

**RECEIVER BASED FLOW CONTROL WITH BACKPRESSURE ROUTING**A. Davincy Merline Sharmya¹, F.Ida²*Computer Science and Engineering, G.K.M College of Engineering and Technology*

Abstract - An optimization approach to achieve maximum utility in a receiver based flow control environment is proposed with extendibility to TCP flows. The mechanism used is backpressure routing which depends mainly on the queue backlog differences when a flow is generated between a source and the receiver. The queues of any feasible traffic can be stabilized through backpressure routing and hence provides higher utilization of network and its resources with increased fairness. In this paper we try to increase the network utility in a TCP environment by applying the backpressure routing mechanism for TCP flows. The flow control scheme is based on the receiver queue length and eliminates the greedy source based control on flows. A receiver based flow control set up is created based on the queue backlogs and it is applied to the conventional TCP flow control mechanism.

Keywords - Flow Control, Backpressure Routing, Queue Length, Finite Buffer Networks, Network Utility

I. INTRODUCTION

Flow control is the management of data flow between computers or devices or between nodes in a network so that the data can be handled at an efficient pace. Too much data arriving before a device can handle it causes data overflow, meaning the data is either lost or must be retransmitted.

Flow control can be achieved either at the sender side or at the receiver. Most studies in network flow control focus on source-based algorithms that require all sources to react properly to congestion signals such as packet loss or delay. However, in the presence of a greedy or malicious source that injects excessive traffic into the network, the throughput of other data flows may be adversely affected or even starved. Generally, a need for flow control arises whenever there is a constraint on the communication rate between two points due to limited capacity of the communication lines or the processing hardware. In such scenarios, source-based flow control may be ineffective. In Receiver based flow control the receiver remains protected and the throughput of all traffic classes is reoptimized without any changes in network operations. In receiver based flow control a network control policy consisting of a set of flow controllers at the receiver is designed so that the flows of the network can be handled efficiently. In contrast, under source-based flow control that relies on end-users or bordering nodes to regulate traffic, if a user becomes greedy or a bordering node is misconfigured, then all downstream nodes are overloaded. The receiver-based policy can be implemented in the whole network or implemented only at nodes. The throughput of the flows and network utility can be obviously controlled and improved in the receiver based flow control rather in source based flow control or reactive flow control where the throughput is mitigated. The relationship between duality theory, utility maximization and the classical network congestion is explored in [1]-[4].

We consider the problem of maximizing throughput utilities in a network, assuming that all traffic flows do not employ flow control and may overload the network. Flows are categorized into classes so that flows in a class have a shared destination. A class may simply be a flow specified by a source-destination pair, or corresponds to a subset of flows that communicate with a common Web site. A utility function is assigned to each traffic class, and the sum of the class based utilities is maximized as a means to control the aggregate throughput of flows in each class. The use of class-based utility functions is partly motivated by the need of mitigating network congestion caused by a collection of data flows whose aggregate throughput needs to be controlled. Without flow control at the sources, some packets will be dropped when the network is overloaded. To provide differentiated services to multiple traffic classes, we consider the scenario where the destinations can perform flow control to regulate the received throughput of each traffic class. The backpressure routing and scheduling paradigm in wireless networks have nodes route and schedule packets based on queue backlog differences, one can stabilize the queues for any feasible traffic. This seminal idea has generated a lot of research interest. Moreover, it has been shown that backpressure can be combined with flow control to provide utility optimal operation in [5]-[10]. One important practical problem that remains open, and is the focus of this paper, is the performance of backpressure with Transmission Control Protocol (TCP) flows. Therefore, it is crucial to exploit throughput improvement potential of backpressure routing and scheduling for TCP flows.

The strengths of these techniques have recently increased the interest in practical implementation of the backpressure framework over wireless networks like TCP. One reason for this recent interest is that a simple quantitative characterization of TCP throughput under given operating conditions offers the possibility of defining a fair share or TCP friendly throughput

for a non TCP flow that interacts with a TCP connection. TCP uses a sliding window flow control protocol.

In each TCP segment, the receiver specifies in the receive window field the amount of additionally received data that it is willing to buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgement and window update from the receiving host.

