

Optimal Rate-Reliability-Delay Tradeoff in Networks with Composite Links¹

Ying Li² Mung Chiang² A. Robert Calderbank² Suhas N. Diggavi³

Abstract

Networks need to accommodate diverse applications with different Quality-of-Service (QoS) requirements. New ideas at the physical layer are being developed for this purpose, such as diversity embedded coding, which is a technique that combines high rates with high reliability. We address the problem of how to fully utilize different rate-reliability characteristics at the physical layer to support different types of traffic over a network and to jointly maximize their utilities.

We set up a new framework based on utility maximization for networks with composite links, meaning that each link consists of sub-links that can attain different rate-reliability characteristics simultaneously. We incorporate delay, in addition to rate and reliability, into the utility functions. To accommodate different types of traffic, we propose distributed algorithms converging to the optimal rate-reliability-delay tradeoff based on capacity division and priority queueing. Numerical results show that compared with traditional codes, the new codes can provide higher network utilities for all traffic types simultaneously. The results also show that priority queueing achieves higher network utility than capacity division.

Index Terms

Networks, Resource management, Optimization methods, Communication systems, Channel coding.

I. INTRODUCTION

It is important and challenging to design a network which accommodates diverse applications, because different types of traffic have very different requirements for rate, reliability and delay. For example, real-time traffic needs lower delay, but non-real time traffic is delay insensitive. Different types of traffic have different utility functions, some of them elastic and some inelastic, all to be provisioned over the network.

Recent innovations at the physical layer offer new perspectives on rate-reliability-delay tradeoff. Embedded diversity [6] [7] is a way of combining high rates with high reliability through innovation in coding at the physical layer. Diversity-embedded codes are high-rate codes that are designed so that they have a high-diversity (reliability) but low-rate code embedded within them. Diversity embedded codes can achieve

¹ Part of this paper was presented at IEEE INFOCOM 2007 [15].

² Department of Electrical Engineering, Princeton University, USA.

³ School of Computer and Communication Sciences, Swiss Federal Institute of Technology, Switzerland.

rate-diversity points that cannot be achieved by simple switching strategies such as time division multiplex (TDM) [5] [6]. Several classes of diversity embedded codes have been constructed. Linear codes were developed in [5] and a class of non-linear multi-level codes were developed in [6]. Codes based on peaky signaling were constructed in [4].

Diversity-embedded codes allow a form of communication where the high-rate code opportunistically takes advantage of good channel realizations whereas the embedded high-diversity code ensures that at least part of the information is received reliably. The new codes offer different levels of error protection on a given link, and they *simultaneously* create data streams with different rate-reliability characteristics, hence the link can be treated as several sub-links with different rate and reliability. We refer to such links as *composite links* and they provide more flexibility for the rate-reliability tradeoff.

These new physical layer techniques allow us to allocate the packets of different traffic types to the sub-links with appropriate rate-reliability characteristics, in a way that the traffic types match the rate-reliability pairs, so that the utilities of different traffic types can be jointly optimized. We address the problem of how to fully utilize the different rate-reliability characteristics at the *physical layer* to support different types of traffic over a *network* and to jointly maximize their utilities.

We study this problem by extending the framework of Network Utility Maximization (NUM). Since the publication of the seminal paper [12], the framework of NUM has found many applications in network resource allocation algorithms and the design of protocol stacks. Consider a communication network with L logical links, each with a fixed capacity of c_l bps, and S sources, each transmitting at a rate of x_s bps. Each source s emits one flow, using a fixed set $L(s)$ of links in its path, and has a utility function $U_s(x_s)$. Each link l is shared by a set $S(l)$ of sources. NUM, in its basic version, is the following problem of maximizing the network utility, over the source rates \mathbf{x} , subject to linear flow constraints for all links:

$$\begin{aligned} & \text{maximize} && \sum_s U_s(x_s) \\ & \text{subject to} && \sum_{s \in S(l)} x_s \leq c_l, \quad \forall l \\ & && \mathbf{x} \succeq 0. \end{aligned} \tag{1}$$

The basic NUM framework has been widely extended (see [3] and the references therein). Recently, [14] extended the NUM framework to the case that utility for each user depends on both transmission rate and signal quality, with an intrinsic tradeoff between the two. This extension is motivated as follows: Basic NUM assumes that each link provides a fixed-size transmission ‘pipe’ and each user’s utility is only a function of transmission rate. These assumptions break down in many practical systems, where, by adapting the physical layer channel coding or transmission diversity, different tradeoffs between rate and

reliability can be achieved.

When the diversity-embedded codes are brought into the NUM framework, we may think of a given link as several parallel sub-links, each with different rate and reliability. The authors of [14] have modeled the NUM framework for adaptive coding schemes, but what is missing is the freedom to assign users' traffic to different sub-links so that users can have different rates, reliabilities, and delays. In this paper, we show the impact of having composite links in networks. Different from [14], the channel coding is assumed to be fixed and the parameters of the rate and reliability at different sub-links achieved by the novel channel coding at the physical layer are passed up to the upper layers, and then our framework decides how to assign users' traffic to different sub-links. The implementation of our framework is a transport layer and application layer functionality, in which parameters are passed up from the lower layers.

An equally important theme in this paper is the incorporation of delay into the utility objective function, in addition to rate and reliability. In basic NUM, the utility function is only a function of rate, which does not sufficiently reflect user requirements for reliability or delay. In [14], the tradeoff between rate and reliability has been discussed, but not delay. In recent work, e.g., [8], the throughput-delay tradeoff is investigated for wireless network transport capacity, but as an asymptotic study based on the concept of transport capacity rather than algorithmic design for utility optimization. Some utility functions of rate and delay have been studied [16] [17], but not reliability. Here we have for each source s a utility that depends on rate x_s , reliability ρ_s and delay δ_s :

$$U_s = U_s(x_s, \rho_s, \delta_s). \quad (2)$$

We consider three types of traffic: (1) voice traffic (e.g., VoIP), where each source has a fixed rate and its utility is a function of delay and packet loss; (2) delay-insensitive traffic, where utility is a function of rate and reliability; (3) and delay-sensitive traffic, where utility depends on rate, reliability and delay.

In addition, we include automatic repeat request (ARQ) [18] as a link-level degree of freedom in optimizing for rate/reliability/delay. ARQ is a protocol that the receiver automatically requests the transmitter to resend the packet when it detects an error in a packet. In some wireless systems, the wireless link is not as reliable as the wired one, and an ARQ protocol is used to enhance the reliability of the wireless link in a feedback manner, as well as forward error correcting (FEC) which enhances the reliability in a feedforward manner. But ARQ may increase the latency. Hence ARQ would affect the rate-reliability-delay tradeoff. Papers [9] and [10] discuss the rate-reliability-delay tradeoff with ARQ, but they focus on a single link where the link has only one rate-reliability pair. Here we investigate the case of general network topologies

and each link has multiple rate-reliability pairs.

Based on the framework, we study distributed algorithms for the optimal rate-reliability-delay tradeoff. In the basic NUM, convexity and separability properties of the optimization problem readily lead to a distributed algorithm that converges to the globally optimal rate allocation. However, our new optimization formulations for the rate-reliability-delay tradeoff are neither separable nor convex. The freedom of allocating traffic to different sub-links introduces more variables and nonconvexity to the optimization problems, the incorporation of delay into utility function leads to more global coupling across the variables, and the possibility of ARQ further complicates the problem structure. Despite these difficulties, we develop distributed algorithms based on network pricing. In contrast to standard pricing-based rate control algorithms for the basic NUM, in which each link communicates the same congestion price to each of its users and/or each user communicates its willingness to pay for rate allocation to the network, in our algorithms each link provides a possibly different congestion price to users of different traffic type and each user provides its willingness to pay for the reduction of delay to the network.

The main contributions of this paper are as follows:

- 1) We set up a new framework based on NUM, which applies to networks with composite links, and quantify the benefits of a joint design over rate control, forward error control, and feedback error control on a network.
- 2) We incorporate delay, in addition to rate and reliability, within the utility function, for the purpose of accommodating different types of traffic effectively. We show that the innovation at the physical layer, such as diversity-embedded codes, can achieve higher network utility in a network setting.
- 3) The NUM formulation for composite links turns out to be nonconvex and we develop new techniques to tackle such nonconvex optimization problems.

The rest of this paper is organized as follows. In Section II, we provide the system model for the network with composite links that we consider in this paper and discuss the delay formulas. In Section III and Section IV, we investigate the optimal rate-reliability-delay tradeoff based on capacity division and priority queueing for mixed traffic, respectively. Also, we propose distributed algorithms for the optimal tradeoff. We provide numerical examples in Section V to illustrate how our framework and algorithms can be used to show the value of the innovation at physical layer through network utility. We conclude in Section VI. The key notations used throughout this paper are summarized in Table I.

TABLE I
SUMMARY OF KEY NOTATION

Notation	Meaning
x_s	Source rate of source s
ρ_s	Reliability of source s
p_s	End-to-end packet loss of source s
δ_s	End-to-end packet delay of source s
γ_s	VoIP R-factor of source s
$p_{l,n}$	Packet error probability for n -th sub-link of link l
$R_{l,n}$	Maximal source rate for n -th sub-link of link l
$q_{l,n,s}$	Fraction of source s ' traffic allocated to n -th sub-link of link l
$q_{l,n}$	Fraction of the traffic allocated to n -th sub-link of link l
$y_{l,n}$	Total data traffic on n -th sub-link of link l
z_l	Total data traffic on link l
$d_{l,n}$	Current packet delay for n -th sub-link of link l
d^s	Current packet delay of source s
λ_l	Congestion price of link l
λ^s	End-to-end aggregate congestion price of source s
μ_s	Delay price of source s
μ^l	End-to-end aggregate delay price of link l
K	Packet length
N	The number of sub-links on a link
$\hat{\cdot}$	Indicating data traffic
\sim	Indicating voice traffic

II. SYSTEM MODEL

Suppose data from a source is transmitted in packets and each packet consists of K bits. We consider VoIP traffic and delay insensitive/sensitive data traffic. We assume that the network can differentiate traffic type and we consider two schemes to multiplex different types of traffic. A simpler scheme in Section III is capacity division, which divides the link capacity and assigns each traffic type a certain capacity within which each traffic type has its own transmission. A more sophisticated scheme in Section IV is priority queueing, which puts all the packets together but annotates them according to the degree of delay sensitivity.

