

Research Article

Experimental Evaluation of Optimal Rate Delay and Power Allocation Algorithm in Wireless Control Networks

Muhammad Tahir¹ and S. K. Mazumder²

¹Department of Electrical Engineering, University of Engineering and Technology Lahore, Lahore 54390, Pakistan

²Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA

Correspondence should be addressed to Muhammad Tahir, mtahir.dr@gmail.com

Received 6 August 2009; Revised 2 February 2010; Accepted 8 March 2010

Academic Editor: Luca De Nardis

Copyright © 2010 M. Tahir and S. K. Mazumder. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The network utility maximization (NUM) framework, widely used for wireless networks to achieve optimal resource allocation, has led to both centralized as well as distributed algorithms. We compare the convergence performance of centralized realization of the NUM framework with that of distributed realization by implementing the algorithms using a hardware test-bed. Experimental results show a superior convergence performance for centralized implementation compared to the distributed implementation, which is attributed to the dominance of communication delay over processing delay. The convergence results for the distributed case also show a tradeoff between processing time and the associated communication overhead providing an optimal termination criterion for the convergence of different subproblems.

1. Introduction

Since the seminal work by Kelly et al. [1] was published, the basic network utility maximization (NUM) problem has been extensively studied for rate allocation and congestion control in wired [2] as well as wireless networks [3–5]. Recently NUM framework has been extended to achieve optimal resource utilization for wireless control networks (WCNs¹) while meeting the stringent end-to-end delay demands of distributed control system [5]. Using the primal-dual decomposition, distributed algorithms are proposed in [1–5] to solve the resource optimization problem based on the NUM framework. This paper compares the convergence performance of a distributed algorithm with that of a centralized algorithm by implementing on a hardware test-bed. We also develop a performance evaluation model and validate its convergence performance using the experimental results. The model can be used for performance evaluation of both distributed as well as centralized realizations of different NUM frameworks.

To compare the convergence performance of distributed and centralized implementations, we have selected the

following NUM problem of [5]:

$$\begin{aligned}
 & \text{maximize} && \sum_s \log(r_s) \\
 \text{s.t.} & && d_s = \sum_{l \in L(s)} (d_l^{(t)} + d_l^{(q)}) \leq D_{\max}(s) \quad \forall s, \\
 & && \frac{0.5\mu}{c_l(\mathbf{P}) - \sum_{s: l \in L(s)} r_s} \leq d_l^{(q)}, \quad \frac{\mu}{c_l(\mathbf{P})} \leq d_l^{(t)} \quad \forall l, \\
 & && R_{\min}(s) \leq r_s \quad \forall s, \quad 0 \leq P_l \leq P_{\max} \quad \forall l.
 \end{aligned} \tag{1}$$

In (1), $d_l^{(q)}$ and $d_l^{(t)}$ are queuing and transmission delays, respectively, at link $l \in L$ (L being the set of network links and $|L|$ is the total number of links in the network), d_s represents the end-to-end delay for transmission session s , μ is the packet length, $c_l(\mathbf{P})$ given by $\log(\text{SINR}_l(\mathbf{P}))$, for high link SINR, is the capacity at link l , r_s is the end-to-end session rate for transmission session $s \in S$ with an associated shortest path consisting of a subset of links $L(s) \subseteq L$, $D_{\max}(s)$ is the end-to-end delay threshold, and $R_{\min}(s)$ is the minimum rate

threshold. In the expression for $c_l(\mathbf{P})$, $\text{SINR}_l(\mathbf{P}) = \gamma_{ll}P_l/(n_l + \sum_{m \neq l} \gamma_{lm}P_m)$ is the signal-to-interference and noise ratio, γ_{lm} is the channel gain from transmitter of link m to the receiver of link l , $P_l \in \mathbf{P}$, $P_l \leq P_{\max} \forall l$ is the transmitter power for link l , P_{\max} is the maximum transmit power level, and n_l is additive noise. We also define *transmission cycle* to be the set Θ of *transmission schedules* where each transmission schedule $\theta(i) \in \Theta$ has an associated subset of simultaneously transmitting links $L(\theta(i)) \subseteq L$.