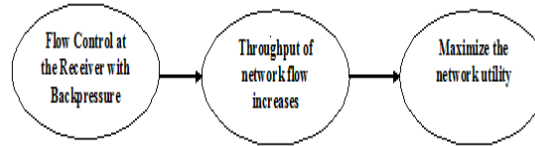


Fig.1. Relationship between Network Utility and Flow Control

However, TCP flows are not compatible with backpressure. Their joint behavior is so detrimental that some flows may never get a chance to transmit. The fundamental goal of TCP, which applies to all TCP variants, is to achieve as much bandwidth as possible while maintaining some level of long-term rate fairness across competing flows. In particular, TCP-aware backpressure takes into account the behavior of TCP flows, and gives transmission opportunity to flows with short queues and congestion window size adjustment. The interaction of TCP with backpressure in and is handled by updating the TCP congestion window evolution mechanism. In particular, if the queue size increases, the window size is reduced, otherwise, the window size is increased. Multipath TCP scheme is implemented over wireless mesh networks for routing and scheduling packets using a backpressure based heuristic, which avoids incompatibility with TCP [11]-[17]. In this paper, we propose “TCP-aware backpressure” that helps TCP and backpressure operate in harmony. In particular, TCP-aware backpressure takes into account the behavior of TCP flows, and gives transmission opportunity to flows with short queues.

The structure of the rest of the paper is as follows. Section II describes the system model followed by Section III that explains the backpressure routing in receiver based flow control. Section IV describes the TCP aware backpressure. Section V explains the implementation and the expected outcome of the proposed system. Section VI lists the related work and Section VII concludes the paper.

II. SYSTEM MODEL

We consider a general network model where flows may originate from a source in the Internet and traverse multiple hops to reach their destination in a wireless network. An end-to-end TCP connection is set up for each flow. Our goal in this paper is to develop TCP-aware backpressure routing and scheduling algorithms that operate in the wireless network. In this direction, we first develop our algorithms using the Lyapunov optimization framework by taking into account the incompatibility of TCP and classical backpressure. In this section, we provide an overview of the system model and assumptions that we use to develop the TCP-aware backpressure.

Wireless Network Setup: The wireless network consists of N nodes and L links, where N is the set of nodes and L is the set of links in the network. In this setup, each wireless node is able to perform routing and scheduling. Let S be the set of unicast flows between source-destination pairs in the network. We consider in our formulation and analysis that time is slotted, and t refers to the beginning of slot t . In every slot, packets randomly arrive at the network and are categorized into a collection C of classes. The definition of a data class is flexible except that we assume packets in a class $c \in C$ have the same destination σ_c .

Let $Q_n^{(c)}(t)$ be the queue backlog of class c packets at node n at time t ; assume $Q_n^{(c)}(0) = 0$ initially for all c and n .

Let $W_1^{(c)}(t)$ be the Backlog difference between two queues at time t . We seek to design a control policy that solves the global utility maximization problem

$$\text{maximize } \sum g_c(r_c).$$

III. RECEIVER-BASED FLOW CONTROL : BACKPRESSURE ROUTING

This section describes the novel feature of the receiver based backpressure/push-back mechanism that regulates data flows at the granularity of traffic classes, where packets can be classified based on their types.

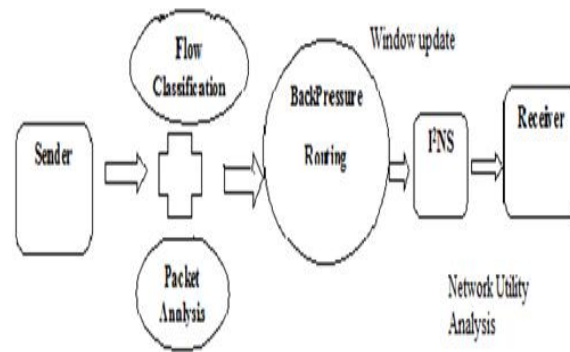


Fig.2. Interaction between sender and receiver with backpressure routing

Flow Classification

The use of class-based utility functions is partly motivated by the need of mitigating network congestion caused by a collection of data flows whose aggregate throughput needs to be controlled. Without flow control at the sources, some packets will be dropped when the network is overloaded.

The question we seek to answer is how to design in-network packet dropping and receiver-based flow control strategies to maximize the sum of class-based utilities and stabilize the network. For example consider a tree network with three classes of traffic. When the network is overloaded by exogenous traffic, nodes form a defense perimeter and perform rate throttling by dropping packets to optimize network performance.