We assume the maximal rate of link l is c_l (bps). Suppose that at the physical layer channel codes with N rate-reliability characteristics are used, so that link l consists of N sub-links, and for the n -th sub-link, the packet error probability is $p_{l,n}$ and the maximal source rate it can support is $R_{l,n} = c_l r_{l,n}$, where $r_{l,n}$ is the code rate, which is defined as the ratio of the input of the n -th sub-link and the output bit rate of the channel encoder of link l ¹.

Assume that one packet is transferred on one of the sub-links, and that source s uses link l in a way

¹We fix the rate-reliability tuple used in the diversity embedded code by choosing the tuple $\{R_{l,1}, p_{l,1}, \dots, R_{l,N}, p_{l,N}\}$ for each link l . This assignment decouples the rate and reliability across sub-links.

that a fraction $q_{l,n,s}$ is randomly allocated to n -th sub-link of link l . We assume that on every sub-link the incoming packets are stored in a queue and transmitted in a first-in-first-out (FIFO) fashion, that the arrivals at the entry points are Poisson streams, and that we have the Kleinrock independence approximation [2]. Note that in general, Kleinrock independence assumption is more accurate in networks where nodes serves many flows from different locations, and when there are moderate number of random flows in a network, Kleinrock independence assumption can reasonably capture the traffic correlation [2]. We use Kleinrock independence assumption to derive the close form average packet delay for further analysis to get the insight of the rate-reliability-delay tradeoff and the insight of how to design the distributed algorithms.

We use ρ_s to denote the reliability of source s , $\rho_s = 1 - p_s$ where p_s is the end-to-end packet error probability of source s . The delay of source s refers to the average end-to-end packet delay, denoted as δ_s . Packet delay consists of processing, queueing, transmission and propagation delays. In this paper, we consider the transmission and queueing delay. We discuss ρ_s and δ_s in two scenarios, without/with ARQ.

A. Without ARQ

For the case of no ARQ, the Kleinrock independence approximation implies that the queues can be approximated as NL independent $M/D/1$ queues. The reliability for source s is

$$\rho_s(\mathbf{q}) = \prod_{l \in L(s)} \sum_n q_{l,n,s} (1 - p_{l,n}). \quad (3)$$

The average packet delay for source s is [2] [13]

$$\delta_s(\mathbf{q}, \mathbf{y}) = \frac{1}{2} K \sum_{l \in L(s)} \sum_n q_{l,n,s} \left(\frac{1}{R_{l,n}} + \frac{1}{R_{l,n} - y_{l,n}} \right), \quad (4)$$

where $y_{l,n}$ is the total rate on n -th sub-link of link l ,

$$y_{l,n}(\mathbf{x}, \mathbf{q}) = \sum_{s \in S(l)} x_s q_{l,n,s}. \quad (5)$$

Adjusting $q_{l,n,s}$, we can change reliability and delay. If we increase the load over the sub-link with higher reliability and lower capacity, reliability could be increased but delay may be increased. If we increase rate x_s , delay may be increased. There is indeed an intrinsic rate-reliability-delay tradeoff.

B. With ARQ

There are many different ARQ schemes. Here we consider the traditional stop-and-wait ARQ where the next packet is not processed until the current one is either accepted or discarded. We assume that if after I rounds of transmission the packet is still in error, we discard it and declare a packet loss.

We assume that each sub-link uses ARQ separately and the failed packet is resent on the same sub-link, not crossing to other sub-links. If the failed packet on a low-reliability sub-link is resent on a high-reliability

sub-link, the packet delay would be reduced [6], but here for analysis we consider the case without crossing the composite links. By the Kleinrock independence approximation, the queues can be approximated as NL independent $M/G/1$ queues. The reliability of source s is the same as without ARQ, except that we have $p_{l,n}^I$ instead of $p_{l,n}$ as the packet error rate.

For delay, first we focus on one link. We can represent the combination of the queue and the link using ARQ as an $M/G/1$ queue [2]. We denote ξ as the time for one round process, and we have $\xi = K/R + \Gamma$, where K/R is the transmission delay, and Γ includes the forward and feedback propagation delay and the receiver processing delay. For the service time, τ , we have

$$\bar{\tau} = \xi a, \quad \bar{\tau}^2 = \xi^2 b, \quad (6)$$

where $a = \frac{1-p^I}{1-p}$, $b = \frac{1}{1-p} + \frac{2p(1-p^{I-1})}{(1-p)^2} - \frac{(2I-1)p^I}{1-p}$, and p is the probability of detected error, which is approximated by packet error rate. Assume that Γ is proportional to K/R , then $\xi = \phi K/R$ where $\phi \geq 1$. The average delay is $\delta = K \frac{\phi a - \phi^2 \frac{x}{R} (a^2 - \frac{b}{2})}{R - \phi x a}$ [2] [13].

For the case of S sources and L links, the average packet delay for source s is

$$\delta_s(\mathbf{q}, \mathbf{y}) = K \sum_{l \in L(s)} \sum_n q_{l,n,s} \frac{a_{l,n} - \frac{y_{l,n}}{\hat{R}_{l,n}} (a_{l,n}^2 - \frac{b_{l,n}}{2})}{\hat{R}_{l,n} - y_{l,n} a_{l,n}}, \quad (7)$$

where $\hat{R}_{l,n} = \frac{R_{l,n}}{\phi}$, $a_{l,n}$ and $b_{l,n}$ are a , b as in (6) for n -th sub-link of link l .

III. OPTIMAL RATE-RELIABILITY-DELAY TRADEOFF BASED ON CAPACITY DIVISION

First we investigate the rate-reliability-delay tradeoff based on capacity division. The network assigns some capacity to VoIP traffic, and the remaining capacity is for data. VoIP and data are decoupled and we discuss them separately.

A. Voice traffic

For VoIP traffic, the ‘‘R-factor’’ formula links delay and loss to Mean Opinion Score (MOS) for speech. Either the R-factor or the MOS can serve as the utility function and we choose R-factor as the utility function. The relationship between R-factor and voice quality is that as the R-factor increases by 10 from 50 to 100, the voice quality is poor, low, medium, high, and best, respectively [11].

The R-factor is defined as [11] $\gamma = \gamma_a - \alpha_1 \delta - \alpha_2 (\delta - \alpha_3) H - \beta_1 - \beta_2 \log(1 + \beta_3 \psi)$ where δ is the one-way end-to-end delay in milliseconds, ψ is the packet loss percentage for the call, and the remaining parameters are constants defined as: $\gamma_a = 94.2$, $\alpha_1 = 0.024$, $\alpha_2 = 0.11$, $\alpha_3 = 177.3$, $H = 0$ if delay

$\delta < \alpha_3$, $H = 1$ otherwise, and β_1 , β_2 and β_3 are codec dependent parameters, for example, for the G729 codec: $\beta_1 = 12$, $\beta_2 = 15$, $\beta_3 = 0.6$.

Packet loss can be caused by several factors including overloaded links and physical media errors. We consider only packet loss due to packet error, so we approximate packet loss by packet error rate p . We may improve the R-factor by assigning VoIP traffic to the more reliable sub-link, but we find that beyond some threshold, higher reliability does not help the R-factor much. For example, for a given delay, the R-factor for a packet error rate $p \leq 10^{-3}$ is only smaller than the R-factor for $p = 0$ by less than 1.

To avoid operating at the region where the utility becomes saturated, we adopt the following **VoIP packet allocation policy**: (Here we illustrate the idea by the case of $N = 2$, where N is the number of sub-links, since it is straightforward to extend it to $N > 2$.)

Denote by p_H and p_L the average packet error on high and low reliability sub-link, respectively.

Case 1: If p_H is greater than a threshold p_{thr} , VoIP packets use high-reliability sub-link.

Case 2: If $p_H \leq p_{thr}$ and $p_L \geq p_{thr}$, VoIP packets use both sub-links by allocating randomly a fraction q_H of the traffic to high-reliability sub-link so that $q_H p_H + (1 - q_H) p_L = p_{thr}$.

Case 3: If $p_L \leq p_{thr}$, VoIP packets use the low-reliability sub-link which is reliable enough for VoIP. The high-reliability sub-link is assigned to data, to increase the reliability of data.

Assume that the above policy is the same for all VoIP sources. The policy gives us $q_{l,n}$, hence by (3) we have the reliability of source s , ρ_s . Then R-factor of source s is only a function of δ_s ,

$$\gamma_s(\delta_s) = \gamma_a - \alpha_1 \delta_s - \alpha_2 (\delta_s - \alpha_3) H - \beta_1 - \beta_2 \log(1 + 100\beta_3(1 - \rho_s)). \quad (8)$$

Assume that each VoIP source is assigned a capacity of B Kbps. For link l , we assign a capacity $|S(l)|B$ in total to voice. We allocate voice packets according to the policy above and accordingly we assign voice a capacity of $q_{l,n}B$ from the n -th sub-link of link l . It follows from (4) that the average delay is given by

$$\delta_s(B) = \sum_{l \in L(s)} \frac{K}{2|S(l)|} \left(\frac{1}{B} + \frac{1}{B - x_s^V} \right) \quad (9)$$

where x_s^V is the rate of VoIP, which is fixed. If B is close to x_s^V , $\delta_s(B)$ is large, hence the R-factor is low. If B is large, the delay is small, but the remaining capacity for data traffic would be less, so an appropriate B should be assigned.