To decompose the NUM problem in (1), we assign dual variables $\lambda_l \in \Lambda$ and $\psi_l \in \Psi$ to each of the queuing and transmission delay constraints, respectively, to form the Lagrangian [5] $L(\mathbf{r}, \mathbf{d}^{(q)}, \mathbf{d}^{(t)}, \Lambda, \Psi) = \sum_s f_1(r_s) - f_2(\mathbf{d}^{(q)}, \mathbf{d}^{(t)}) + f_3(\mathbf{P})$, where $f_1(r_s) = \log(r_s) - \sum_{l \in L(s)} \lambda_l r_s$, $f_2(\mathbf{d}^{(q)}, \mathbf{d}^{(t)}) = \sum_l \mu(\lambda_l/2d_l^{(q)} + \psi_l/d_l^{(t)})$, and $f_3(\mathbf{P}) = \sum_l (\lambda_l + \psi_l)c_l(\mathbf{P})$, respectively, represent the rate, delay, and power allocation subproblems. The associated dual problem is given by $\text{minimize}_{\lambda_l, \psi_l \geq 0} \{g(\Lambda, \Psi)\}$, where $g(\Lambda, \Psi) = \text{maximum}_{\mathbf{r}, \mathbf{d}^{(q)}, \mathbf{d}^{(t)}, \mathbf{P}} \{L(\mathbf{r}, \mathbf{d}^{(q)}, \mathbf{d}^{(t)}, \mathbf{P}, \Lambda, \Psi) \mid d_s < D_{\max}(s), R_{\min}(s) < r_s \forall s, P_l \leq P_{\max} \forall l\}$.

The test-bed used for realization is equipped with wireless transceiver [6] using binary frequency-shift-keying (FSK) with 1.5 Mbps operating in 902–928 MHz ISM band providing tunable transmitter power from 0–21 dBm with 10-bit resolution. Manchester encoding, to reduce the bit error rate due to saturation of the data slicer, as well as the fixed modulation used by the RF transceiver do not allow us to realize $c_l(\mathbf{P})$ in the form of $\log(\text{SINR}_l(\mathbf{P}))$. Instead power control is used to maximize the link packet success rate, ρ_l . With these hardware limitations, we compare the convergence performance, of the distributed and centralized implementations, using Texas Instrument's TMS320C6713 DSP platform.

2. Rate Delay and Power Allocation Algorithm

The dual decomposition of the problem in (1) leads to the following rate, delay, and power allocation subproblems:

$$\text{maximize}_{R_{\min}(s) \leq r_s} \sum_s \left(\log(r_s) - \sum_{l \in L(s)} \lambda_l r_s \right), \quad (2)$$

$$\text{minimize}_{d_s \leq D_{\max}(s)} \sum_l \mu \left(\frac{\lambda_l}{2d_l^{(q)}} + \frac{\psi_l}{d_l^{(t)}} \right), \quad (3)$$

$$\text{maximize}_{P_l \leq P_{\max}} \sum_l (\lambda_l + \psi_l)c_l(\mathbf{P}). \quad (4)$$

2.1. Rate Allocation Subproblem. The rate-allocation subproblem in (2) is further primal decomposable and can be solved individually for each session s at the source nodes. The choice of the “log” utility function and the resulting structure of rate allocation problem make it possible to have a closed form solution using KKT conditions. But we opt for an iterative solution, as choosing a different utility function may lead to a situation for which closed form solution is not available. Since the objective function for the rate allocation

subproblem is differentiable, projected gradient method [7] is used leading to the following rate updates for each r_s :

$$r_s(k+1) = [r_s(k) + \beta_r \nabla f_1(r_s)]^+. \quad (5)$$

In (5), β_r is the fixed step size and $[x]^+$ is defined as $\max\{R_{\min}(s), x\}$. Using the definition of $\nabla f_1(r_s)$ in (5) and evaluating the gradient, we obtain $r_s(k+1) = [r_s(k) + \beta_r(1/r_s - \sum_{l \in L(s)} \lambda_l)]^+$. We would like to point out that compared to the gradient method an algorithm based on the Newton's method or an interior point method will further improve the convergence performance.