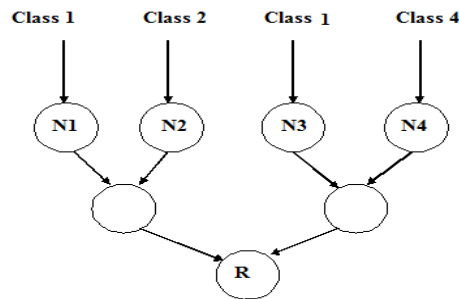


Fig.3. Tree network with three classes of traffic. When the network is overloaded by exogenous traffic, nodes(N1,N2,N3,N4) form a defense perimeter based on receiver R to optimize network performance.

Packet Analysis

In-network packet dropping and receiver-based flow control enhance the robustness of network operations. Consider the tree network in Fig. 3 that serves three classes of traffic. When the network is overloaded, an optimal packet-dropping policy implemented at all network nodes guarantees that the receiver is protected from excessive traffic, and the throughput of all traffic classes is optimized via receiver-end flow control. Suppose that node in Fig.3 is misconfigured and forwards everything it receives; effectively, node becomes a greedy user.

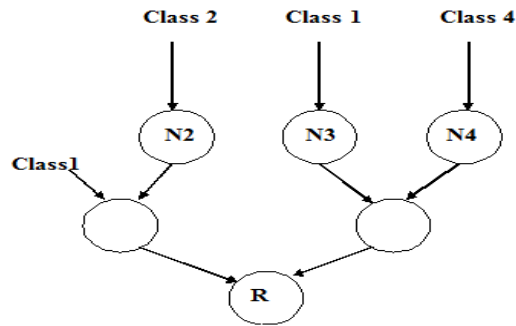


Fig.4 Resulting network topology when node N1 in Fig.3 is misconfigured so that it forwards everything it receives.

In this scenario since all nodes perform packet dropping, we have a new controlled network domain in Fig.4 in which the receiver remains protected and the throughput of all traffic classes is reoptimized without any changes in network operations. In contrast, under source-based flow control that relies on end-users or bordering nodes to regulate traffic, if a user becomes greedy or a bordering node is misconfigured, then all downstream nodes are overloaded. The control policy developed in this paper performs packet analysis at all intermediate nodes and seamlessly establishes a new defense perimeter without any changes in network operations.

Backpressure Routing

The flow control scheme creates virtual queues at the receivers as a push-back mechanism to optimize the amount of data delivered to the destinations via back-pressure routing.

The receiver-based flow controllers adjust throughput by modifying the differential backlogs between the receivers and their neighboring nodes—a small (or negative) differential backlog is regarded as a push-back mechanism to slow down data delivery to the receivers. To deal with undeliverable data due to network overload, we design a threshold-based packet-dropping mechanism that discards data whenever queues grow beyond certain thresholds.

IV. TCP-AWARE BACKPRESSURE ROUTING - INTERACTION WITH TCP

In this section, we design and analyze the TCP-aware backpressure scheme. In particular, we provide a stochastic control strategy including routing and scheduling to address the incompatibility between TCP and classical backpressure.

Window Size Update

It is crucial to exploit throughput improvement potential of backpressure routing and scheduling for TCP flows. However, TCP flows are not compatible with backpressure. Their joint behavior is so detrimental that some flows may never get a chance to transmit. The packets are stored in per-flow queues whereas in TCP it is based on the window size. The backpressure scheduling algorithm, also known as max-weight scheduling, determines the queue, hence the flow from which packets should be transmitted at time t . The arrival rates are independent from the scheduling decisions. The upper bound is called the window size or, simply, the window. The transmitter and receiver can be, for example, two nodes of the communication subnet, a user's machine and the entry node of the communication subnet, or the users' machines at the opposite ends of a session. Finally, the data units in a window can be messages, packets, or bytes.