Given an R-factor requirement, we can find the minimum B by solving the following problem,

$$\begin{aligned}
& \text{minimize} && B \\
& \text{subject to} && \frac{1}{S} \sum_s \gamma_s \geq \gamma_{thr} \\
& && \gamma_s(\delta_s) \geq \gamma_s, && \forall s \\
& && \delta_s(B) \leq \delta_s, && \forall s \\
& && |S(l)|B \geq \sum_{s \in S(l)} x_s, && \forall l \\
& \text{variables} && B, \gamma, \delta
\end{aligned} \tag{10}$$

and the simple upper and lower bounds (“box constraints”): $\gamma_s^{min} \leq \gamma_s \leq \gamma_s^{max}$ and $\delta_s^{min} \leq \delta_s \leq \delta_s^{max}, \forall s$, where $\gamma_s(\delta_s)$ is as in (8) and $\delta_s(B)$ is as in (9). Note that throughout the paper we use δ to denote delay as a function and as the value of this function (and similarly for γ). Here the first constraint ensures that the average R-factor is no less than a given value γ_{thr} . The second states that the network must provide an R-factor to each source that is no less than its requested one. The third states that the network must provide delay to each source that is no greater than its requested delay. The fourth is the rate constraint.

Being piecewise linear, the second constraint can be replaced by two linear constraints and problem (10) is then a convex optimization. We can solve it distributively by a standard dual decomposition [3], and we can assign different B to different sources using different links.

There is also a suboptimal but simpler way to find B without any message passing. From (9) we can have a typical value of δ_s as $|L(s)|_{typ} \frac{K}{2|S(l)|_{typ}} \left(\frac{1}{B} + \frac{1}{B-x_s} \right)$, where $|L(s)|_{typ}$ and $|S(l)|_{typ}$ are the typical values (for example, the mean value) based on the network topology. For a given R-factor requirement, we can estimate B by using the typical value, and fix it instead of dynamically assigning it.

B. Delay insensitive/sensitive data traffic

1) *Throughput*: For delay insensitive traffic, the utility is a function of the throughput. The problem is formulated as

$$\begin{aligned}
& \text{maximize} && \sum_s U_s(x_s \rho_s) \\
& \text{subject to} && \sum_{s \in S(l)} x_s q_{l,n,s} \leq R_{l,n}, && \forall l, \forall n \\
& && \sum_n q_{l,n,s} = 1, && \forall l, s \in S(l) \\
& \text{variables} && \mathbf{x}, \mathbf{q},
\end{aligned} \tag{11}$$

together with the box constraints on \mathbf{q} and \mathbf{x} , where ρ_s is as in (3) and $x_s \rho_s$ is the throughput of source s . To make the rest of the development concrete, we focus on the maximization of the total throughput of all the sources, where the utility function is $U_s(x_s \rho_s) = x_s \rho_s$.

The error probability p is usually small, and the reliability function ρ_s as in (3) satisfies

$$\rho_s \approx 1 - \sum_{l \in L(s)} \sum_n q_{l,n,s} p_{l,n}. \tag{12}$$

By (12) and (5), problem (11) with $U_s(x_s\rho_s) = x_s\rho_s$ can be changed into a linear programming with variables \mathbf{x} and \mathbf{y} . From the optimal \mathbf{x} and \mathbf{y} , we can find \mathbf{q} . The only equation to be satisfied is (5), hence we have flexibility to choose \mathbf{q} . The simplest way is to make $q_{l,n,s}$ equal for all s , given l and n . The following lemma is straightforward.

Lemma 1: Given (12), problem (11) with $U_s(x_s\rho_s) = x_s\rho_s$ is optimized if $q_{l,n,s}$ are equal for all $s \in S(l)$, given l and n .

The lemma means that we can simply write the variable $q_{l,n,s}$ into $q_{l,n}$, with $q_{l,n} = \frac{y_{l,n}}{\sum_{s \in S(l)} x_s}$, $\forall l, \forall n$.

2) *Throughput and Delay:* For delay sensitive data traffic, the packet delay may enter the utility function, as well as the throughput. For the case of no ARQ, the extended NUM problem is formulated as

$$\begin{aligned} & \text{maximize} && \sum_s U_s(x_s\rho_s, \delta_s) \\ & \text{subject to} && \delta_s(\mathbf{q}, \mathbf{y}) \leq \delta_s, \forall s \\ & && y_{l,n} + y_{l,n}^V \leq R_{l,n}, \forall l, \forall n \\ & \text{variables} && \mathbf{x}, \mathbf{y}, \boldsymbol{\delta}, \mathbf{q}. \end{aligned} \tag{13}$$

and constraints $\sum_n q_{l,n,s} = 1$, for all l, s , together with box constraints on \mathbf{q} , \mathbf{x} and $\boldsymbol{\delta}$, where ρ_s is as in (12), $y_{l,n}$ is as in (5), $\delta_s(\mathbf{q}, \mathbf{y})$ is as in (4), $y_{l,n}^V$ is the capacity taken by voice on n -th sub-link of link l . The first constraint states that the network must provide delay to each source that is no greater than its requested delay. The second is the rate constraint.

For the utility function for user s , we can use the following weighted sum of utility on throughput and utility on delay, with the weight $w_s \in [0, 1]$ indicating the relative importance of throughput and delay,

$$U_s(x_s\rho_s, \delta_s) = w_s \frac{x_s\rho_s - (x_s\rho_s)^{\min}}{(x_s\rho_s)^{\max} - (x_s\rho_s)^{\min}} - (1 - w_s) \frac{\delta_s - \delta_s^{\min}}{\delta_s^{\max} - \delta_s^{\min}}. \tag{14}$$

Different users can have different weights for delay and throughput, and different maximum delay constraints based on their traffic types.

We consider the case that all the users on link l have the same rate allocation rule, i.e., the $q_{l,n,s}$ are equal for all $s \in S(l)$ given l and n . This makes the system simple, and by Lemma 1 it is close to the optimal rate allocation rule if the weight on delay is small. The sum of weighted throughput $\sum_s w_s x_s \rho_s$ cannot be changed into a linear function in \mathbf{x} and \mathbf{y} using (5), but when $p_{l,n}$'s are small, we can approximate it by $\sum_s w_s x_s - w \sum_l \sum_n y_{l,n} p_{l,n}$, where w is the average w_s .

Problem (13) is a nonconvex optimization problem. But we can make it convex under certain conditions, or approximate it as a convex problem. Note that if we have further constraint that for every link the fraction of the load on every sub-link $(y_{l,n} + y_{l,n}^V) / \sum_n (y_{l,n} + y_{l,n}^V)$ is fixed, which may be required by directly

applying some diversity embedded codes at the physical layer, then the problem can be shown to be convex optimization by expressing $q_{l,n}$ in terms of $y_{l,n}$ only and showing the average delay is convex in $y_{l,n}$. The additional constraint is not always necessary and in general we don't have the convexity.

We denote z_l as the total rate on link l ,

$$z_l = \sum_{s \in S(l)} x_s \quad (15)$$

and we also have $y_{l,n} = q_{l,n} z_l$ where $y_{l,n}$ is as in (5), so problem (13) becomes

$$\begin{aligned} & \text{maximize} && \sum_s w_s x_s - w \sum_l \sum_n y_{l,n} p_{l,n} - \sum_s v_s \delta_s \\ & \text{subject to} && \delta_s(\mathbf{y}, \mathbf{z}) \leq \delta_s, \forall s \\ & && y_{l,n} \leq R_{l,n} - y_{l,n}^V, \forall l, n \\ & && \sum_n y_{l,n} = z_l, \forall l \\ & && \sum_{s \in S(l)} x_s \leq z_l, \forall l \\ & \text{variables} && \mathbf{x}, \mathbf{y}, \boldsymbol{\delta}, \mathbf{z}, \end{aligned} \quad (16)$$

where v_s are the corresponding constants, $\delta_s(\mathbf{y}, \mathbf{z}) = \sum_{l \in L(s)} \sum_n \frac{K}{2z_l} f(y_{l,n})$ where $f(y_{l,n}) = y_{l,n} \left(\frac{1}{R_{l,n}} + \frac{1}{R_{l,n} - y_{l,n}} \right)$. The last constraint can be an inequality because at optimality it will always be satisfied with equality.

We introduce auxiliary variables $d_{l,n}$, and write the first constraint into $\delta_s \geq \sum_{l \in L(s)} \sum_n d_{l,n}$ and $d_{l,n} \geq \frac{K}{2z_l} f(y_{l,n})$. If we take log on both sides of the inequality $d_{l,n} \geq \frac{K}{2z_l} f(y_{l,n})$, to make the problem convex, we need $\log[f(y_{l,n})]$ to be convex in $y_{l,n}$, but it is not. However, we can find an approximation of $f(y_{l,n})$, denoted as $\hat{f}(y_{l,n})$, such that $\log[\hat{f}(y_{l,n})]$ is convex in $y_{l,n}$, and $|f(y_{l,n}) - \hat{f}(y_{l,n})| \leq \Lambda$, where Λ is small.

If for delay $\delta_s(\mathbf{y}, \mathbf{z}) = \sum_{l \in L(s)} \sum_n \frac{K}{2z_l} f(y_{l,n})$ we use \hat{f} instead of f , the relative approximation error is upper bounded by $\frac{0.5\Lambda N \sum_{l \in L(s)} K/z_l}{\delta_s}$, where N is the number of sub-links, z_l is the total rate on link l . Suppose the packet length $K = 400$, $z_l = 1$ Mbps, then $K/z_l = 0.4$ ms. The average delay δ_s may typically be tens of ms because source s must have at least one heavy-loaded queue on its path, otherwise it could increase its rate to enlarge the throughput. Suppose δ_s is about 20ms, then the upper bound of the approximation error becomes $\frac{0.5\Lambda N |L(s)| 0.4}{20}$. If we use $|L(s)| = 4$ and $N = 2$, the bound becomes 0.08Λ . So if $\Lambda = 0.5, 1, 2$, the approximation error is bounded by 4%, 8%, 16%, respectively. Furthermore, if $(f(y_{l,n}) - \hat{f}(y_{l,n}))$ can be positive on some sub-links and negative on some others, the approximation error would be averaged out because $f(y_{l,n})$ are summed over sub-links as in the delay formula of $\delta_s(\mathbf{y}, \mathbf{z})$. As will be shown in the last graph in the numerical examples, the approximation is good.