2.2. Delay Allocation Subproblem. Using the gradient projection method for convex objective with linear inequality constraints [7], we obtain the updates for the queuing and transmission delay-allocation subproblems given by

$$\begin{aligned} \mathbf{d}^{(q)}(k+1) &= \mathbf{d}^{(q)}(k) - 0.5\beta_q \mathbf{\Pi} \left(\nabla_q f_2(\mathbf{d}^{(q)}, \mathbf{d}^{(t)}) \right), \\ \mathbf{d}^{(t)}(k+1) &= \mathbf{d}^{(t)}(k) - \beta_t \mathbf{\Pi} \left(\nabla_t f_2(\mathbf{d}^{(q)}, \mathbf{d}^{(t)}) \right). \end{aligned} \quad (6)$$

In (6), β_q and β_t are the fixed step sizes and $\mathbf{\Pi}$ is the projection matrix obtained from $\mathbf{\Pi} = \mathbf{I} - \mathbf{D}(\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T$, where \mathbf{I} is the identity matrix, \mathbf{D} is $2|L| \times m$ matrix with $m < 2|L|$, and m is equal to the number of active set of constraints. The columns of \mathbf{D} are the gradients of the active set of end-to-end delay constraints and are obtained from $\mathbf{D}^T[\mathbf{d}^{(q)}, \mathbf{d}^{(t)}]^T = \mathbf{D}_{\max}$. Since the end-to-end delay constraints are linear, the matrices \mathbf{D} and $\mathbf{\Pi}$ do not change at each iteration and need not to be computed at each iteration.

2.3. Power Control Subproblem. The power allocation subproblem in (4) is not convex but can be transformed into a convex problem by “log” transformation, and then is solved efficiently by power-update with fixed step size β_p [3, 4] as

$$\begin{aligned} P_l(k+1) &= P_l(k) + \beta_p \left(\frac{\lambda_l + \psi_l}{P_l(k)} - \sum_{j \neq l} \frac{\gamma_{jl}(\lambda_j + \psi_j)}{\sum_{m \neq j} \gamma_{jm}P_m(k) + n_j} \right). \end{aligned} \quad (7)$$

In the implementation of the power updates, we assume that there is not a considerable change in the channel gains $\gamma_{jl} \forall j, l$ for durations comparable to convergence time of the algorithm.

2.4. Dual Problem. For the dual problem of minimizing $g(\Lambda, \Psi)$ s.t. $\lambda_l, \psi_l \geq 0 \forall l$ [6], we observe that $(c_l(\mathbf{P}) - \mu/d_l^{(t)})$ and $(c_l(\mathbf{P}) - \sum_{s: l \in L(s)} r_s - 0.5\mu/d_l^{(q)})$ are subgradients of $g(\Lambda, \Psi)$ with respect to ψ_l and λ_l , respectively. As a result, solving the

dual problem is equivalent to updating the dual variables for each link l by

$$\begin{aligned}\psi_l(k+1) &= \left[\psi_l(k) - \beta_\psi \left(c_l(\mathbf{P}) - \frac{\mu}{d_l^{(t)}} \right) \right]^+, \\ \lambda_l(k+1) &= \left[\lambda_l(k) - \beta_\lambda \left(c_l(\mathbf{P}) - \sum_{s:l \in L(s)} r_s - \frac{\mu}{2d_l^{(q)}} \right) \right]^+.\end{aligned}\quad (8)$$

In (8), $[x]^+$ is defined as $\max\{0, x\}$ and β_ψ and β_λ are the fixed step sizes. The communication overhead for different realizations involves exchanging the updated primal (r_s , P_l , $d_l^{(q)}$, and $d_l^{(t)}$) and dual (ψ_l and λ_l) variables.

3. Performance Evaluation Model

To compare convergence performance of distributed and centralized implementations, we develop performance evaluation model for processing and communication overheads. The per iteration processing overhead is the time required to update the primal and dual variables, and the per iteration communication overhead corresponds to the time spent in obtaining the required updated primal and dual variables.

3.1. Processing Overhead Model. Processing overhead comprises of gradient and projection evaluations denoted by $t_\nabla(\cdot)$ and $t_\Pi(\cdot)$, respectively. For instance, per iteration processing overhead for delay allocation subproblem, for centralized $t_{d_{\text{proc}}}^{(\text{cent.})}$ and distributed $t_{d_{\text{proc}}}^{(\text{dist.})}$ implementations, is given by

$$\begin{aligned}t_{d_{\text{proc}}}^{(\text{cent.})} &= \sum_{s \in S} \sum_{l \in L(s)} t_\nabla(d_l^{(t)}, d_l^{(q)}) + \sum_{s \in S} t_\Pi(L(s)), \\ t_{d_{\text{proc}}}^{(\text{dist.})} &= \sum_{s \in S} \sum_{l \in L(s)} (t_\nabla(d_l^{(t)}, d_l^{(q)}) + t_\Pi(L(s))).\end{aligned}\quad (9)$$

