MP-DSM: A Distributed Cross Layer Network Control Protocol

Daniel C. O'Neill, Yan Li, and Stephen Boyd
Department of Electrical Engineering
Stanford University
dconeill, liyan, boyd@stanford.edu

***Abstract* –** We present a distributed cross layer approach to controlling network performance under various QoS requirements in interference limited systems. The interaction between different layers of the OSI protocol stack requires a cross layer approach in order to optimally allocate the resources of the network. The Message Passing Direct Step Method presented here is an adaptive and distributed algorithm that achieves maximum network performance. Using the forward and backwards networks, this algorithm finds the left and right Perron Frobenius eigenvectors for the system and automatically adjusts the operating point of the system (data rates, link rates and transmitter powers) to their optimal values while satisfying QoS constraints. The approach is developed and simulated using TCP Reno.

***Keywords* – network management, TCP, optimization and control, QoS, cross layer, utility functions**

I. Introduction

WIRELESS network performance is complicated by the interaction among different protocols operating at different layers of the OSI protocol stack and various QoS requirements. This is particularly true for the interaction between protocols at the transport, network, and physical layers and QoS constraints on rates, transmitter powers and delay. The Direct Step Method, *DSM*, which captures this interaction and controls system performance under TCP or other protocols has been described in [1]. DSM finds optimal routes, rates, and transmitter powers in multi-hop networks and has recently been extended to multicasting applications [2]. This paper extends this work by describing a distributed, message passing based algorithm termed Message Passing Direct Step Method, *MP-DSM*.

This paper focuses on a *Distributed Cross Layer*, *DCL*, approach to network control. The approach allocates network resources of source data rates, link rates, and transmitter powers to adaptively optimize overall network performance under various QoS requirements. Data sources are taken as operating under protocols that respond to the network congestion (or other network attributes) by adjusting their data rates. Likewise wireless links adjust their transmitter powers and link rates to best carry the packet traffic. QoS requirements are included in the formulation by suitably modifying functions representing source protocols.

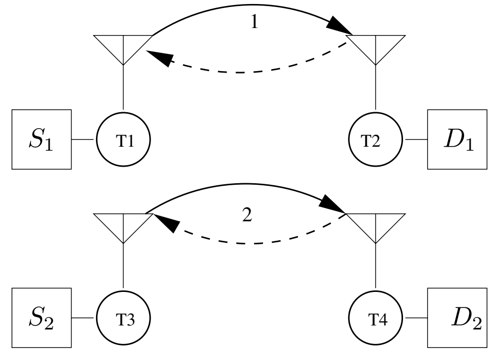

Fig. 1. Conceptual Wireless Network. Sources S_1 and S_2 send packets to destinations D_1 and D_2, which respond by sending ACKs

II. Wireless Network

A cross layer approach is used to model the network. Figure 1 depicts the network, which includes both the sources of packet data and the wireless links which transport them. Sources S_1 and S_2 send data packets to destinations D_1 and D_2 respectively and D_1 and D_2 send acknowledgements (ACK) to those sources. Sources are governed by protocols that respond to the network congestion or other network parameters. Both TCP Vegas and TCP Reno are examples of such protocols by using window congestion control to modify the average data rate sent by a source.

The unidirectional wireless links are shown as solid and dashed arrows corresponding to the forward network and the backwards network respectively. Each forward link and the corresponding backwards link is numbered $i = 1 \cdots N$. The forward network is the set of links used by packets traversing the network from source to destination. The reverse network is used by destinations

