A Framework for Exploiting Hard and Soft LLRs for Low Complexity Decoding in VRAN Systems

Conference Paper · March 2021

DOI: 10.1109/WCNC49053.2021.9417504

6 authors, including:


Satya kumar Vankayala
Indian Institute of Science Bangalore


Abhay Kumar Sah
Indian Institute of Technology Roorkee


Swaraj Kumar
Samsung Research

A Framework for Exploiting Hard and Soft LLRs for Low Complexity Decoding in VRAN Systems

Satya Kumar Vankayala, RaviTeja Gundeti, K Gautam Shenoy, Abhay Kumar Sah, Swaraj Kumar, Seungil Yoon
Samsung Research Institute, Bangalore, India; IIT Roorkee, India
satyakumar.v@samsung.com, {ravi5gundeti, gautamshenoydmc}@gmail.com, abhaysah@ece.iitr.ac.in,
swaraj.kumar@samsung.com, siyoon72@samsung.com

***Abstract*—Owing to improved coverage and flexibility, the radio access network (RAN) functionalities are being virtualized in a sense that the base station will merely act as a radio unit, and all the baseband processing will occur in the cloud. Therefore, the baseband signal-processing algorithms need to be designed in a way that it can match the latency requirements. In this paper, we address one of the inherent but complex issues in baseband signal processing, namely, the log-likelihood ratio (LLR) computation. In general, soft-decision rules are used for calculating the LLRs, which is computationally expensive. Thus, we attempt to exploit the benefits of hard-decision based LLRs for proposing a framework that uses soft decision only when the received symbols are closed to the decision boundary; otherwise, the framework uses hard decision. This helps us to keep the complexity low while meeting the desirable error performance. These schemes are suitable for incorporation in virtual RAN systems while considering appropriate QoS requirements.**

***Index Terms*—QAM, LLR, BER, VRAN.**

I. Introduction

Radio access network (RAN) has continuously evolved with the advances in mobile communications. In a typical RAN architecture, the radio unit (RU) is attached to a dedicated baseband unit (BBU) through a common public radio interface (CPRI), where RU acts as a front head, and BBU is used for signal processing related to the physical layer. With the advances in RAN, the interplay between RU and BBU is progressive. Starting from a co-located radio and baseband units, the BBUs moved to a centralized location and, more recently, to a cloud. These two developments are respectively referred to centralized and virtual (C/V) RAN [1]. Fig. 1 shows the interconnections between these RANs and BBUs.

CRAN improves the coverage and deployment flexibility while lowering the operational and capital expenditures [2]. VRAN further extends the flexibility by replacing dedicated baseband hardware to general-purpose hardware [1] like x86 processors. This allows running network functions from multiple vendors on the same hardware. In some cases, even the hardware can be shared between the service providers [3]. However, due to isolated RU and BBU, the gains are feasible if the latency requirements can be satisfied.

In 5G, with rapid increase in number of applications the requirement for end-to-end latency is in the order of 1-10ms [4] as shown in Table I. Number of independent data streams and packet sizes in 5G are much higher than 4G systems. Further in 5G the slot duration is a function of numerology, it can be in the order of 0.01ms. Thus, there is a need for faster decoding algorithms which are computationally efficient. In this paper, we attempt to reduce the computational complexity of LLR decoding. Formally, an LLR is a measure that determines the likelihood of a bit being zero or one, which subsequently utilized for channel decoding [5]. The higher the magnitude, greater the confidence. We have illustrated architecture of baseband processing unit in Fig. 2. The LLR is computed by the soft demapper block, are processed in cloud in VRAN.

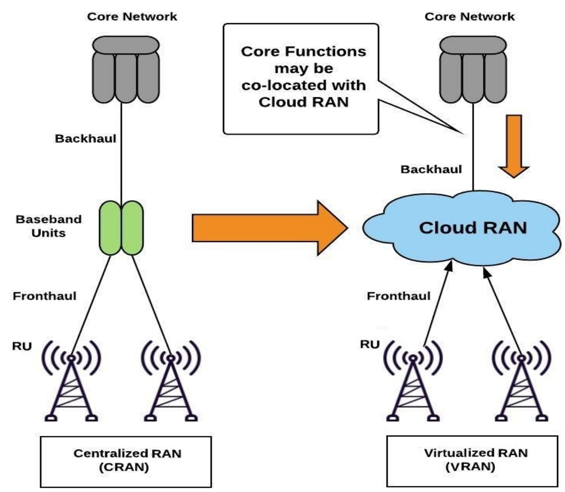

Fig. 1. Illustration of CRAN and VRAN architecture.