The higher processing overhead in (9) for distributed case compared to centralized is attributed to the projection evaluation at each node along the path in contrast to centralized implementation where it is evaluated once. Expressions similar to (9) can be obtained for rate, power, and dual subproblems with the exception that there is no associated projection overhead due to the absence of any linear constraints. The per iteration processing overhead for distributed case is then given by $t_{p_{\text{proc}}}^{(\text{dist.})} = t_{d_{\text{proc}}}^{(\text{dist.})} + t_{r_{\text{proc}}}^{(\text{dist.})} + t_{\lambda_{\text{proc}}}^{(\text{dist.})} + t_{\psi_{\text{proc}}}^{(\text{dist.})}$ and a similar expression can be obtained for the centralized case.

3.2. Communication Overhead Model. The communication overhead is a function of the number of node pairs involved in information exchange, the associated number of hops, and link packet success rate ρ_l . Let $h_l(N_j)$ be the number of hops from the transmitter of link l to node N_j . For the distributed case, communication overhead is the sum of

subproblem overheads. For instance, communication overhead for rate subproblem is given by $\sum_{s \in S} \sum_{l \in L(s)} h_l(N_s)/\rho_l$, where the summation over path $L(s)$ is to receive all dual variables to evaluate r_s for a given s . Now the per iteration communication overhead is obtained as

$$\begin{aligned}t_{\text{comm.}}^{(\text{dist.})}(s) &= \sum_{s \in S} \sum_{l \in L(s)} \frac{h_l(N_s)}{\rho_l} \\ &+ \sum_{k \in L} \sum_{l \in L(s)} \frac{h_l(N_k)}{\rho_l} + \sum_{k \in L} \sum_{l \in L(\theta(k))} \frac{h_l(N_k)}{\rho_l},\end{aligned}\quad (10)$$

where N_k is the receiver of link k and the second and the third terms on right-hand side correspond to the communication overheads for link delay and power allocation subproblems, respectively. There is no communication overhead associated with dual subproblem as all the required variables are available either locally or due to the evaluation of other subproblems at that node. For centralized implementation, the communication overhead is independent of the number of iterations and corresponds to transmitting the optimal variables to the required nodes once the algorithm has converged.

4. Experimental Results

For convergence performance comparison, we use an example network shown in Figure 1(a), where the dashed lines represent the transmission sessions between source and destination nodes. The centralized implementation is realized at node N_1 ; while for distributed implementation, we consider two cases. In the first case of distributed implementation (Dist₁), the assignment is done based on the subproblem functionality where the rate and delay optimization subproblems are assigned to nodes N_1 and N_5 , respectively, and the link power and dual variable updates are assigned to N_6 . In the second case (Dist₂), the assignment is done based on the variable localization by assigning the rate (r_s) updates to the source nodes (e.g., rate update for session s_4 is assigned to node N_4) and the link delay, power, and dual variable updates to the transmitting node on that link (e.g., P_{l_3} , $d_{l_3}^{(q)}$, $d_{l_3}^{(t)}$, ψ_{l_3} , λ_{l_3} are updated at the transmitting node N_2).

To validate the processing and communication overhead models, we obtain $t_\nabla(\cdot)$, $t_\Pi(\cdot)$ and packet success rate ρ_l using experiments. Overhead model is then validated by comparing its convergence performance with that of implementation results and is shown in Figure 1(b). For the centralized case, the model matches well with implementation results, while for the two cases of distributed implementation, minor difference is due to the dominating communication overhead for which accurate evaluation of parameter ρ_l is challenging. The results in Figure 1(b) also show that distributed implementation leads to higher convergence time compared to the centralized counterpart. It can also be noticed that the convergence performance of Dist₁ implementation is superior compared to that of Dist₂ implementation due to its dominant communication overhead.