Lemma 2: Let f and \hat{f} be given by $f(x) = x + x/(1-x)$ and $\hat{f}(x) = \frac{1-10^{-m}x}{(1-x)(1-10^{-m})} - 0.5$, where $m > 1$. Then $f(x) - \hat{f}(x) \in [-0.5, 0.5]$ for $x \in [0, 1 - 10^{-m}]$ and $\log(\hat{f}(x))$ is convex in x .

Applying Lemma 2, problem (13) can be approximated as a convex optimization problem,

$$\begin{aligned}
& \text{maximize} && \sum_s w_s x_s - w \sum_l \sum_n y_{l,n} p_{l,n} - \sum_s v_s \delta_s \\
& \text{subject to} && \sum_{l \in L(s)} \sum_n d_{l,n} \leq \delta_s, \forall s \\
& && \log(\hat{f}(y_{l,n})) \leq \log(d_{l,n}) + \log(z_l), \forall l, \forall n \\
& \text{variables} && \mathbf{x}, \mathbf{y}, \boldsymbol{\delta}, \mathbf{z}, \mathbf{d},
\end{aligned} \tag{17}$$

together with the last three constraints in problem (16) and the box constraints on \mathbf{x} and $\boldsymbol{\delta}$, where $\hat{f}(y_{l,n}) = \frac{K}{2} \frac{R_{l,n} - 10^{-m} y_{l,n}}{R_{l,n} - y_{l,n}}$. We can set m to be 3 to let the interval required by Lemma 2 $y_{l,n}/R_{l,n} \in [0, 1 - 10^{-m}]$ close to $y_{l,n}/R_{l,n} \in [0, 1)$.

For the case with ARQ, by the same technique, we can approximate the problem to be convex. As in the average packet delay formula (7), we denote $f(y_{l,n}) = y_{l,n} \frac{a_{l,n} - \frac{y_{l,n}}{R_{l,n}} (a_{l,n}^2 - \frac{b_{l,n}}{2})}{R_{l,n} - y_{l,n} a_{l,n}}$, and we can have its approximation $\hat{f}(y_{l,n}) = \frac{R_{l,n} - 0.01 y_{l,n} a_{l,n}}{(R_{l,n} - y_{l,n} a_{l,n})^T}$ for $\phi = 1$ and $I = 3$, where $T = 3.2 \frac{(a_{l,n}^2 - \frac{b_{l,n}}{2})}{a_{l,n}^2} + 0.35$. For the same reason as in the no-ARQ case, we care about $|f(y_{l,n}) - \hat{f}(y_{l,n})|$ and we observed that it is less than 1 when $a_{l,n} y_{l,n}/R_{l,n} \in [0, 0.99]$. The optimization problem can be approximated as the same as (17), except that $\hat{f}(y_{l,n})$ has a different expression.

To solve problem (17), by the standard dual decomposition approach [14], we propose the following distributed algorithm where each source and each link solve their own problem with only local information. We introduce small positive numbers ϵ_1 and ϵ_2 to make the objective function in (17) strictly concave in x_s and δ_s ,

$$\sum_s (w_s x_s - \epsilon_1 x_s^2) - w \sum_l \sum_n y_{l,n} p_{l,n} - \sum_s (v_s \delta_s + \epsilon_2 \delta_s^2). \tag{18}$$

If we choose $\epsilon \leq \frac{a\delta}{x_{max}}$, we have $\frac{|(ax - \epsilon x^2) - ax|}{ax} \leq \kappa$, which means the approximation error is bounded by κ if ax is approximated by $(ax - \epsilon x^2)$, where a is positive.

Distributed Algorithm 1:

In each iteration t , by solving the following problem (19) over (x_s, δ_s) , each source s determines its information data rate and requested delay (i.e., $x_s(t)$ and $\delta_s(t)$) that maximize its net utility based on the prices $(\lambda^s(t), \mu_s(t))$ in the current iteration. Furthermore, by price update equation (20), the source adjusts its offered price per unit delay reduction for the next iteration.

Source problem and delay price update at source s :

Source problem over (x_s, δ_s) :

$$\text{maximize} \quad w_s x_s - \epsilon_1 x_s^2 - v_s \delta_s - \epsilon_2 \delta_s^2 - \lambda^s(t) x_s + \mu_s(t) \delta_s \tag{19}$$

with the box constraints on x_s and δ_s , where $\lambda^s(t) = \sum_{l \in L(s)} \lambda_l(t)$ is the end-to-end congestion price.

Delay price update:

$$\mu_s(t+1) = [\mu_s(t) - \alpha(t) (\delta_s(t) - d^s(t))]^+, \quad (20)$$

where $d^s(t) = \sum_{l \in L(s)} \sum_n d_{l,n}(t)$ is the end-to-end delay, $\alpha(t)$ is the stepsize, and $[a]^+ = \max\{a, 0\}$.

Concurrently at each iteration t , by solving problem (21) over $(z_l, y_{l,n}, d_{l,n})$, each link l determines its total rate, rate on sub-link n and packet delay on sub-link n , (i.e., $z_l(t)$, $y_{l,n}(t)$ and $d_{l,n}(t)$) that maximize the ‘net revenue’ of the network based on the prices in the current iteration. In addition, by price update equations (22), the link adjusts its congestion price per unit rate for the next iteration.

Link problem and congestion price update at link l :

Link problem over $(z_l, y_{l,n}, d_{l,n})$:

$$\begin{aligned} & \text{maximize} && -w \sum_n y_{l,n} p_{l,n} + \lambda_l(t) z_l - \mu^l(t) \sum_n d_{l,n} \\ & \text{subject to} && \log(\hat{f}(y_{l,n})) \leq \log(d_{l,n}) + \log(z_l), \forall l, \forall n \\ & && y_{l,n} \leq R_{l,n} - y_{l,n}^V, \forall l, \forall n \\ & && \sum_n y_{l,n} = z_l, \forall l. \end{aligned} \quad (21)$$

where $\mu^l(t) = \sum_{s \in S(l)} \mu_s(t)$ is the aggregate delay reduction price paid by sources using link l .

Congestion price update:

$$\lambda_l(t+1) = [\lambda_l(t) - \alpha(t) (z_l(t) - x^l(t))]^+, \quad \forall l, \quad (22)$$

where $x^l(t) = \sum_{s \in S(l)} x_s(t)$ is the aggregate information rate on link l at iteration t .

The message passing is as same as the integrated dynamic reliability policy in [14], except that the price μ_s is the price per unit delay reduction that the source s must pay to the network, instead of the price per unit reliability as in [14], and for the update of μ_s , the delay on every sub-link of link $l \in L(s)$, $d_{l,n}$, is needed, not just the link message. The stepsize $\alpha(t)$ satisfies $\lim_{t \rightarrow \infty} \alpha(t) = 0$ and $\lim_{t \rightarrow \infty} \sum_{i=1}^t \alpha(i) = \infty$.

After the above dual decomposition, the following result can be proved using standard techniques from the convergence analysis of the distributed gradient algorithm:

Theorem 1: By Algorithm 1, dual variables $\lambda(t)$ and $\mu(t)$ converge to the optimal dual solutions λ^* and μ^* and the corresponding primal variables \mathbf{x}^* , δ^* , \mathbf{z}^* , \mathbf{y}^* and \mathbf{d}^* are the globally optimal primal solutions of (17) where the objective function is as in (18).

Proof: Since strong duality holds for the primal problem and its Lagrange dual, we solve the dual problem through distributed gradient method and recover the primal optimizers from the dual optimizers.

The algorithm is a gradient projection algorithm for dual problem. The proof uses the results of [1] and is similar to that given in [14] and we omit the details. \blacksquare

IV. OPTIMAL RATE-RELIABILITY-DELAY TRADEOFF BASED ON PRIORITY QUEUEING SCHEME

We now investigate the priority queueing scheme, where voice packets have high priority, and data packets have low priority. We use $\check{\cdot}$ to indicate voice and $\hat{\cdot}$ to indicate data.

We assume that at the source nodes, the arrival processes of voice and data are independent, Poisson, and independent of the service times. Two separate queues are maintained for voice and data packets, respectively. On link l , voice and data packets are randomly allocated to the n -th sub-link with probabilities $\check{q}_{l,n}$ and $\hat{q}_{l,n}$, respectively. For voice, $\check{q}_{l,n}$ are decided by the VoIP packet allocation policy in section III-A.

In addition to the Kleinrock independence approximation, we assume that the arrival processes of voice and data at each link are independent. In general, this may not be true since the departure processes of voice and data are correlated. But if for any node in the network, the total number of the streams starting from or ending at the node is large compared with the number of the streams passing this node somewhere in the middle of their paths, the correlation of the departure processes of voice and data will not be significant. For a network with moderate number of random flows, such assumption can be reasonably feasible. We consider the nonpreemptive priority rule whereby a packet undergoing service is allowed to complete service without interruption even if a packet of higher priority arrives in the meantime. The case of preemptive priority is similar, and will not be presented here.

The average delay of a voice packet for source \check{s} is [2]

$$\delta_{\check{s}}(\hat{\mathbf{y}}) = \sum_{l \in L(\check{s})} \sum_n \check{q}_{l,n} \left(\frac{K}{R_{l,n}} + \frac{K}{2R_{l,n}} \frac{\check{y}_{l,n} + \hat{y}_{l,n}}{R_{l,n} - \check{y}_{l,n}} \right) \quad (23)$$

and the average delay of a data packet for source \hat{s} is

$$\delta_{\hat{s}}(\hat{\mathbf{q}}, \hat{\mathbf{y}}) = \sum_{l \in L(\hat{s})} \sum_n \frac{\hat{q}_{l,n} K}{R_{l,n}} + \frac{\hat{q}_{l,n} K}{2(R_{l,n} - \check{y}_{l,n})} \frac{\check{y}_{l,n} + \hat{y}_{l,n}}{R_{l,n} - \check{y}_{l,n} - \hat{y}_{l,n}}, \quad (24)$$

where $\check{y}_{l,n} = \sum_{s \in \check{S}(l)} \check{x}_s \check{q}_{l,n}$ and $\hat{y}_{l,n} = \sum_{s \in \hat{S}(l)} \hat{x}_s \hat{q}_{l,n}$ are the total voice and data traffic on n -th sub-link of link l , respectively. Note that $\check{\mathbf{q}}$ and $\check{\mathbf{y}}$ are fixed by the voice packet allocating policy in section III-A. If for some sub-link, there is only one type of stream, we use the delay formula of the $M/D/1$ queue (4).