TABLE I
LATENCY REQUIREMENTS FOR 5G SERVICES

Resource Type	Packet Latency Budget
GBR	100-500ms
Non-GBR	10-300ms
Delay critical GBR	1-10ms

In the literature, there are primarily two approaches for determining the LLRs [6], [7]. One depends on a hard decision, and the other (and more prominent one) depends on a soft decision rule [8], [9]. The hard decision rule is easy to compute but suffers from severe loss in decoding performance. Whereas the soft decision can yield an optimal performance but at the cost of an exponential complexity. Several attempts have been made to reduce its exponential complexity to a polynomial complexity at a marginal to a considerable loss in decoding performance [10]–[12].

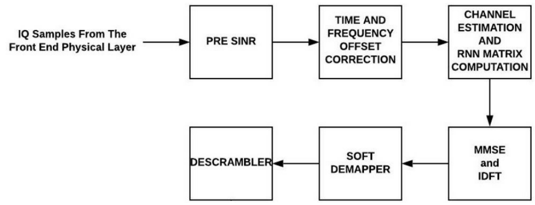

Fig. 2. Base band receiver block diagram.

Despite these efforts, the complexity increases rapidly with the modulation order, and therefore, not suitable for higher modulation order like 256- or 1024- QAM [8], [12]. As the next-generation technologies are likely to use a higher-order modulation, there is a need to revisit this problem. The idea is to use soft LLRs whenever necessary and to use hard LLRs otherwise. Consequently, we present a framework to exploit the realms of hard and soft LLRs such that QoS are met, keeping the computational cost low. For this, we propose a threshold to define the regions of uncertainties for hard decision and soft decision. We compute the soft LLRs only if the received symbol falls into the uncertainty region otherwise, we compute hard LLRs. In this paper, we proposed various hybrid algorithms.

In most of the real-time systems, LLR decoding is done using proprietary algorithm which are implemented in hardware. In 5G systems, different applications have different QoS requirements, using single LLR decoding algorithm for all applications is computationally inefficient. There is a need for programmable algorithms based on applications which are compatible for Cloud based architecture. So we have proposed various algorithms which uses decision rule to combine soft decoding and hard decoding effectively. Proposed algorithms have recursive structure there by making them suitable for hardware implementation also.

This paper is organized as follows: In Section II, we give the notation and the channel model. Section III discusses the proposed algorithms. An error analysis of Algorithms 1 and 2 are provided in Section IV, with certain in appendices. Next, we discuss the performance of aforementioned algorithms in Section V. Finally, we conclude the paper.

II. Preliminaries

In any communication system, the decoder complexity is one of the key factors affecting maximum communication rate. For optimal detection and estimation of signals, the maximum aposteriori (MAP) rule [6], [7] is chosen. If the input symbols are equi-probable, it suffices to evaluate the maximum likelihood (ML) rule. When decoding multiple symbols, the ML computation involves multiplying (respectively adding) a series of likelihood (respectively log-likelihood) functions and then making a decision on which symbol was sent based on the value. In particular, this is the decision rule when performing soft decoding for each bit at the receiver.

The likelihood function (also called likelihood ratio) is a ratio of two probability functions (alternatively probability density functions) that is widely applied in detection/estimation theory [13]. For example, if we are performing hypothesis testing with distributions P and Q with Q being the null hypothesis' distribution, then for any incoming sample x, the LLR is defined as

$$\Lambda(x) := \log\left(\frac{P(x)}{Q(x)}\right). \tag{1}$$

We consider a fast fading channel corrupted with AWGN noise. Denote by $X = X_I + jX_Q$ the channel input where X_I is the in-phase component (I component), X_Q is the quadrature component (Q component) and $j = \sqrt{-1}$. The channel output Y is given by

$$Y = HX + Z \tag{2}$$

where $H = H_I + jH_Q$ is the channel fading coefficient and Z is a circularly symmetric complex Gaussian random variable with noise variance σ^2 (denoted by $\mathcal{CN}(0, \sigma^2)$). Across n channel uses, we will assume that H_i, Z_i, $1 \leq i \leq n$, are independent and identically distributed (i.i.d.). We additionally assume an average input power constraint of P i.e. $\sum_{k=1}^{n} |X_k|^2 \leq nP$ as is usually imposed for fading channels.