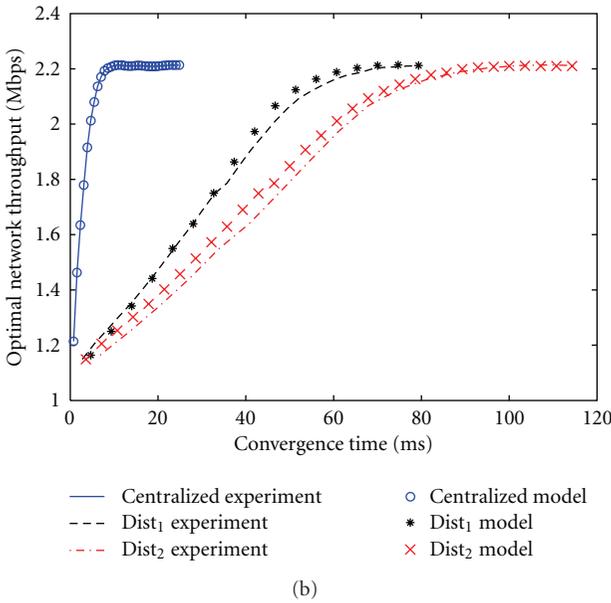
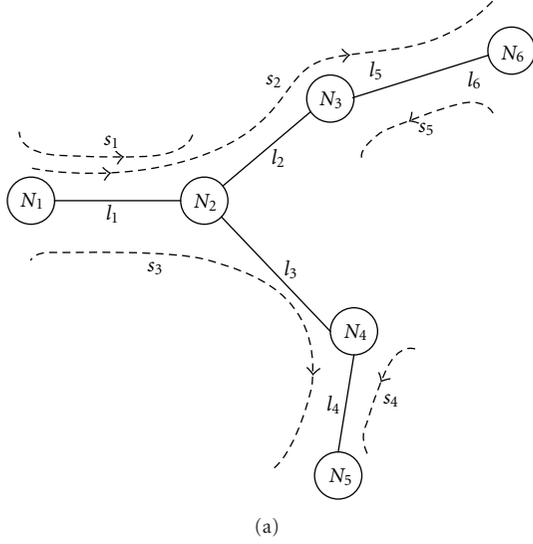


FIGURE 1: (a) Example network with each node equipped with TI's DSP (TMS320C6713) running at 225 MHz and MicroLinear's ML2722 RF transceiver; (b) Convergence performance comparison of overhead model with experimental realization for centralized and distributed implementations.

Higher communication overhead for the two cases of distributed implementation compared to its centralized counterpart is attributed to the fact that updated variables are exchanged at each iteration among the nodes executing different subproblems in case of distributed realization. Table 1 provides the distribution of the processing time and the communication overhead for centralized and distributed cases. A lower processing time of 27.4 ms for the centralized case, when realized at node N_1 , compared to the sum of processing times for distributed case, totaling 51.9 ms, is mainly due to the projection evaluation for end-to-end delay at each node along the shortest path for distributed case, compared to single evaluation at the central node. It should

TABLE 1: Overall convergence time components.

Algorithm component	Communication time (ms)	Processing time (ms)	Convergence time (ms)
N_1		13.1	
N_2		11.8	
Dist ₁ N_3	46.8	6.3	98.7
N_4		10.9	
N_6		9.8	
N_1 (Rate)		6.5	
Dist ₂ N_5 (Delay)	41.7	10.2	72.5
N_6 (Dual + Power)		14.1	
Centralized (N_1)	6.2	27.4	33.6

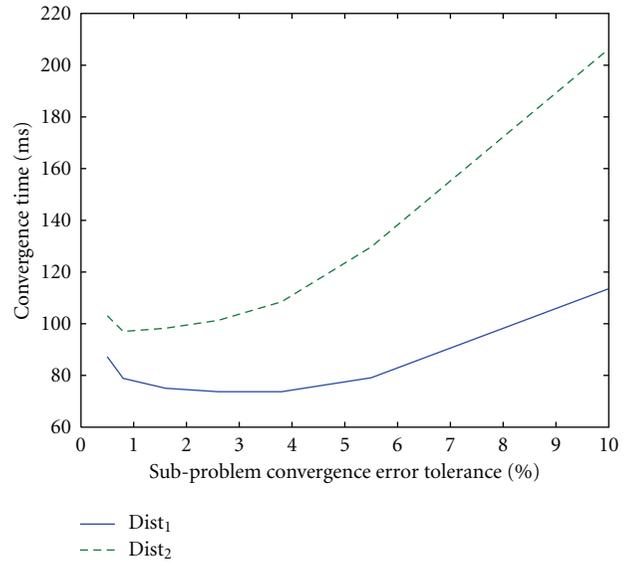


FIGURE 2: Convergence time as a function of percentage tolerance for the termination of iterative subproblem algorithm.

be noted that processing at different nodes is performed sequentially to avoid any variable update synchronization issues.