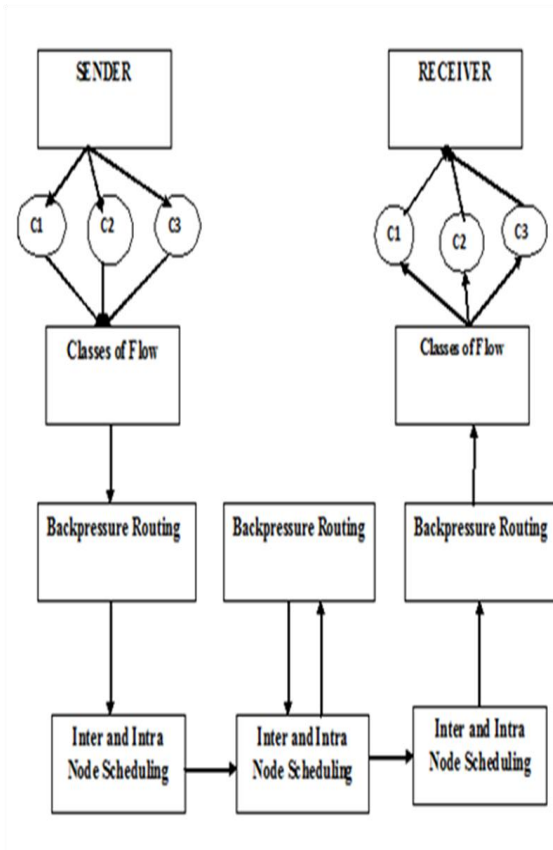


Fig.5 Interaction between the nodes with Backpressure Routing and Inter and Intra node Scheduling

Window update occurs based on the size. Window size increases for every acknowledgement is received in TCP flows. Hence the shorter windows TCP flow remains stagnant and not given a chance to transmit. The shorter windows are given the chances to increase in TCP aware backpressure.

I^2NS

The routing & intra node scheduling part of TCP-aware backpressure determines a flow from which packets should be transmitted at slot t from node as well as the next hop node to which packets from flow should be forwarded.. The inter-node scheduling as also called resource allocation part of TCP aware backpressure determines link transmission rates considering the link state information and interference constraints. The algorithm requires each node to know the queue size of their neighbors. To achieve this, each node transmits a message containing the size of its per-flow queue sizes at time t .

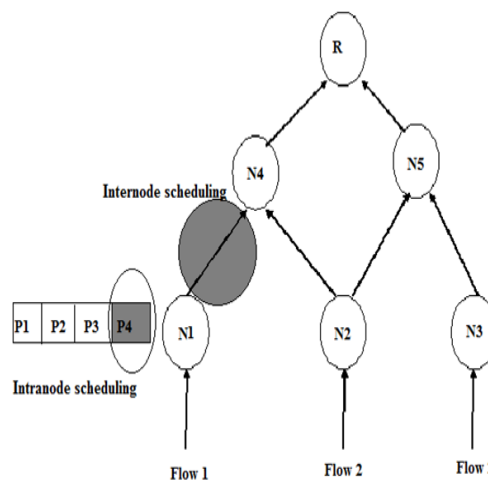


Fig.6 Inter and Intra Node Scheduling between the nodes in TCP aware Backpressure Routing

Network Utility Analysis

Network utility maximization problems have been studied to optimize network performance through a combination of flow control, routing, and scheduling, whose optimal operations are revealed as the solution to the utility maximization problems. Utility optimal policies that combine receiver-end flow control with back-pressure routing have been implemented. These policies optimize per-flow utilities and require infinite-capacity buffers. The utility maximization problem is formulated and the set of achievable throughput vectors in terms of queue overflow rates are characterized. A flow control increase indicates that the throughput of the flow increases. This in turn gives an entire throughput of all the flows in the network to increase. Hence the network utility increases.

A design control policy that solves the global utility maximization problem is to maximize the throughput of the network flow. A distributed control policy that allows us to transform the network utility maximization problem into an optimal queue control problem is developed similar to the primal-dual method that solves the optimization problems in stochastic networks.

V. IMPLEMENTATION AND PERFORMANCE EVALUATION

This section explains the basic steps in implementing the backpressure routing to achieve network utility and the expected outcome of the implementation.