We assume that the input signal X is modulated as M-QAM. The minimum distance between symbols is denoted by d_m. Under moderate to high SNR and assuming no fading, the ML rule breaks down to minimum distance decoding (MDD) [6]. While MDD is suboptimal under fading, it has the benefits of faster decoding and robustness to noise.

A. *Canonical encoding for QAM*

While generally speaking, any bit encoding may be used for QAM. That is, a symbol may be assigned any $\log_2 M$ bit codeword as long as no other symbol has this codeword. However, in this paper, we follow the Gray coding scheme [6] which has the property that adjacent codewords only differ by one bit. This has the advantage that if a symbol was erroneously decoded because the received symbol was outside a decision boundary but close to it, it would differ in only one bit. As a consequence, the BER is reduced. We refer to this as the canonical encoding for QAM.

III. LLR Algorithms

In this section, we describe four algorithms for LLR based decoding. The first algorithm, also referred to as the base or centroid algorithm is a well known decoding method. The remaining algorithms are novel and utilize soft decoding. All of these are derived by exploiting the geometry of signal constellations.

A. *Algorithm 1*

Algorithm 1 uses a hard decoding approach using centroids to find the symbol closest to y, where y is the received symbol. This is a recursive algorithm that takes $\log_4(M)$ steps to decode the correct symbol. Refer to Fig. 3 for illustration. The centroids may be pre-computed and stored or even computed on runtime as they are simple to compute; the most complicated function being the square root function. The LLR for a bit that is hard decoded would theoretically

this works for patent Fig 2
but not Fig 5.

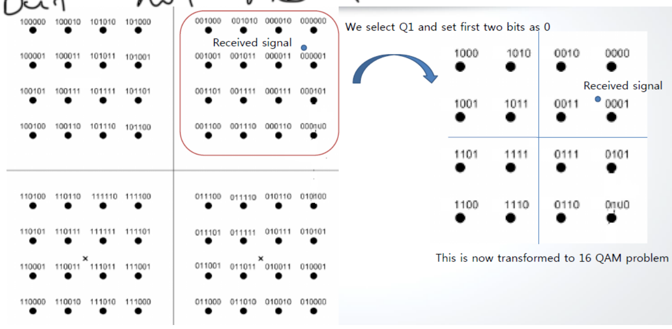

Fig. 3. Illustration of Algorithm 1 for 64 QAM.

Algorithm 1 Hard decoding via the method of centroids. For $M = 4^k,\ k \geq 1$.

1: Given received symbol $\mathbf{y} = y_I + jy_Q$, M-QAM. Write y as a point in 2-D space i.e. (y_I, y_Q).
2: **repeat**
3: Compute the centroids $C_k,\ 1 \leq k \leq 4$ of the 4 quadrants. These are given by

$$C_1 = \left(\sqrt{M}d_m, \sqrt{M}d_m\right),$$
$$C_2 = \left(-\sqrt{M}d_m, \sqrt{M}d_m\right),$$
$$C_3 = \left(-\sqrt{M}d_m, -\sqrt{M}d_m\right),$$
$$C_4 = \left(\sqrt{M}d_m, \sqrt{M}d_m\right). \tag{3}$$

4: Look at the signs of y_I and y_Q. This will identify the quadrant of the received vector. Denote that quadrant by q. Recover the two bits corresponding to the quadrant. The LLR for each bit of the two bits is correspondingly set as $L_{\max}$ if the corresponding bit is decoded as a 0 and $-L_{\max}$ for 1. This is hard decoding.
5: Go to the qth quadrant where $\mathbf{y}$ lies. This is a sub-constellation of the original constellation with $M/4$ points.
6: Update $y \leftarrow y - C_q$ and $M \leftarrow M/4$.
7: **until** $M == 1$

be $\pm\infty$ for $0/1$ respectively. Practically, we assume that $|\text{LLR}_k| \leq L_{\max}$, where $L_{\max}$ is a large number.

An issue with Algorithm 1 is that hard decoding may not accurately decode the received points when they land on the boundaries. These issues are mitigated by the use of soft decoding rules which do not operate on sharp boundaries but rather on *soft* boundaries. LLRs are used for this purpose. While soft decoding is more complex than hard decoding, it is superior in terms of performance.