Next, for distributed implementation, we analyze how the termination criterion for the convergence of the subproblems affects the overall network resource optimization convergence. Figure 2 shows the overall convergence time as a function of subproblem convergence error-tolerance corresponding to two cases of distributed implementation. From the result in Figure 2, we observe that there is an optimal value of subproblem convergence error-tolerance which leads to minimum overall convergence time. When using a larger error-tolerance for subproblem convergence, it takes a more number of iterations of the algorithm to achieve overall convergence resulting in larger communication overhead per iteration of the algorithm. On the other hand, using smaller error-tolerance, for each subproblem

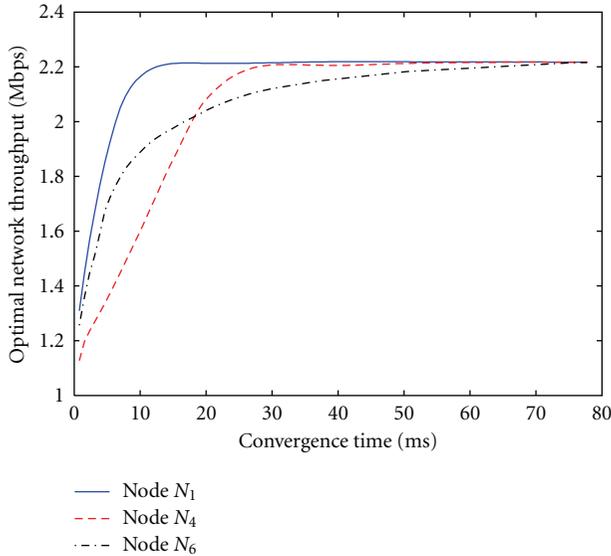


FIGURE 3: Convergence performance comparison for centralized implementation at the network nodes N_1 , N_4 , and N_6 .

convergence, leads to higher computation overhead per iteration of the algorithm. The optimal subproblem convergence error-tolerance effectively achieves a tradeoff between communication overhead and the associated computation time for distributed implementation. The processing and communication times in Table 1 are obtained for the optimal termination criterion shown in Figure 2.

Finally, we compare the convergence performance of centralized implementation for its realization at nodes N_1 , N_4 , and N_6 using performance evaluation model which is shown in Figure 3. It is observed that the choice of the node for centralized implementation as well as the termination criterion of the algorithm affect the convergence performance considerably. As a result, an improved convergence performance is achievable by carefully selecting the node for centralized implementation of the algorithm.

5. Conclusion

An iterative algorithm is implemented for the case of distributed and centralized realizations to solve the network utility maximization problem for wireless control networks. It is observed that the convergence performance for centralized case is better than that of distributed implementation due to the dominance of communication delay over processing delay. For distributed implementation, we observe a tradeoff between processing delay and the associated communication overhead for different subproblems, which provides an optimal overall convergence performance. A further performance improvement, for centralized case, can be achieved by using faster algorithms and higher processing power at the central node. However, for the case where processing delay dominates the communication overhead (e.g., fiber-optic link with a relatively slower centralized processing capability), we expect the distributed implementation to perform better compared to the centralized implementation.

Endnotes

1. We define WCN as a wireless network that supports exchange of feedback and control command information for a distributed control system.

References

- [1] F. P. Kelly, A. K. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.
- [2] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, 2003.
- [3] S. Ye and R. S. Blum, "Optimized signaling for MIMO interference systems with feedback," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2839–2848, 2003.
- [4] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, 2005.
- [5] M. Tahir and S. K. Mazumder, "Delay constrained optimal resource utilization of wireless networks for distributed control systems," *IEEE Communications Letters*, vol. 12, no. 4, pp. 289–291, 2008.
- [6] MicroLinear, "ML2722 900 MHz Low-IF 1.5 Mbps FSK Transceiver," http://www.rfmd.com/CS/Documents/ML2722_ML2722SPACEDatasheet.pdf.
- [7] R. Haftka and Z. Gürdal, *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