We can see that voice and data are coupled, with a tradeoff between voice utility and data utility. To characterize this tradeoff, we optimize the weighted sum of voice utility and data utility, with $v \in [0, 1]$ and $(1 - v)$ being the weights for voice and data respectively. The utility function $U_{\check{s}}$ is given by (14) and

for $U_{\hat{s}}$, we use $U_{\hat{s}}(\gamma_{\hat{s}}) = \gamma_{\hat{s}}/(\gamma_{\hat{s}}^{max} - \gamma_{\hat{s}}^{min})$. The extended NUM problem is formulated as

$$\begin{aligned}
& \text{maximize} && \frac{\nu}{|\hat{S}|} \sum_{\hat{s}} U_{\hat{s}}(\gamma_{\hat{s}}) + \frac{1-\nu}{|\hat{S}|} \sum_{\hat{s}} U_{\hat{s}}(x_{\hat{s}}\rho_{\hat{s}}, \delta_{\hat{s}}) \\
& \text{subject to} && \check{y}_{l,n} + \hat{y}_{l,n} \leq R_{l,n}, \forall l, \forall n \\
& && \gamma_{\hat{s}} \leq \gamma_{\hat{s}}(\delta_{\hat{s}}), \forall \hat{s}; \quad \delta_{\hat{s}}(\hat{\mathbf{y}}) \leq \delta_{\hat{s}}, \forall \hat{s}; \quad \delta_{\hat{s}}(\hat{\mathbf{q}}, \hat{\mathbf{y}}) \leq \delta_{\hat{s}}, \forall \hat{s} \\
& \text{variables} && \boldsymbol{\gamma}, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \check{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}, \hat{\mathbf{q}},
\end{aligned} \tag{25}$$

and the box constraints on $\boldsymbol{\gamma}$, $\check{\boldsymbol{\delta}}$, $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\delta}}$. $\gamma_{\hat{s}}(\delta_{\hat{s}})$, $\delta_{\hat{s}}(\hat{\mathbf{y}})$ and $\delta_{\hat{s}}(\hat{\mathbf{q}}, \hat{\mathbf{y}})$ are as in (8), (23) and (24), respectively. Here the first constraint is the rate constraint, the second states that the network must provide an R-factor no less than its requested one to each VoIP source, and the third and fourth ensure the delay requirements. We use the same technique as in Section III-B.2 to find a convex approximation to this problem.

Lemma 3: If $0 \leq \nu \leq 0.5$, then $f(x) = x(1 + 0.5\frac{\nu+x}{(1-\nu)(1-\nu-x)})$ can be approximated by $\hat{f}(x) = 0.5\frac{1-10^{-m}x}{1-\nu-x}$ with the approximation error $|\hat{f}(x) - f(x)|$ less than 1 for $x \in [0, 1 - 10^{-m} - \nu]$, where m is a positive real number.

We apply Lemma 3 to problem (25), where $\nu = \check{y}_{l,n}/R_{l,n}$. Hence if $\nu \leq 0.5$, the delay formula (24) can be approximated as $\delta_{\hat{s}}(\hat{\mathbf{q}}, \hat{\mathbf{y}}) \approx \sum_{l \in L(\hat{s})} \sum_n K \hat{q}_{l,n} \hat{f}(y_{l,n})$ where $\hat{f}(\hat{y}_{l,n}) = 0.5\frac{R-10^{-m}\hat{y}_{l,n}}{R-\check{y}_{l,n}-\hat{y}_{l,n}}$. Here we have $\Lambda = 1$ (typically the approximation error is upper bounded by 8%) where Λ is defined in Section III-B.2.

If $\check{y}_{l,n} \leq 0.5R_{l,n}$, Lemma 3 can be applied directly. Otherwise, an approximation is needed with errors readily characterized, for which the details are omitted due to space limitations.

If ARQ is used for the data packets, the delay formulas are almost the same as the ones without ARQ. The only difference is that in the numerators $\check{y}_{l,n} + \hat{y}_{l,n}$ is changed into $\check{y}_{l,n} + b_{l,n}\hat{y}_{l,n}$, and in the denominator, $R - \check{y}_{l,n} - \hat{y}_{l,n}$ is changed into $R - \check{y}_{l,n} - a_{l,n}\hat{y}_{l,n}$ where $b_{l,n}$ and $a_{l,n}$ are as in (7).

The delay formula is approximated using the following observation: If $\nu \in [0, 0.5]$, $f(x) = ax(1 + 0.5\frac{\nu+bx}{(1-\nu)(1-\nu-ax)})$ can be approximated by $\hat{f}(x) = (0.5+0.5(\frac{b}{a}-1)(1-\nu))\frac{(1-0.001ax)}{(1-\nu-ax)} - ax$ with $|\hat{f}(x) - f(x)|$ less than $\frac{0.5}{1-\nu} + 0.5(b/a - 1)$ for $x \in [0, (0.99 - \nu)/a]$ if $b/a \in [1, 2]$. Given $I = 3$, if the packet error rate $p \leq 0.57$, then $b/a \in [1, 2]$, and $\Lambda = 2$ (typically the approximation error is upper bounded by 16%).

Similar to Section III-B.2, the problems (25) with and without ARQ are equivalent to the following,

$$\begin{aligned}
& \text{maximize} && \sum_{\hat{s}} \gamma_{\hat{s}} + \sum_{\hat{s}} v_{1\hat{s}} x_{\hat{s}} - \sum_l \sum_n v_{2l,n} p_{l,n} - \sum_{\hat{s}} v_{3\hat{s}} \delta_{\hat{s}} \\
& \text{subject to} && \sum_{l \in L(\hat{s})} \sum_n d_{l,n} \leq \delta_{\hat{s}}, \forall \hat{s} \\
& && \log(\hat{f}(\hat{y}_{l,n})) \leq \log(d_{l,n}) + \log(z_l), \forall l, \forall n \\
& && \sum_n \hat{y}_{l,n} = z_l, \sum_{\hat{s} \in \hat{S}(l)} x_{\hat{s}} \leq z_l, \forall l \\
& \text{variables} && \boldsymbol{\gamma}, \hat{\mathbf{x}}, \hat{\mathbf{y}}, \check{\boldsymbol{\delta}}, \hat{\boldsymbol{\delta}}, \mathbf{z}, \mathbf{d},
\end{aligned} \tag{26}$$

and the first three constraints in problem (25), where v_{1s} , v_2 and v_{3s} are the corresponding constants. The four constraints listed above correspond to the last constraint in (25), with auxiliary variables \mathbf{d} and \mathbf{z} .

To solve problem (26), we use a standard dual decomposition approach and we propose the following distributed algorithm. Similar to Algorithm 1, we add the following terms to the objective function in (26) to make it strictly concave in $\gamma_{\check{s}}$, $x_{\check{s}}$ and $\delta_{\check{s}}$

$$-\sum_{\check{s}} \epsilon (\gamma_{\check{s}})^2 - \sum_{\check{s}} (\epsilon_1 x_{\check{s}}^2 + \epsilon_2 \delta_{\check{s}}^2), \quad (27)$$

where ϵ , ϵ_1 and ϵ_2 are small positive numbers.

Distributed Algorithm 2:

At each iteration t , by solving the following problem (28) over $(\gamma_{\check{s}}, \delta_{\check{s}})$, each source \check{s} determines its R-factor and requested delay (i.e., $\gamma_{\check{s}}$ and $\delta_{\check{s}}$) that maximize its net utility based on the prices $\mu_{\check{s}}(t)$ in the current iteration. Furthermore, by price update equation (29), the source adjusts its offered price per unit delay reduction for the next iteration.

Source problem at VoIP source \check{s} :

Source problem over $(\gamma_{\check{s}}, \delta_{\check{s}})$:

$$\text{maximize } \gamma_{\check{s}} - \epsilon (\gamma_{\check{s}})^2 + \mu_{\check{s}}(t) \delta_{\check{s}} \quad (28)$$

with the constraints $\gamma_{\check{s}} \leq \gamma_{\check{s}}(\delta_{\check{s}})$ and box constraints on $\gamma_{\check{s}}$ and $\delta_{\check{s}}$.

Delay price update:

$$\mu_{\check{s}}(t+1) = [\mu_{\check{s}}(t) - \alpha(t) (\delta_{\check{s}}(t) - d^{\check{s}}(t))]^+, \forall \check{s}, \quad (29)$$

where $d^{\check{s}}(t) = \sum_{l \in L(\check{s})} \sum_n \delta_{\check{s}}(\hat{\mathbf{y}})$ is the end-to-end average delay of VoIP packet at iteration t .

At each iteration t , by solving problem (30) given below over $(x_{\hat{s}}, \delta_{\hat{s}})$, each source \hat{s} determines its rate and requested delay (i.e., $x_{\hat{s}}$ and $\delta_{\hat{s}}$) that maximize its net utility based on the prices $(\lambda^{\hat{s}}(t), \mu_{\hat{s}}(t))$ in the current iteration. Furthermore, by price update, the source adjusts its offered price per unit delay reduction for the next iteration.

Source problem at data source \hat{s} :

Source problem over $(x_{\hat{s}}, \delta_{\hat{s}})$:

$$\text{maximize } v_{1\hat{s}} x_{\hat{s}} - \epsilon_1 x_{\hat{s}}^2 - v_{2\hat{s}} \delta_{\hat{s}} - \epsilon_2 \delta_{\hat{s}}^2 - \lambda^{\hat{s}}(t) x_{\hat{s}} + \mu_{\hat{s}}(t) \delta_{\hat{s}} \quad (30)$$

with the box constraints on $x_{\hat{s}}$ and $\delta_{\hat{s}}$, where $\lambda^{\hat{s}}(t) = \sum_{l \in L(\hat{s})} \lambda_l(t)$ is the end-to-end congestion price.

Delay price update: Follow the same equation as (20) except that s is changed into \hat{s} .