B. Algorithm 2

Each boundary is now expanded by a width of ε where $0 < \varepsilon << d_m/2$. Refer to Fig. 4. ε is chosen proportional to the noise variance N_0. In Algorithm 2, at each stage we check if the received vector y lies in the expanded boundary (which is henceforth referred to as the *red region*). If it does, we perform soft decoding for all bits otherwise we proceed

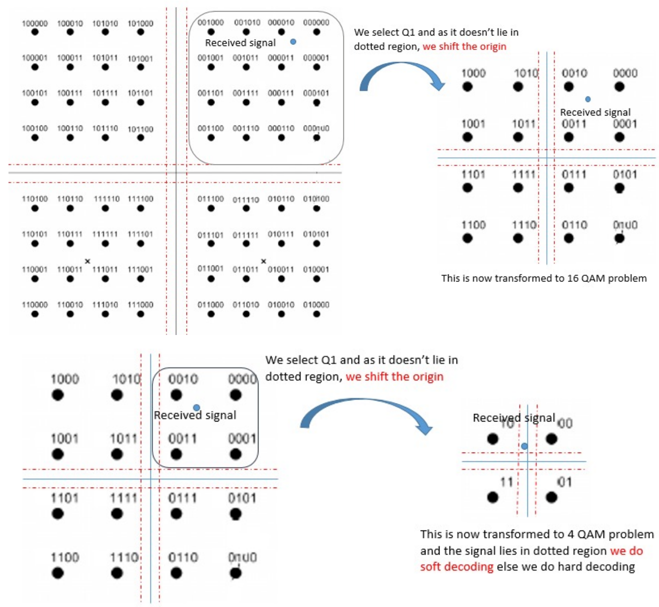

Fig. 4. Illustration of Algorithm 2 for 64QAM

with Algorithm 1. In Fig. 4 we have provided an example for illustration of Algorithm 2. In this example, at first stage we see that the received symbol does not lie in the red region, so we shift to first quadrant and move to second stage. In the next second stage too, we see that the received symbol does not lie in the red region, so we move to third stage. In the third stage, we see that the received symbol lies in the red region, hence we perform soft decoding for the received symbol.

Below, we propose two variants of Algorithm 2, namely Algorithm 3 and 4.

1) Algorithm 3: In Algorithm 3, at any stage, if the received symbol falls in the red region, we proceed to do soft decoding for all the remaining bits, else we do hard decoding for those two bits. This is performed until all bits are decoded. For example in Fig. 4, as the received symbol falls in red region in third stage, we do soft decoding from there on and hard decoding for first four bits.

2) Algorithm 4: In Algorithm 4, at any stage, if the received symbol falls in the red region, we proceed to do soft decoding for those two bits, else we do hard decoding for those two bits. This is performed until all bits are decoded. For example in Fig. 4, as the received symbol falls in red region in third stage, we do soft decoding for only those two bits and hard decoding for remaining bits.

IV. Error Analysis

In this section, we analyze the probability of error of the three provided algorithms. We do not take errors due to finite precision into account. Note that in any QAM $(M = 4^n)$,

$$N_C = 4,\ N_E = 4(\sqrt{M} - 2),\ N_I = (\sqrt{M} - 2)^2 \tag{4}$$

where N_C is the number of symbols on the constellation corners, N_E along the edges excluding corners and N_I are the points in the interior.

In QAM, the symbols on the corners, edges and interior of the constellation have different detection probabilities. Under basic model and canonical encoding, the probability of error expressions for direct hard decoding [6] are given as

$$p_e^C = 1 - \Phi^2\left(\frac{\sqrt{2}d_{\min}}{\sigma}\right) \quad (5)$$

$$p_e^E = 1 - \left(2\Phi\left(\frac{\sqrt{2}d_{\min}}{\sigma}\right) - 1\right)\Phi\left(\frac{\sqrt{2}d_{\min}}{\sigma}\right) \quad (6)$$

$$p_e^I = 1 - \left(2\Phi\left(\frac{\sqrt{2}d_{\min}}{\sigma}\right) - 1\right)^2 \quad (7)$$

where p_e^C, p_e^E and p_e^I are the respective probability of errors for points lying at the corner, edge and interior of the constellation; $d_{\min}$ is the minimum distance between two points and $\Phi(x) := \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-u^2/2\right) du$, is the cumulative distribution function (cdf) of a standard Gaussian distribution. For equiprobable symbols, the average probability of error hence would be

$$p_e = \frac{N_C p_e^C + N_E p_e^E + N_I p_e^I}{M}. \quad (8)$$

These expressions will be used for comparison purposes.

