology is now widely used in datacenters for building scalable transport networks [10].

A large-scale MU-MIMO was realized using a distributed architecture of servers in [19]. The authors claim that bandwidth needed for baseband transport is not an issue as modern switches support up to 40Gbps links. However, they do not consider the real-time guarantees of transporting baseband samples. ARGOS [4] is another practical multi-antenna setup that suggests daisy-chained radios with Tree-based aggregation. However, no real-time analysis was provided there.

To eliminate jitter, scheduled Ethernet using global scheduling has been proposed for fronthaul transport in a C-RAN [24]. On the other hand, the authors of [11], [12] studied the compression of baseband signals. They propose quantization schemes for lossy compression of the baseband samples. In [11], the authors propose prioritization of baseband frames, though, the priorities are not based on packet delays.

Real-time transport of Ethernet packets is a well-known problem in real-time systems literature [7], [8], [16]. While [16] provides a general approach for schedulability, the authors propose the use of flow regulators at each switch, which is difficult to realize in practice.

VI. CONCLUSION

In this paper, we have designed a baseband transport network based on Fat-Tree topology that is scalable to a large number of radios while meeting the e2e delay bounds. We have provided sufficient criteria for e2e schedulability, which
is validated via simulations. While we only consider one aggregation link per switch, the design can be generalized to multiple aggregation links and cross-links for improved fault-tolerance. The only requirement is that one should be able to bound the queuing time at each switch.

We also characterize the wireless capacity with the schedulability constraint, and provide an efficient search algorithm to maximize the capacity. Overall, our design can enable processing of wireless signals away from hardware while ensuring no performance loss from the underlying transport network.

VII. APPENDIX

Theorem A.1. Under deadline scheduling, a set of traffic flows \( \tau_i = (T_i, d_i), i = 1, 2, \ldots, n \), is schedulable on a link with transmission time \( C \), and preemption if and only if [13, Theorem 6]:

\[
\forall t > 0, \quad \frac{C}{t} \left( \sum_{i=1}^{n} \left( \frac{d_i}{T_i} \right) \right) \leq 1,
\]

and without preemption if and only if [13, Theorem 6]:

\[
\forall t \geq d_{\text{min}}, \quad \frac{C}{t} \left( 1 + \sum_{i=1}^{n} \left( \frac{d_i}{T_i} \right) \right) \leq 1
\]

where \( d_{\text{min}} = \min\{d_i : 1 \leq i \leq n\} \) and the function \([x]^+ = \max(0, [x])\).

Theorem A.2. Under fixed-priority scheduling, a set of traffic flows \( \tau_i = (T_i, d_i), i = 1, 2, \ldots, n \), and priorities \( \pi_i, i = 1, 2, \ldots, n \), such that \( \pi_1 \geq \pi_2 \geq \cdots \geq \pi_n \), is schedulable on a link with transmission time \( C \), if the function \( W_m(k, x) \) satisfies:

\[
\max_{1 \leq m \leq n, k \leq N_m} W_m(k, (k-1)T_m + d_m) \leq 1
\]

where \( N_m = \min\{k : W_m(k, kT_m) \leq 1\} \), and function \( W_m(k, x) \), for preemption, is defined as [14]:

\[
W_m(k, x) = \frac{C}{T_m} \left( k + \sum_{i=1}^{m-1} \left( \frac{d_i}{T_i} \right) \right),
\]

and without preemption, \( W_m(k, x) \) is defined as [25]:

\[
W_m(k, x) = \frac{C}{T_m} \left( k + 1 + \sum_{i=1}^{m-1} \left( 1 + \frac{d_i}{T_i} \right) \right)
\]

REFERENCES

[1] “C-RAN: The road towards green RAN,” http://labs.chinamobile.com/ cran/wp-content/uploads/2014/06/20140613-C-RAN-WP-3.0.pdf. [Online; accessed 29-Nov-2015].

[2] A. Checko, H. Christiansen, Y. Yan, L. Sciliar, G. Kardaras, M. Berger, and L. Dittmann, “Cloud ran for mobile networks – a technology overview,” Communications Surveys Tutorials, IEEE, vol. 17, no. 1, pp. 405–426, 2015.