Concurrently at each iteration t , by solving problem (31) over $(z_l, \hat{y}_{l,n}, d_{l,n})$, each link l determines its total rate, rate on sub-link n and packet delay on sub-link n , (i.e., $z_l(t)$, $\hat{y}_{l,n}(t)$ and $d_{l,n}(t)$) that maximizes

the ‘net revenue’ of the network based on the prices in the current iteration. In addition, by price update, the link adjusts its congestion price per unit rate for the next iteration.

Link problem and congestion price update at link l :

Link problem over $(z_l, \hat{y}_{l,n}, d_{l,n})$:

$$\begin{aligned}
& \text{maximize} && - \sum_{\hat{s} \in \hat{S}(l)} \sum_n v_{1\hat{s}} \hat{y}_{l,n} p_{l,n} + \lambda_l(t) z_l - \check{\mu}^l(t) \sum_n \delta_{\hat{s}}(\hat{\mathbf{y}}) - \hat{\mu}^l(t) \sum_n d_{l,n} \\
& \text{subject to} && \log(\hat{f}(\hat{y}_{l,n})) \leq \log(d_{l,n}) + \log(z_l), \forall l, \forall n \\
& && \hat{y}_{l,n} \leq R_{l,n} - \check{y}_{l,n}, \forall l, \forall n \\
& && \sum_n \hat{y}_{l,n} = z_l, \forall l
\end{aligned} \tag{31}$$

where $\check{\mu}^l(t) = \sum_{\hat{s} \in \check{S}(l)} \mu_{\hat{s}}$ is the aggregate delay reduction price paid by VoIP sources using link l at iteration t , and $\hat{\mu}^l(t) = \sum_{\hat{s} \in \hat{S}(l)} \mu_{\hat{s}}$ is the aggregate delay reduction price paid by data sources using link l .

Congestion price update: Follow the same equation as (22) with $x^l(t) = \sum_{\hat{s} \in \hat{S}(l)} x_{\hat{s}}(t)$.

We illustrate Algorithm 2 in Fig. 1. Algorithm 2 reduces to Algorithm 1 if we remove the voice source from Fig. 1. Comparing two algorithms, we can see that for each data source, it solves the same source problem except that the delay formula is different. The link problem in Algorithm 2 has an additional term in the objective compared with Algorithm 1, $-\check{\mu}^l(t) \sum_n \delta_{\hat{s}}(\hat{\mathbf{y}})$, and it requires the aggregate delay reduction price paid by VoIP sources $\check{\mu}^l(t)$. In Algorithm 2 every VoIP source needs to solve its own problem and update its price for delay reduction. The message passing for data traffic is the same for both algorithms, and for voice traffic it is similar to data traffic. In Algorithm 1, if the capacity assigned to voice B is pre-calculated and fixed, the message passing related to VoIP sources is not needed. From Algorithm 1 to Algorithm 2, the number of source nodes is effectively increased by $|\check{S}|$, the number of the VoIP flows. Since subgradient projection approach is used, both algorithms have at least linear convergence speed [1].

After the above dual decomposition, we can obtain the following analog of Theorem 1:

Theorem 2: By Algorithm 2, dual variables $\lambda(t)$, $\check{\mu}(t)$ and $\hat{\mu}(t)$ converge to the optimal dual solutions λ^* , $\check{\mu}^*$ and $\hat{\mu}^*$ and the corresponding primal variables γ^* , $\check{\delta}^*$, \hat{x}^* , \hat{y}^* , $\hat{\delta}^*$, \mathbf{d}^* , \mathbf{z}^* are the globally optimal primal solutions of (26) with (27) added to the objective function.

V. NUMERICAL RESULTS

In this section, we present numerical examples by analyzing a simple network, shown in Fig. 2. On each stream, there is both VoIP and data traffic. We consider the case that data traffic may use ARQ, but VoIP traffic does not use ARQ. We let the channel capacity on each link be $c_l^{max} = 4$ Mbps. At the physical

layer, for the single-diversity codes we use Alamouti codes, with code rate $r = 1/2$, hence the maximum source rate link l can support is $R_l = 2$ Mbps. For the diversity-embedded codes, we use the codes in [4] with two rate-reliability pairs, hence each link is composed by two sub-links. The code rates for the high and low reliability sub-link are $r_H = 1/6$ and $r_L = 3/8$, respectively. The maximum source rates the high and low reliability sub-link can support are $R_{l,H} = 0.667$ Mbps and $R_{l,L} = 1.5$ Mbps, respectively.

For performance of codes over a single link, [4] gives the bit-error-rate (BER) performance with respect to signal-noise-ratio (SNR) over quasi-static fading channel. Note that in [4], for fair comparison, the diversity-embedded codes have the same average signal power as the single-diversity codes. The packet error rate p is approximated as $p = 1 - (1 - \text{BER})^K$, assuming that each bit is independent, where K is the packet length. Fig. 3 shows the packet error rate for $K = 400$. Fig. 4 shows throughput on single link if the link is fully loaded, $R_l(1 - p)$ for Alamouti codes, $R_{l,H}(1 - p_H) + R_{l,L}(1 - p_L)$ for diversity-embedded codes. We can see the advantage of diversity-embedded codes.

We set the VoIP R-factor saturation threshold of packet loss p_{thr} to be 10^{-3} . From Fig. 3, $p_H \leq p_{thr}$, so we send voice over the high-reliability link, and data over both sub-links. The VoIP source codec is set to be G729, where for simplex data, the rate is 40 Kbit/s and the packet size $K = 400$ bits. For data traffic, $K = 400$ as well. The constant parameters have the following values: $x_s^{min} = 0.1$ Mbps, $x_s^{max} = 2$ Mbps, $\gamma_s^{min} = 53.3$, $\gamma_s^{max} = 100$, $\theta_s^{min} = 0$, $\theta_s^{max} = 2$ Mbps, $\delta_s^{max} = 300$ ms, and $\delta_s^{min} = \delta_s^{min} = 0$.

A quick summary of the key engineering implications from the graphs, before we discuss each graph:

- 1) The superiority of diversity-embedded codes in terms of network utility is shown in Fig. 5-8 (upper two curves in Fig. 5, right-upper two curves in Fig. 6, right four curves in Fig. 7 and lower two curves in Fig. 8).
- 2) The advantage of priority queueing scheme over capacity division is shown in Fig. 6 and Fig. 7 (compare any two solid lines or two dashed lines, either in the left or right cluster).
- 3) The impact of ARQ that it enhances the reliability but increases the delay is shown in Fig. 5 (upper first v.s. second, third v.s. fourth) and Fig. 7 (dashed lines v.s. solid lines).

For the case of assigning VoIP traffic a fixed capacity $B = 60$ Kbps, we optimize the data traffic through a weighted sum of throughput and delay. Fig. 5 shows the tradeoff between throughput and delay for data traffic, at SNR=20dB. The tradeoff is made by adjusting the weight of throughput from 0 to 1. We can see that the network with diversity-embedded codes has larger achievable throughput-delay region than single-diversity codes. ARQ improves the reliability but has larger delay, which can be seen if we compare the

throughput and delay for the same weight. Also for the same weight, the network with diversity-embedded codes has larger average packet delay, which is not surprising, because diversity codes have two narrower ‘pipes’. But the network with diversity-embedded codes can be designed to add more weight to delay, to push the operating point to the place of less delay as well as larger throughput. Fig. 5 also shows the advantage of the diversity-embedded codes when there is only one traffic type in the network. Such advantage is due to the freedom that we can allocate every user’s traffic to the sub-links.

In Fig. 6, we show the tradeoff between VoIP traffic and delay insensitive data traffic. Here data traffic utility is throughput only. For capacity division, we can see the tradeoff as the capacity assigned to voice B varies, and for priority queueing, the tradeoff is shown as the weight of voice traffic in the objective function varies. This figure shows that the network with diversity-embedded codes has better tradeoff.

Fig. 6 also shows that the priority queueing can achieve higher R-factor of voice traffic and higher throughput of data traffic. The priority queueing may raise the packet delay of data traffic, but since we consider delay insensitive data in Fig. 6, the enlarged delay of data packet doesn’t hurt. In addition, we can see that the achieved R-factor by priority queueing is not sensitive to the weight of the voice v . This is because the voice packet can get very small average packet delay due to higher priority, no matter how large the data throughput is. For the same reason, the throughput of data traffic can be as large as possible and insensitive to v except when v is close to one where the system only cares about voice R-factor.

In Fig. 7 and 8, we set the weight of the voice v in the objective function as $v = 0.5$, and for data traffic, different sources have different requirements for delay, indicated by the weight w_s in the objective function and the maximal delay constraints. The weight parameters on throughput for sources are $w_1 = w_3 = 0.7$, and $w_2 = w_4 = w_5 = 0.95$. The maximal delay parameters are $\delta_1^{max} = \delta_3^{max} = 300$ ms, and $\delta_2^{max} = \delta_4^{max} = \delta_5^{max} = 2000$ ms. The capacity division scheme uses a fixed capacity assigned to VoIP traffic, $B = 43$ Kbps.

In Fig. 7, we show the tradeoff of data and voice utilities at different SNR’s. To compare two types of codes, we push the network with diversity-embedded codes to operate at the point where the packet delay is almost the same or less compared with single-diversity codes, by multiplying the throughput in the objective function by a constant 0.4, which reduces the delay and throughput, but the throughput is still larger than the single-diversity case.

Fig. 8 shows the average data packet delay, for capacity division. The approximation of the delay formula is good, as shown in the figure. The average data packet delay for priority queueing is similar to the delay

in this figure, but with larger approximation error of the delay. At high SNR, the delay of the ARQ case is reduced because of the higher reliability. In summary, Fig. 7 and 8 show that the network with diversity-embedded codes can operate concurrently at higher throughput and less delay for data, as well as higher R-factor for VoIP, than the single-diversity codes.