A. Soft decoding

In order to evaluate the effects of soft decoding, we need to compute the probability of a point falling in the red decision region. This probability will depend on which point was actually transmitted. Note that we do not have this issue in hard decoding as the measure on the boundaries is zero.

Proposition 1. *Denote by $\gamma_\varepsilon(x) := \Phi(x+\varepsilon) - \Phi(x-\varepsilon)$. For a M-QAM, suppose point p in the constellation has R_p points to its right and U_p points above it (excluding itself), then assuming only basic conditions, the probability that the received vector falls in the red region given that p was transmitted is given by*

$$\begin{aligned} P_r(p; M) = \zeta_\varepsilon^M(d_m, R_p) + \zeta_\varepsilon^M(d_m, U_p) \\ -\zeta_\varepsilon^M(d_m, R_p)\zeta_\varepsilon^M(d_m, U_p) \end{aligned} \quad (9)$$

where

$$\zeta_\varepsilon^M(d_m, m) = \sum_{j=m-\sqrt{M}+2}^{m} \gamma_\varepsilon\left(\frac{\sqrt{2}d_m(2j-1)}{\sigma}\right). \quad (10)$$

Moreover if p is of the form $((2k-1)d_m/2, (2l-1)d_m/2)$ for some $k, l \in \mathbb{Z}$, then

$$R_p = \frac{\sqrt{M}}{2} - k, \; U_p = \frac{\sqrt{M}}{2} - l. \quad (11)$$

Proof. Refer Appendix A. □

While, technically, we may opt for any good soft decoding rule, for the purposes of error comparison, the scheme described in Appendix B is considered. This is a suboptimal but relatively fast algorithm for soft decoding.

B. Proposed Algorithms Analysis

Let $p_e(x; M)$ denote the probability of error when symbol x is transmitted in M-QAM. Using conditional probability, we obtain

$$\begin{aligned} p_e(x; M) &= p_e(x; M | y \in \text{Red}) P_r(x; M) \\ &\quad + p_e(x; M | y \notin \text{Red})(1 - P_r(x; M)) \quad (12) \\ &= p_e(x; M, \text{Soft}) P_r(x; M) \\ &\quad + p_e(x; M, \text{Hard})(1 - P_r(x; M)), \quad (13) \end{aligned}$$

where $\{y \in \text{Red}\}$ is the event that the received vector y lies in the red region, $p_e(x; M, \text{Hard})$ and $p_e(x; M, \text{Soft})$ are probabilities of error for symbol x while employing hard and soft decoding respectively and $P_r(x; M)$ is defined in (9). In this case, depending on the symbol, equations (5), (6) and (7) are equal to $p_e(x; M, \text{Hard})$. A sketch of the derivation for $p_e(x; M, \text{Soft})$ is provided in Appendix C.

It can be seen that the probability of error for Algorithm 2 is a convex combination of the cases where we use only hard or soft decoding. Consequently, it's performance, i.e. probability of error, will be between hard and soft decoding. This will be illustrated in Section V.

The error analysis of algorithms 3 and 4 will be upper bounded by (13). This is primarily due to the fact that by doing the decoding in steps as opposed to in one shot, the error profile can only improve.

The red region thickness, ε, may be obtained using the error formulae provided. Given the target probability of error, we numerically invert the formula and find the value of ε that would achieve the desired performance.

V. Simulation Results

We consider a Turbo coded system with Rayleigh fading channel and examine the performance of proposed algorithms for two different values of M, i.e, 256 QAM and 1024 QAM modulation schemes. The common simulation parameters used for our simulation are described in Table II and the corresponding results have been outlined in Fig. 5 and 6.