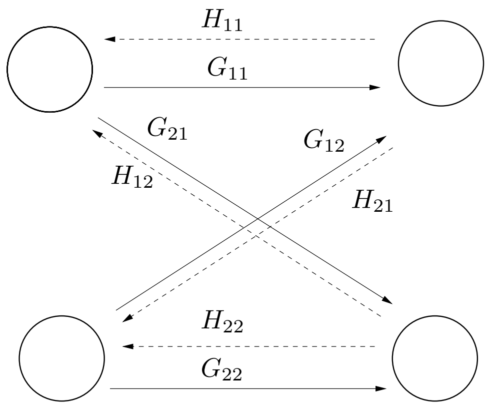

Fig. 2. Forward and backwards network

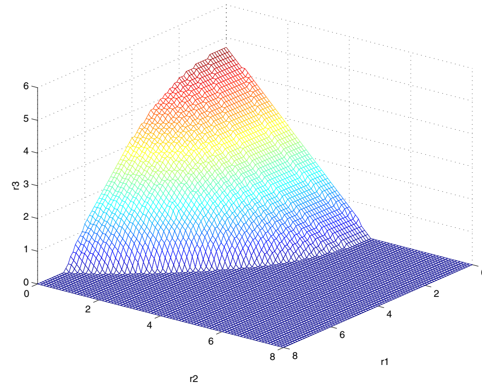

Fig. 3. Network Rate Region

to acknowledge receipts of packets. Each link in both directions is assumed to operate at variable transmitter powers and variable data rates. These rates are termed the link rates, and model the lowest layer of the OSI protocol stack.

Each forward link i can transmit up to the link rate, R_i, bits per second. As shown in Figure 2, each link i is also associated with a transmission gain G_{ii} that relates the power of link i's transmitter, p_i, to the power received by the link's receiver, $G_{ii}p_i$. The interference induced by the transmitter on link i at the receiver on link j is given by $G_{ij}p_j$. CDMA is assumed through out the paper, and the gains G_{ii} are assumed to capture coding gain, antenna gain, etc.

The network is taken to be interference limited, with link i's Signal to Interference Ratio, ρ_i, given as the following:

$$\rho_i \quad = \quad \frac{G_{ii}P_i}{\sum_{j\neq i} G_{ij}p_j}. \tag{1}$$

The link rate is defined as [3]

$$R_i = \ln(\rho_i). \tag{2}$$

This approximation has been justified on both empirical and analytical grounds for interference limited systems operating at moderate values of ρ_i.

Figure 2 also depicts the backwards network. The H_{ii} and H_{ij} fill roles identical to those played by the G_{ii} and G_{ij} in the forward network. Link i's backwards transmitter power is defined as u_i, its backwards SIR as, $\acute{\rho_i}$, and its backwards link rate as $\acute{R_i}$.

In many systems the forward and backwards networks will share the same system bandwidth using a time division protocol or in other cases operate on different but nearby frequencies. In cases where TDMA is used and time slots are short relative to changes in the wireless environment or where FDMA is used and the separate frequencies used are close enough, $H_{ij} \approx G_{ji}$ as can be seen in figures. Consequently the network gain matrix G for the forward matrix and the gain matrix for the backwards network are related by $H = G^T$.

Source i injects packet traffic into the network at source rates r_i. A feasible source rate must be less than the associated link rate. This can be written as

$$r_i \leq R_i \quad \Leftrightarrow \quad p \geq D(r)\tilde{G}p \tag{3}$$

where $D(r) = \textbf{diag}(\frac{e^{r_i}}{G_{ii}})$, $\tilde{G}$ is the gain matrix, G, with zero's along the diagonal, and p is the vector of transmitter powers.

Destinations inject acknowledgements into the backwards network at rate $\acute{r}$, where $\acute{r_i}$ is the ACK rate for the destination i. For simplicity each source is assumed to have a unique destination. As in the forward network case,

$$\acute{r_i} \leq \acute{R_i} \quad \Leftrightarrow \quad u \geq D(\acute{r})\tilde{G^T}u \tag{4}$$

where u is the vector of transmitter powers in the reverse network. Because $H = G^T$ the backwards network is also the adjoint network of the forward network when $\acute{r} = r$.

III. Network Characteristics

A source rate r is feasible if it is possible for the forward network to simultaneously support r_i $\forall i$; that is there exists a p such that equation 3 is satisfied. Formally the rate region is defined as

$$\mathcal{R} = \{r \in \mathbf{R}_+^n | r \leq R(p) \text{ for some } p > 0\}, \tag{5}$$

where $r \leq R$ for two vectors means component-wise inequality, i.e. $r_i \leq R_i$ for all i. Figure 3 shows the rate region for a simple three link network.