The diversity-embedded codes can help reduce the delay of ARQ case compared with the single-diversity codes. By multiplying the throughput in the objective function by a constant 0.4 for the diversity-embedded codes, the delay of ARQ case can be smaller than no ARQ case when SNR=24 dB, as in Fig. 7. For the same example as in Fig. 7 and 8, suppose we do not multiply the throughput in the objective function by 0.4, Fig. 9 shows the ratio of the average delays for ARQ case and for no ARQ case. We can see that the diversity-embedded codes can help reduce the delay of ARQ case. This is because that when some of the links in a network are not heavily loaded, the diversity-embedded codes provide the freedom to allocate the traffic to the sub-link with higher reliability as much as possible, therefore the overall reliability is improved, hence the chance of retransmission and average delay are both reduced.

The following example shows that the advantage of diversity-embedded codes can be enlarged in the network setting compared with single link setting. Consider the same network topology (with the same parameters, such as capacity, etc.) as in Fig. 2, but with only two flows, user 1 from node V_1 to V_4 , and user 2 from V_5 to V_6 , each with delay insensitive data traffic only. By the symmetry, the rate of each user can be half of the link capacity. Fig. 10 shows the ratio of the average reliabilities $\frac{1}{|S|} \sum_s \rho_s$ for the case of diversity-embedded codes and single-diversity codes, for this network and for a single link, respectively. We can see that the diversity-embedded codes can increase the average reliability in this network more than on a single-link. The reason is again that for a single link, it is heavily loaded to get large throughput; but for a network, some links may not be heavily loaded, so we can assign the traffic to the sub-link with higher reliability as much as possible, hence the overall average performance can be improved.

VI. CONCLUSIONS

Motivated by the innovation at physical layer that new channel codes, such as diversity-embedded codes, can achieve rate-reliability points not achievable by simple switching strategies, we investigate utility maximization for networks where each logic link has multiple sub-links with different rate-reliability characteristics. We incorporate rate, reliability and delay into the utility function, which reflects the various QoS requirements of different applications. We also jointly control the source rates, forward error correction, and ARQ to amplify the benefits of new codes to delay-sensitive network applications.

We answer the question on how to allocate the physical layer resources to a mixture of traffic types, including VoIP and data traffic, with appropriate rate-reliability characteristics so that their utilities are jointly maximized. Distributed algorithms are proposed for the optimal rate-reliability-delay tradeoff in the networks using capacity division and those using priority queueing to do QoS differentiation. Compared with the traditional channel codes, the new diversity-embedded codes can provide higher network utilities for each traffic type *simultaneously*, for either cases with or without ARQ. Having ARQ allows us to further tradeoff reliability with delay. We also show that the (more complicated) priority queueing scheme attains higher total network utility than the capacity division scheme. In addition, we show that the diversity-embedded codes can work even better in the network setting than the single link setting, especially to improve the average reliability, and to help reduce the delay for a system with ARQ.

REFERENCES

- [1] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [2] D. Bertsekas and R. Gallager, *Data networks*. Prentice Hall, 1992.
- [3] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of IEEE*, Jan. 2007.
- [4] J. Chui and A. R. Calderbank, "Embedded diversity codes using peaky signaling," Technical report, Princeton University, 2007.
- [5] S. N. Diggavi, N. Al-Dhahir and A. R. Calderbank, "Diversity embedded multiple antenna communications," *AMS edited volume on "Network Information Theory"*, edited by P. Gupta, G. Kramer and A. J. van Wijngaarden, AMS volume 66, Series on Discrete Mathematics and Theoretical Computer Science, pp. 285-302, 2004.
- [6] S. N. Diggavi, A. R. Calderbank, S. Dusad and N. Al-Dhahir, "Diversity embedded space-time codes," to appear, *IEEE Trans. on Inform. Theory*, Jan. 2008.
- [7] S. N. Diggavi and D. Tse, "Fundamental limits of diversity-embedded codes over fading channels," *IEEE ISIT*, Sept. 2005.
- [8] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Optimal throughput-delay scaling in wireless networks: part I: the fluid model," *IEEE Trans. on Inform. Theory*, vol. 52, no. 6, pp. 2568-2592, June 2006.
- [9] H. El Gamal, G. Caire and M. O. Damen, "The MIMO ARQ channel: diversity-multiplexing-delay tradeoff," *IEEE Trans. on Inform. Theory*, vol. 52, no. 8, 3601-3621, Aug. 2006.
- [10] T. Holliday, A. Goldsmith and H. V. Poor, "The impact of delay on the diversity, multiplexing and ARQ tradeoff," *Proceedings of IEEE ICC2006*, Istanbul, June 2006.
- [11] ITU-T Recommendation G.107, *The Emodel, a computational model for use in transmission planning*.
- [12] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, pp. 237-252, March 1998.
- [13] L. Kleinrock, *Queueing Systems*, Vol.1: Theory, Vol. 2: Computer Applications. New York: Wiley, 1976.
- [14] J. W. Lee, M. Chiang, and A. R. Calderbank, "Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization," *IEEE J. Sel. Area Comm.*, vol. 24, no. 5, pp. 962-976, May 2006.
- [15] Y. Li, M. Chiang, A. R. Calderbank and S. N. Diggavi, "Optimal rate-reliability-delay tradeoff in networks with composite links," *Proceedings of IEEE INFOCOM*, May 2007.

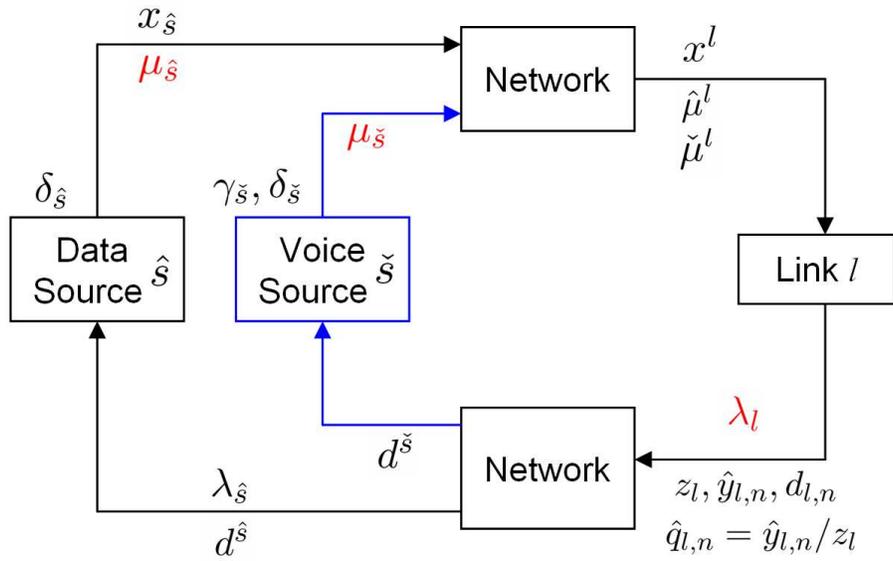


Fig. 1. Illustration of Algorithm 2.

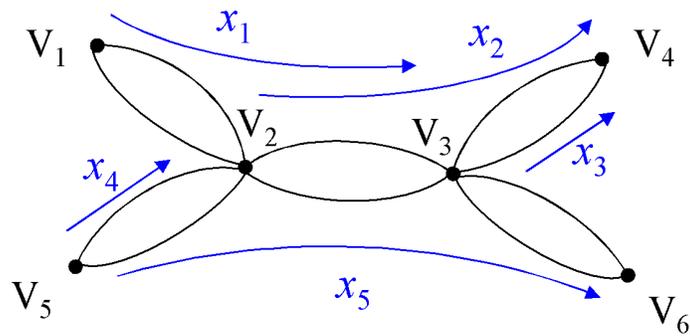


Fig. 2. Network topology and flow routes in numerical examples.

- [16] J. Pongsajapan and S. H. Low, "Reverse engineering TCP/IP-like networks using delay-sensitive utility functions," *Proceedings of IEEE INFOCOM*, May 2007.
- [17] S. Stidham, Jr., "Pricing and congestion management in a network with heterogeneous users," *IEEE Trans. on Auto. Control*, vol. 49, no. 6, pp. 976-981, June 2004.
- [18] S. B. Wicker, *Error control systems for digital communication and storage*. Prentice Hall, 1995.

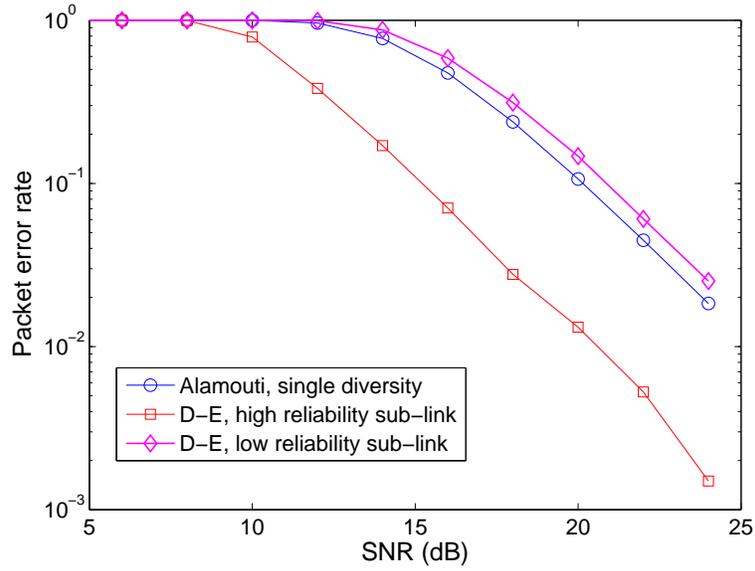


Fig. 3. Packet error rate. Alamouti (single-diversity), D-E: Diversity-Embedded.