TABLE II
System Parameters

Code rate	5/27
E_b/N_0 range	0dB-40dB
Fading	Rayleigh with parameter 1
Subcarrier spacing	15 kHz
System Bandwidth	20 MHz

TABLE III
Algorithm 2, 3 and 4 results for 256 QAM

Loss	0.2 dB	1 dB	2 dB
Computational Gain	90% Soft	80 % Soft	70 % Soft
	10 % Hard	20 % Hard	30 % Hard

In Fig. 5 and 6, we compare the performance of the proposed algorithms with the centroid method (Algorithm 1), Soft decoding, TOBI [11] and Min-sum (Appendix B) methods. We see that the Algorithm 2, 3 and 4 improves the BER performance compared to Algorithm 1, TOBI and

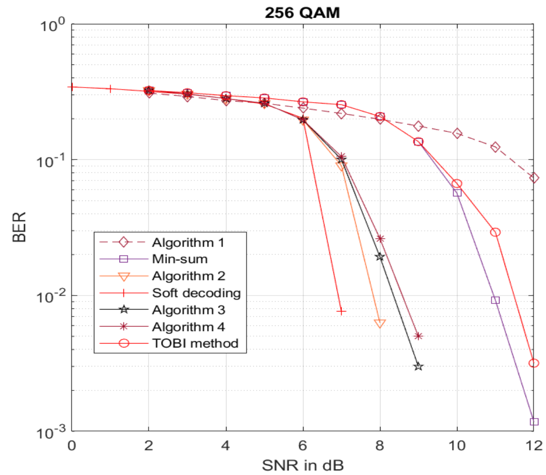

Fig. 5. BER comparison for Turbo coded system with 256 QAM.

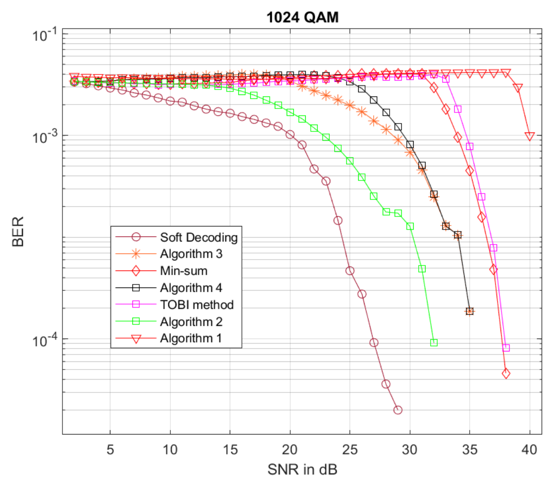

Fig. 6. BER comparison for Turbo coded system with 1024 QAM.

Min-sum algorithms. Thus, Algorithms 2, 3 and 4 are good at balancing the tradeoff between computation complexity and BER. Computational complexity gains for 256 QAM are listed in Table III. Note that all error curves are between Soft decoding and hard decoding. This agrees with our discussion in Section IV-B. At a fixed BER, we see that proposed Algorithm 2 has 4dB gain and 7 dB gain for 256 QAM and 1024 QAM respectively w.r.t. TOBI and Min-sum algorithms. We see that with increase in constellation size, gains are increasing when compared to existing algorithms.

From Table III, we see that for 256 QAM same BER at the loss of 1-2dB there is computational gain of 20-30%. Due to this LLRs are computed faster thus reducing time for decoding. This results in reduction of latency. As we move to higher and higher modulation schemes, the computational complexity of soft decoding increases which results in increase in decoding time. Hence the above computational gain using proposed algorithms can be effective.

VI. Conclusion

We have described and illustrated three algorithms for fast LLR decision making. We have provided the probability of

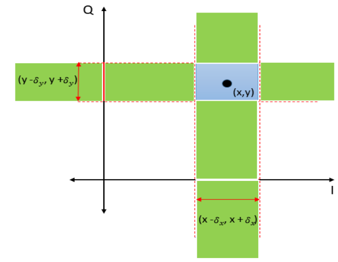

Fig. 7. Measuring the red region for a M-QAM point.

error analysis for Algorithm 2. As mentioned, these are easy to compute and do not involve computation of complicated functions. We observed that Algorithm 2 was outperforming most of the proposed methods. These schemes can be integrated in 5G cloud based VRAN based on appropriate QoS requirements, which include latency, packet error rate and throughput.

Appendix A
Proof of Proposition 1

Let p be the point under consideration. Since error is distributed as $\mathcal{CN}(0, \sigma^2)$, this is equivalent to a product of Gaussian measures, each measure along the I and Q axis respectively but with half the variance. Using measure theoretic ideas from [14], we note the following. If $A \subset \mathcal{X}$, $B \subset \mathcal{Y}$, measurable, and P_X and P_Y are probability measures on $\mathcal{X}$ and $\mathcal{Y}$ respectively. Then $(P_X \times P_Y)(A \times B) = P_X(A)P_Y(B)$.