As shown in [1] the forward rate region $\mathcal{R}$ is convex [4]. By *Perron Frobenius* matrix theory the region can also be described by

$$\mathcal{R} = \{r | \lambda_{\text{pf}}(D(r)\tilde{G} \leq 1\} \tag{6}$$

where $\lambda_{\rm pf}(D(r)\tilde{G})$ is the Perron Frobenius eigenvalue associated with the matrix $D(r)\tilde{G}$. The eigenvalue $\lambda_{\rm pf}(D(r)\tilde{G})$ can be interpreted as a capacity utilization measure for the system. The Perron Frobenius eigenvalue is monotonic increasing in r. When r is increased to $\lambda_{\rm pf}(D(r)\tilde{G}) = 1$, system capacity is fully utilized and no additional traffic can be carried by the system. Further, transmitter powers p satisfying equation 2 are given by the right Perron Frobenius eigenvector of $D(r)\tilde{G}$. There may exist other power vectors $\hat{p}$ that satisfy equation 3, but $p \leq \hat{p}$. The left Perron Frobenius eigenvector is denoted by q and can be interpreted [1] as moderating the effect of power increases in the network. The backwards or adjoint network has properties similar to the forward network.

Theorem 1: The forward and backwards rate regions are identical.

Proof: Let r be a feasible point in $\mathcal{R}$, then there exists q such that

$$\begin{aligned} q^T \geq q^T D(r)\tilde{G} \quad &\Leftrightarrow \quad q \geq \tilde{G}^T D(r) q \\ &\Leftrightarrow \quad u \geq D(r)\tilde{G}^T u \end{aligned} \tag{7}$$

where $u = D(r)q$ identifies transmitter powers in the backwards network. Thus r is feasible for the adjoint network. A similar argument shows that if $\acute{r}$ is feasible for the backwards network it is also feasible for the forward network. ■

Thus the forward and backwards networks can always be operated at identical source rates r. Further, the left Perron Frobenius eigenvector q can always be found from the backwards network when both networks are operated at the same rate, explicitly,

$$q = D(r)^{-1} u. \tag{8}$$

Because $D(r)$ is diagonal and positive, the inverse always exists. Similar results hold for the backwards network.

IV. Network Performance Metrics

System performance is modelled by a performance metric or utility function U that represents the value of rate vector r to the system. This function can represent the utility a user derives from using the system at a particular rate or can be implied by data protocols, service rate agreements, or other system level metrics [5], [6]. Implicitly many protocols can be described in this way. TCP in particular has been modelled as a concave function of the source rates [7].

In many situations U can be treated as a separable function,

$$U(r) = \sum_{i=1}^{N} U_i(r_i) \tag{9}$$

where source i has performance function $U_i(r_i)$. In this view each source is concerned with only its own rate r_i. By assumption, a higher data rate is valued at least as much as a lower data rate and there is a diminishing return to additional data rate, U_i is a concave function of r_i.

Data traffic controlled by TCP-Reno, for example, can be modelled [5] using

$$U_i(r_i) = \frac{2}{D_s} \ln(2 + r_i D_s) \tag{10}$$

where D_s is the round trip delay. Voice traffic can also be modelled

$$U_i(r_i) = \begin{cases} -\infty, & r_i < R_{\min_i} \\ c, & r_i \geq R_{\min_i}. \end{cases} \tag{11}$$

Rates below $R_{\min_i}$ will never be selected by the system.

QoS constraints can be added to this formulation by embedding them in the U_i. A rate threshold for TCP data traffic might be formulated as

$$U_i(r_i) = \frac{2}{D_s} \ln(2 + r_i D_s) + \Delta \ln(r_i - R_{\min_i}) \tag{12}$$

where Δ is a constant. The second term acts a barrier preventing the network from allocating less than $R_{\min_i}$ to this source. Delay constraints can also be added to this formulation.