[3] F. Rusek, D. Persson, B. K. Lau, E. Larsson, T. Marzetta, O. Edfors, and F. Tufvesson, “Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays,” IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 40–60, Jan 2013.

[4] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, “Argos: practical many-antenna base stations,” in Proc. of Mobicom, 2012.

[5] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, “Sora: High performance software radio using general purpose multi-core processors,” in Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, 2009, pp. 75–90.

[6] N. Niakein, “Processing radio access network functions in the cloud: Critical issues and modeling,” in Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services, 2015.

[7] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks,” Proceedings of the IEEE, vol. 83, no. 10, pp. 1374–1396, 1995.

[8] C. M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne, “Real-time communication in packet-switched networks,” Proceedings of the IEEE, vol. 82, no. 1, pp. 122–139, 1994.

[9] F. Kaltenberger, R. Knopp, N. Niakein, D. Nussbaurn, L. Gauthier, and C. Bonnet, “OpenAirInterface: Open-source software radio solutions for 5G,” in Proc. of EUCNC, 2015.

[10] M. Al-Fares, A. Loukissas, and A. Vahdat, “A Scalable, Commodity Data Center Network Architecture,” in Proceedings of the ACM SIGCOMM, 2008.

[11] E. Chai, K. Shin, S. Lee, J. Lee, and R. Etkin, “SPIRO: Turning Elephants into Mice with Efficient RF Transport,” in Proc. of IEEE INFOCOM, 2015.

[12] K. F. Nieman and B. L. Evans, “Time-Domain Compression of Complex-Baseband LTE Signals for Cloud Radio Access Network,” in IEEE SLP, 2013.

[13] Q. Zheng and K. G. Shin, “On the ability of establishing real-time channels in point-to-point packet-switched networks,” IEEE Transactions on Communications, vol. 42, no. 234, pp. 1096–1105, Feb 1994.

[14] J. P. Lehoczky, “Fixed priority scheduling of periodic task sets with arbitrary deadlines,” in Proceedings of Real-Time Systems Symposium, 1990.

[15] “Universal Software Radio Peripheral,” http://ettus.com/.

[16] D. D. Kandlur, K. G. Shin, and D. Ferrari, “Real-time Communication in Multi-hop Networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 10, pp. 1044–1056, 1994.

[17] K. W. Tindell, A. Burns, and A. J. Wellings, “An Extendible Approach for Analyzing Fixed Priority Hard Real-time Tasks,” Real-Time Systems, vol. 6, no. 2, pp. 133–151, 1994.

[18] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Std. 802.11ac Draft 3.0, 2012.

[19] Q. Yang, X. Li, H. Yao, J. Fang, K. Tan, W. Hu, J. Zhang, and Y. Zhang, “Raspbroadcast: Enabling scalable real-time signal processing in large memmio systems,” in Proc. of SIGCOMM, 2013.

[20] M. Karol, M. Hluchyj, and S. Morgan, “Input Versus Output Queueing on a Space-Division Packet Switch,” IEEE Transactions on Communications, vol. 35, no. 12, pp. 1347–1356, 1987.

[21] “NS–3,” https://www.nsnam.org/, [Online; accessed 14-Apr-2016].

[22] L. Schumacher, K. L. Pedersen, P. E. Mogensen, N. J. Vej, and D. A. st, “From Antenna Spacings To Theoretical Capacities - Guidelines For Simulating Mimo Systems,” in IEEE PIMRC, 2002.

[23] R. Bhatia, Perturbation Bounds for Matrix Eigenvalues. Society for Industrial and Applied Mathematics, 2007.

[24] “A Performance Study of CPRI over Ethernet,” http://www.ieee1904. org/meetingarchive/20150202f51502ashwoodla.pdf. [Online; accessed 14-Apr-2016].

[25] Y. Wang and M. Sakseha, “Scheduling fixed-priority tasks with preemption threshold,” in International Conference on Real-Time Computing Systems and Applications, 1999.