Let (x, y) be an arbitrary M-QAM point. The red region, part of which is shown in Fig. 7, consists of 3 types of rectangles. In our case, $\delta_x = \delta_y = \varepsilon$. Type 1 first projects an interval $(x - \varepsilon, x + \varepsilon)$ on the I axis, is unbounded in the Q axis and is of the form $A \times \mathcal{Y}$. Type 2 is the same as above but with I and Q interchanged and is of the form $\mathcal{X} \times B$. The last one, type 3, is an intersection of the previous two, and of the form $A \times B$. In our case, we have $\mathcal{X} = \mathcal{Y} = \mathbb{R}$. While the Gaussian measures are shifted, we can, without loss of generality, assume that p is 0, and work with centralized Gaussian measures.

Applying the respective Gaussian measures, for a rectangle of type 1, which is at a distance x from origin, the measure is equal to $\gamma_\varepsilon(\sqrt{2}x/\sigma)$. The same holds for type 2. Type 3, which is centered at (x, y) as shown in Fig. 7, will have measure $\gamma_\varepsilon(\sqrt{2}x/\sigma)\gamma_\varepsilon(\sqrt{2}y/\sigma)$.

Suppose the origin has R_0 points to the right and U_0 points above it. Then there are R_0 type 1 rectangles to the right, $\sqrt{M} - R_0 + 1$ type 1 rectangles to the left, U_0 type 2 rectangles above, $\sqrt{M} - R_0 + 1$ type 2 rectangles below. The measure of the red region can now be computed as the sum of measure of the type 1 and type 2 rectangles minus the measure of type 3 rectangles to avoid over-measuring. We see that (10) gives the measure of all type 1 rectangles for $m = R_0$ and type 2 for $m = U_0$. The type 3 rectangles measure is nothing but the product of the previous two. This gives us (9); (11) can be verified by inspection.

Appendix B
Soft Demapper

The standard M-QAM constellation for $M = 4^m$, consists of M points with coordinates $(2i-1, 2j-1)$ for $-\frac{\sqrt{M}}{2}+1 \leq i, j \leq \frac{\sqrt{M}}{2}$. To ensure that the constellation, for equiprobable symbols, has average power constraint P, each symbol is scaled by $\alpha_P = \sqrt{\frac{3P}{2(M-1)}}$. The minimum distance, originally $d_m = 2$, also gets similarly scaled to $d_m(P) = \sqrt{\frac{6P}{M-1}}$. The LLR is computed on a bit by bit basis i.e. for M-QAM, you get $\log_2(M)$ LLRs. The LLR for bit i is by definition

$$\text{LLR}_i(y) = \log \frac{P\left[\text{Bit } i = 0|\text{Received symbol } y\right]}{P\left[\text{Bit } i = 1|\text{Received symbol } y\right]} \tag{14}$$

Assuming equiprobable input symbols and denoting $\mathcal{M}$ as the set ofpoints in M-QAM, this may be further simplified to

$$\text{LLR}_i(y) = \log \left[\frac{\sum\limits_{x\in\mathcal{M}, b_i(x)=0} W(y|x)}{\sum\limits_{x\in\mathcal{M}, b_i(x)=1} W(y|x)}\right] \tag{15}$$

where $b_i(x)$ is the i^{th} bit of the binary sequence associated with x and $W(y|x)$ is the effective channel density function that for input symbol x and output symbol y, with reference to (2), is given by:

- If the fading coefficients are unknown at receiver, then $W(y|x) = \int_h f_{Y|X,H}(y|x,h)f_H(h)dh$, where $f_{Y|X,H}$ is the fading channel density and f_H is the probability density of fading coefficients. In other words, we are working with an averaged channel.
- If the fading coefficient h is known at the receiver, then $W(y|x) = f_{Y|X,H}(y|x,h)$.