Since the backwards network is used to acknowledge traffic sent from the forward network, its performance is indirectly linked to U. In what follows U is assumed to have continuous second derivatives.

V. Problem Formulation

A cross layer approach is used to control the network. The goal is to adaptively control the network and operate it at the best set of source rates r and transmitter powers p such that the system performance metric or utility function U is maximized.

As shown in [1] optimal system performance lies along the rate surface. At the optimal source rates r^*, $\lambda_{\rm pf}(D(r^*)\tilde{G}) = 1$. Thus only the surface of the rate region need to be searched to find the optimal operating point for the network. Formally, this problem can be expressed as:

$$\begin{array}{ll} \text{maximize}_r & U(r) \\ \text{subject to} & \lambda_{\rm pf}(D(r)\tilde{G}) = 1 \end{array} \tag{13}$$

This is a convex problem in the variable r since the constraint set is convex and the objective concave. Each $r \in \mathcal{R}$ is associated with one or more power vectors p, and optimum p^* are calculated from r^*.

The backwards network is not explicit in this formulation but is assumed to be operated at a rate $\acute{r} \in \mathcal{R}$ sufficient to support the forward rate r.

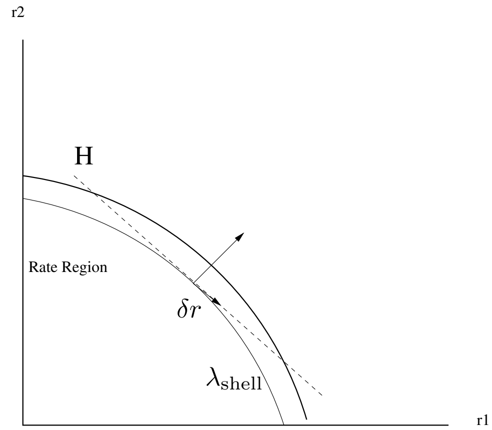

Fig. 4. MP-DSM operates inside the rate region along λ_{shell}

VI. Distributed Method of Solution

This section describes a message passing based and distributed algorithm, MP-DSM, that calculates the optimal source rates, link rates, and link transmitter powers. The algorithm is based on the Direct Step Method. The algorithm uses the backwards network to calculate quantities needed to find the optimal operating point. Simple messages are periodically shared among links in the network, but power and link rate are computed locally. In particular three types of messages are shared within the network. The first is the marginal aggregate performance of the network, $1^T\nabla_r U(r)$, the second is the marginal utility of the network, $1^T\nabla_r\lambda_{\text{pf}}$ and the third is the network's total transmitter powers in both forward and reverse directions. We assume the number of links and users, and gain matrix G are fixed. This need not be the case, as the iterative nature of the algorithm will adapt to these changes.

The MP-DSM operates entirely within the feasible rate region to make it possible for the network to compute $\lambda_{\text{pf}}(D(\hat{r})\tilde{G})$, r, p, and q. Figure 4 illustrates the concept. The network is operated slightly below capacity at λ_{shell}. Both the forward and backwards networks operates at the same source rates r. The detail of the algorithm is as follows:

Assume discrete time and t is the current system time. Then at time $t+1$:

Step 1. Update source rate vector $r(t)$ to $\hat{r}(t+1)$ along a feasible ascent direction on the tangent **H** of λ_{shell} surface to decrease the normalized error between $\nabla_r U(r)$ and $\nabla_r\lambda_{pf}(D(r)\tilde{G})$. The ith element of $r(t)$ is updated on each link locally. It conceptually follows the DSM method, but adapts to ensure $\hat{r}(t+1)$ lies within the feasible rate region. As shown in [1], the network performance is optimal at r^* when $\nabla_r\lambda_{\text{pf}}(D(r^*)\tilde{G})$ is parallel to $\nabla_r U(r^*)$. At the surface of λ_{shell}