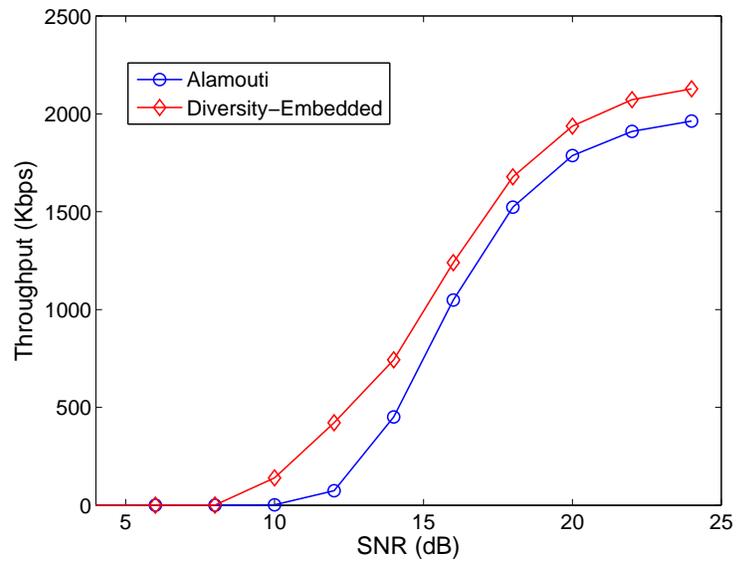


Fig. 4. Single link throughput.

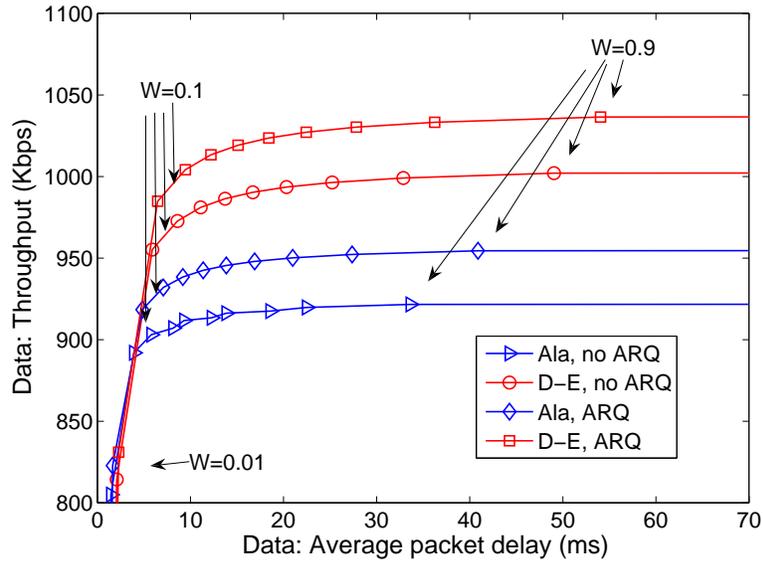


Fig. 5. Tradeoff between throughput and delay. Ala: Alamouti (single-diversity), D-E: Diversity-Embedded.

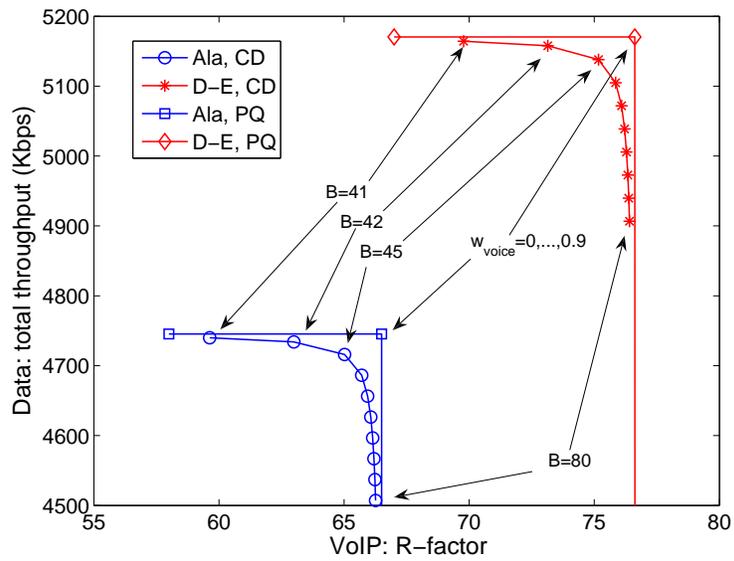


Fig. 6. Tradeoff between voice traffic and delay insensitive data traffic. Ala: Alamouti (single-diversity), D-E: Diversity-Embedded, CD: Capacity Division, PQ: Priority Queueing.

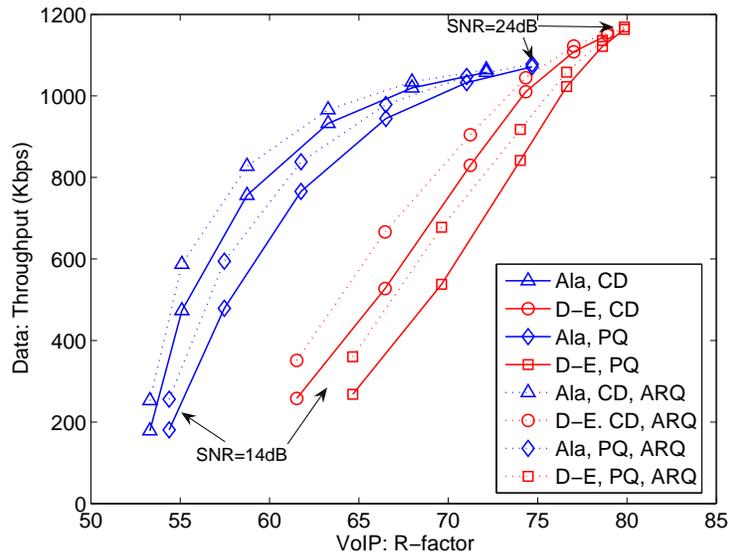


Fig. 7. Tradeoff between voice traffic and delay sensitive data traffic. Ala: Alamouti (single-diversity), D-E: Diversity-Embedded, CD: Capacity Division, PQ: Priority Queueing.

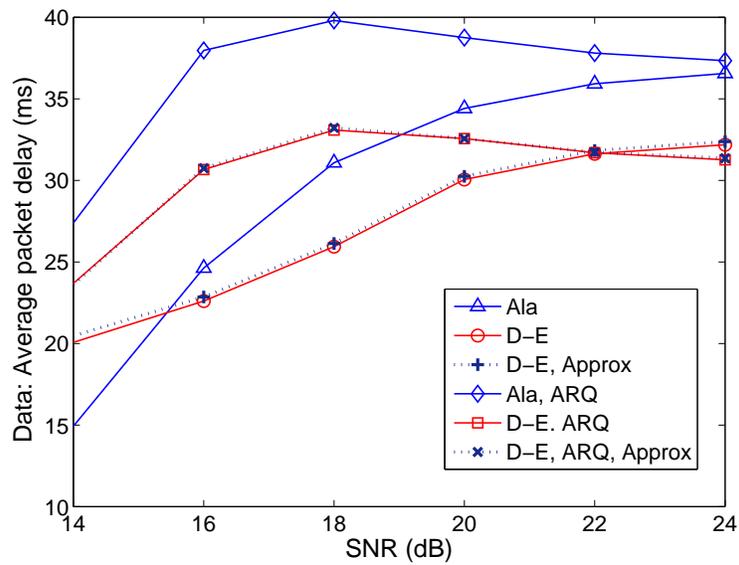


Fig. 8. Average data packet delay and delay approximation. Ala: Alamouti (single-diversity), D-E: Diversity-Embedded, Approx: Approximation.

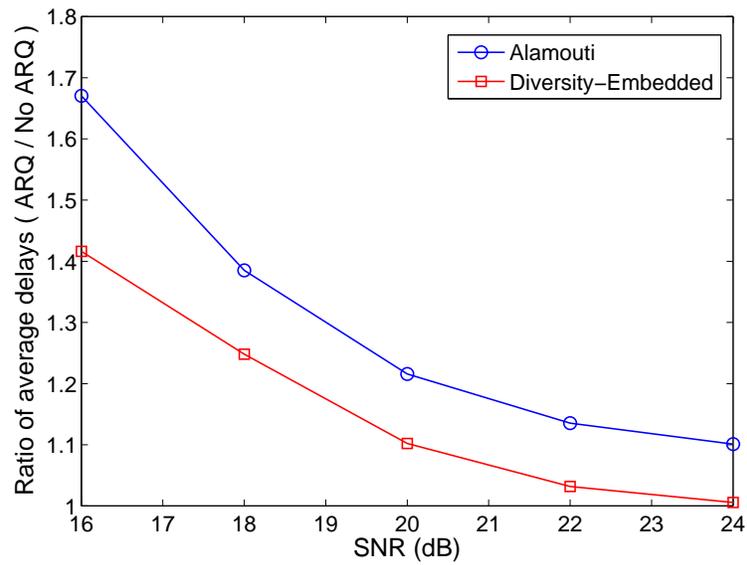


Fig. 9. Improvement of the average delay.

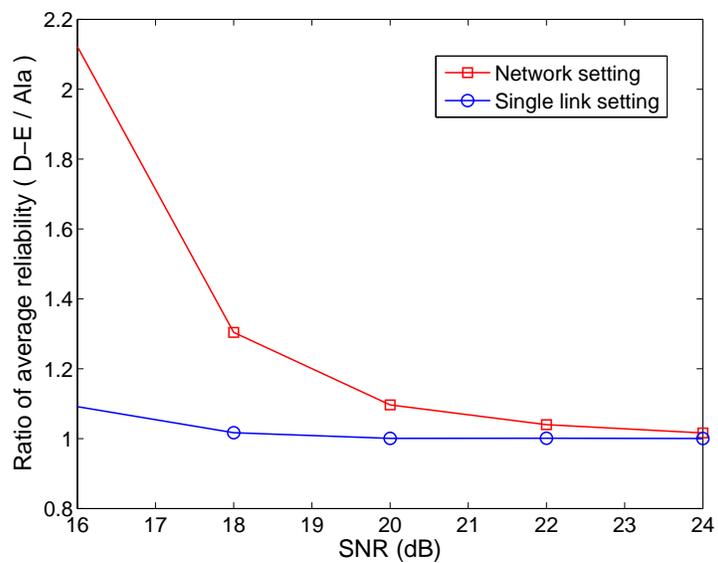


Fig. 10. Enhanced average reliability. Ala: Alamouti (single-diversity), D-E: Diversity-Embedded.