The soft decoding rule is simply to declare bit i as 0 if $LLR_i(y) > 0$ and 1 otherwise. However this has some inherent complexities. For example, with Gaussian noise and fading known at the receiver, $W(y|x) \propto \exp\left(\frac{-|y-hx|^2)}{\sigma^2}\right)$. Hence the numerator and denominator of (15) will be sum of such exponential terms, which usually are too small to be calculated accurately. Moreover, the log function thereafter further complicates things as it tends to amplify very small differences. A well known suboptimal approximation is often deployed at the cost of some accuracy. This exploits the well known *log-sum-exp* bound given by:

$$-\min_k x_k \leq \log\left[\sum_{k=1}^{n}\exp(-x_k)\right] \leq \log n - \min_k x_k. \tag{16}$$

In practice, the expression will be closer to LHS since a randomly generated output symbol will be more likely to have different distances from QAM points. This justifies the approximation of this quantity to LHS, which is referred to as the *min sum* approximation. This is considerably less complex than directly computing exact LLR values.

Appendix C
Algorithm 2: Soft Decoding Error

Let x be the input symbol and Y be the output of the channel as per (2), which we currently assume lies in the red region. In soft decoding, each LLR determines one bit of the binary sequence of the input symbol. As each decision is made independently, the probability of error density when receiver receives symbol $Y = y$ under soft decoding will be given by

$$p_e(x; M, Y = y) = 1 - W(y|x) \prod_{k=1}^{\log M} \tilde{P}_k(x, y) \tag{17}$$

where

$$\tilde{P}_k(x,y) = \begin{cases} Pr(\text{LLR}_k(y) > 0) & \text{if } b_k(x) = 0 \\ Pr(\text{LLR}_k(y) \leq 0) & \text{if } b_k(x) = 1 \end{cases} \tag{18}$$

and $W(y|x)$ is the effective channel as described in Appendix B. We can then express $p_e(x; M, \text{Soft})$ as

$$p_e(x; M, \text{Soft}) = \frac{\int_{y\in\text{Red}} p_e(x; M, y)dy}{P_r(x; M)}. \tag{19}$$

For the case with Gaussian noise with fading coefficient h known at receiver, the suboptimal *min* decoder gives us

$$Pr(\text{LLR}_k(y) > 0) = 1\Bigg(\min_{x:b_k(x)=1} |y - hx|^2 \tag{20}$$

$$- \min_{x:b_k(x)=0} |y - hx|^2 > 0\Bigg) \tag{21}$$

where $1(.)$ is the indicator function. Depending on the assignment of bit sequence to symbols, the above quantities may be computed.

References

[1] M. A. Habibi, M. Nasimi, B. Han, and H. D. Schotten, "A comprehensive survey of ran architectures toward 5g mobile communication system," *IEEE Access*, vol. 7, pp. 70 371–70 421, 2019.

[2] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud ran for mobile networks—a technology overview," *IEEE Communications surveys & tutorials*, vol. 17, no. 1, pp. 405–426, 2014.

[3] M. Begley and M. Riback, "virtualized 5g ran: why, when and how?"

[4] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5g: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, 2017.

[5] S. Shao, P. Hailes, T. Wang, J. Wu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Survey of turbo, ldpc, and polar decoder asic implementations," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2309–2333, 2019.

[6] A. J. Viterbi and J. K. Omura, *Principles of digital communication and coding*. Courier Corporation, 2013.

[7] R. G. Gallager, *Principles of digital communication*. Technical Publications, 2008.

[8] X. Cao, Y. Liu, and D. Hu, "Simplified llr algorithm for m-QAM demodulation," *The Journal of Engineering*, vol. 2019, no. 21, pp. 7370–7375, 2019.

[9] Z. Hu, F. Chen, M. Wen, F. Ji, and H. Yu, "Low-complexity llr calculation for ofdm with index modulation," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 618–621, 2018.

[10] J. Sidi and R. Sundaresan, "Log-likelihood ratio (LLR) computation using piecewise linear approximation of LLR functions," Jul. 29 2014, uS Patent 8,793,560.

[11] L. Guerrieri and P. Bisaglia, "Method of approximating log-likelihood ratios in QAM digital transmissions and relative QAM soft-output de-mapper," Sep. 13 2011, uS Patent 8,019,024.

[12] S. Allpress, C. Luschi, and S. Felix, "Exact and approximated expressions of the log-likelihood ratio for 16-qam signals," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1. IEEE, 2004, pp. 794–798.

[13] H. V. Poor, *An introduction to signal detection and estimation*. Springer Science & Business Media, 2013.

[14] W. Rudin, *Real and complex analysis*. Tata McGraw-hill education, 2006.

dates. See
Al press paper