$$\begin{aligned}\nabla_r\lambda(D(r)\tilde{G}) &= \left[\frac{q_i e^{r_i}[\tilde{G}p]_i}{G_{ii}},\cdots\right]^T \\ &= \left[u_i[\tilde{G}p]_i,\cdots\right]^T\end{aligned} \tag{14}$$

where $[\tilde{G}p]_i$ is the forward interference power on the ith link. The ith element of $\nabla_r\lambda_{\text{shell}}$ can be found from the product of the transmitter power in the backwards direction and the interference in the forward direction. Both of these values are known to the receiver in the forward direction, which is also the transmitter in the backwards direction. Transmitters in the forward direction can also obtain these values from receivers. Thus each element of $\nabla\lambda(D(\hat{r})\tilde{G})$ can be calculated locally at each link.

Given $1^T\nabla_r U(r)$, $1^T\nabla_r\lambda_{\text{pf}}$ are available through message passing, the feasible ascent direction can be calculated locally as:

$$\delta r_{\text{ds,i}} = -\alpha\frac{1}{[\nabla\lambda]_i}\left(\frac{[\nabla\lambda]_i}{\mathbf{1}^T\nabla\lambda} - \frac{[\nabla U]_i}{\mathbf{1}^T\nabla U}\right) \tag{15}$$

And each element of source rate vector is updated as $\hat{r}_{\text{i}}(t+1) = r_{\text{i}}(t) + \delta r_{\text{ds,i}}(t)$ locally. As intuition would suggest, a step size α can be chosen so that $\hat{r}(t+1)$ lies within the rate region.

Theorem 2: The estimated update vector $\hat{r}(t+1)$ lies in the feasible rate region, for small enough α.

Step 2. Iteratively calculate the transmitter powers p and u associated with source rate $\hat{r}(t+1)$. This procedure asymptotically converges to the eigenvectors for the forward and backwards network yielding p and u. At each iteration $l = 0,\cdots,M$, the transmitter at each link i adjusts its transmitter power to match a scaled version of the interference received at link i 's receiver. This interference information is passed to transmitters by the receivers. At each link i, the iteration is given by

$$\begin{aligned}p_i^{l+1} &= D(\hat{r}(t+1))_{ii}[\tilde{G}p^l]_i \\ u_i^{l+1} &= D(\hat{r}(t+1))_{ii}[\tilde{G}^T u^l]_i\end{aligned} \tag{16}$$

where $D(\hat{r}(t+1))_{ii} = \frac{e^{\hat{r}(t+1)}}{G_{ii}}$. The starting powers $p^0 = p(t)$ are the powers in use by the system at the end of time t, and are always positive.

Theorem 3: The vector p converges to the Perron Frobenius eigenvector of $D(r)\tilde{G}$ and $\mathbf{1}^T D(r)\tilde{G}p$ converges to the associated eigenvalue.

The resulted power vectors p and u are normalized using message passing.

$$\begin{aligned}p(t+1) &= \frac{p^{M+1}}{\mathbf{1}^T p^{M+1}} \\ u(t+1) &= \frac{u^{M+1}}{\mathbf{1}^T u^{M+1}}\end{aligned} \tag{17}$$

Step 3. The network executes an additional iteration of equation 16 and then uses message passing to compute $\lambda_{\text{pf}}(D(\hat{r}(t+1))\tilde{G}) = \mathbf{1}^T D(\hat{r}(t+1))\tilde{G}p(t) = \mathbf{1}^T p$.

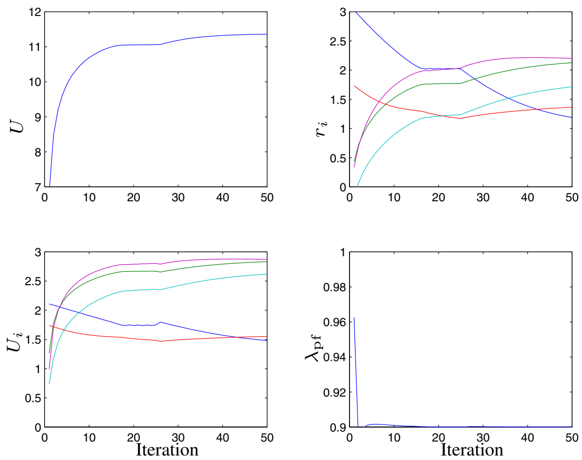

Fig. 5. System performance, link rates and $\lambda_{\text{pf}}(D(\hat{r})\tilde{G})$

Step 4. The value $\lambda_{\text{pf}}(D(\hat{r}(t+1))\tilde{G})$ is broadcast throughout the network, and each source corrects its source rate back onto the λ_{shell} surface as follows:

$$r(t+1) = [\hat{r}(t+1) + \ln(\frac{\lambda_{\text{shell}}}{\lambda_{\text{pf}}(D(\hat{r}(t+1))\tilde{G})})]^+. \quad (18)$$

Thus a single variable $\lambda_{\text{pf}}(D(\hat{r})\tilde{G})$ controls the utilization of the system and controls aggregate and individual demand for source rate bandwidth r.

VII. SIMULATION

A data network using TCP Reno is considered in this section. Time is discrete, and the simulations run from $t = 1, \ldots, 50$. The network has 5 sources and 5 links. The network weighs each source equally and the system performance measure is

$$U(r) = \sum \frac{2}{D_s} \ln(2 + r_i D_s) + \Delta \ln(r_i - R_{\min_i}) \quad (19)$$

The parameter $\Delta = .01$ and $\lambda_{\text{pf}} = .9$ corresponding to 90 percent network efficiency. The gain matrix G was chosen randomly. QoS constraints require $R_{\min_1} = 2$ and $R_{\min_i} = 0 \ \forall i \neq 1$.

The simulation is shown in Figure 5. During the initial 25 time slots MP-DSM improves system utility from 7 to 11.5 units, while restricting link 1 to a minimum rate of 2. This threshold is reset to $R_{\min_1} = 0$ at step 26. The system adapts by adjusting link rates and increasing overall system utility. The initial drop in utility is an artifact of removing the constraint abruptly. As can be seen the system continuously operates inside the rate region defined by $\lambda_{\text{pf}}(D(\hat{r})\tilde{G}) \leq 1$. The initial spike to .95 reflects the very large change in r made in the first iteration of the algorithm.

VIII. SUMMARY

This paper describes MP-DSM, a distributed cross layer, DCL, approach to network control. The approach optimally controls network source rates and link rates and transmitter powers to maximize the network performance under various QoS constraints. Protocols are modelled as network performance metrics or utility functions and reflect congestion control protocols such as TCP operating at the transport layer of the OSI model. QoS requirements are also reflected in utility functions. The physical layer is modelled by link rates and transmitter powers. MP-DSM is adaptive, responding to changes in the network automatically and with out exceeding the capacity of the network. . Both the forward and backwards networks are utilized to readily find the left and right Perron Frobenius eigenvectors for the system. The associated eigenvalue is readily computed, and can be interpreted as the effective utilization of the system.

REFERENCES

[1] D. ONeill, D. Julian, and S. Boyd, "Seeking foschini's genie: Optimal rates and powers in wireless networks," *submitted to IEEE Trans. Vehicular Technology*, 2003.

[2] D. ONeill and Y. Li, "Optimal routes and flows in multicasting over ad hoc networks," *Submitted to VTC Spring 2004*, 2004.

[3] D. Julian, M. Chiang, and D. ONeill, "Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad-hoc networks," in *Proc. IEEE Infocom*, June 2002, pp. 477–486.

[4] S. Boyd and L. Vandenberghe. (2003) Convex optimization. [Online]. Available: http://www.stanford.edu/ boyd/cvxbook.html

[5] S. Low, L. Peterson, and L. Wang, "Understanding vegas: A duality model," *Journal of ACM*, vol. 49(2), pp. 207–235, Mar. 2002.

[6] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communications networks: Shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, pp. 237–252, Nov. 1998.

[7] S. Low. (2002) Duality model of tcp and queue management algorithms. [Online]. Available: http://netlab.caltech.edu