UORA-Utility Optimal Overload Resilient Algorithm

Parameter Selection

Choose a parameter to increase the value of the utility function for the flow. Initial queue length is assumed to be zero such that no packets are sent and received by the nodes. The utility function is represented as g_c for each flow where 'c' represents the class the flow belongs to. Let g_c be a small constant that affects the performance of the policy, the parameter to satisfy solution to the utility maximization for a given exogenous arrival rate vector. This choice of utility function ensures that the node queues can be stabilized when its arrival rate is the optimal throughput. One feasible choice of g_c is the sum of capacities of all links connected to the receivers plus the sender nodes. The parameter is used to bound the queues away from zero and center them around. The parameters are chosen to satisfy for all the classes of flow. The value of g_c ensures that the service rate of the queue is zero whenever the network utility is optimal. This enforces the second condition to equalize the arrival rate and the service rate of the queue.

Packet Analysis

Based on the parameter chosen for the utility function the packets are analyzed. The packet analysis includes the difference between the utility function and the parameter chosen which is a constant. The packet-analysis subroutine in this policy is threshold based. The choice of g_c in the UORA is a back-pressure operation between the two queues. The bang-bang choice of g_c results from the aforementioned optimal queue control problem that has two conflicting goals: Stabilizing the queue needs large service allocations, but minimizing the weighted average service rate requires small values of g_c . It is notable that the policy needs only local information exchange between neighboring nodes and does not require the knowledge of exogenous arrival rates. Thus, network overload is autonomously resolved by each node making local decisions of routing, scheduling, and packet analysis.

Backpressure routing

The traffic in the network can be elastic or inelastic. If the traffic is inelastic, i.e., the flows' rates are fixed and within the capacity region, then the goal is to route/schedule the traffic through the network while ensuring that the queues in the network are stable. If the traffic is elastic, then the goal is to allocate the network's resources to all flows in some fair manner. The traditional back-pressure algorithm is throughput optimal. Furthermore, for elastic traffic, the authors in [11] have shown that this algorithm, jointly with the congestion control algorithm, can solve the optimal resource allocation problem. The traditional back-pressure algorithm requires per-flow per destination queues. For a link (m,n) the backlog difference is calculated for each link. The class of each link denoted as C_l is specified for finding the queue backlog difference between the receiver and the sender.

$$W_l^{(c)}(t) = Q_n^{(c)}(t) - Q_m^{(c)}(t)$$

Receiver Based Flow Control

At the destination, choose the queue rate to control the flow at the receiver end such that the network utility is increased. The utility maximization problem needs each receiver to maximize the new utility function subject to receiver, where the g_c is an auxiliary control variable and is the throughput of class packets.

$$\text{maximize } \sum g_c(r_c)$$

Queue Update

Update the queue $Q_n^{(c)}(t)$ i.e the queue length between two queues at time t after each slot.

Algorithm For Backpressure Routing

The traditional back-pressure algorithm uses either per-flow or per-destination queues in the fixed-routing scenario. However, for the sake of simplicity, we will consider the per-flow implementation. In the per flow implementation, each node maintains a separate queue for each flow going through it.

The queue maintained at node n for flow f is for buffering packets of f which reach n . Let $Q_n^{(c)}(t)$ denote the length of that queue at the beginning of time slot t .

Finding the Backlog difference of the queues

At time slot t , For each link (l,m) the maximum differential backlog of all flows going through that link is determined as

$$W_l^{(c)}(t) = Q_n^{(c)}(t) - Q_m^{(c)}(t)$$

Backpressure Scheduling

Determine the link with maximum backlog difference in the network among all the links considered. The maximum backlog difference is denoted as π^* .

At time slot t ,

$$\pi^*(t) = \max(W_l^{(c)}(t))$$

Expected Outcome

The expected outcome of the proposed system with three classes of traffic and any simple topology tree or diamond is to achieve a utility optimal network. The proposed system uses the backpressure routing and achieves the maximum network usage in the TCP environment. The TCP environment is incompatible to the backpressure algorithm. To overcome the complexity the intra and inter node scheduling is applied. The proposed technique can increase the utility of the network up to 0.39Mbps.

The parameter considered is the queue backlog difference of the nodes in the network. The aim is to achieve a concave utility function that increases from the minimum of 0.07Mbps increase(achieved) to a achievable maximum point of 0.39Mbps increase(expected).

VLRELATED WORK

Backpressure, a routing and scheduling framework over communication networks [1], [2] has generated a lot of research interest [7], mainly in wireless ad-hoc networks. It has also been shown that backpressure can be combined with flow control to provide utility-optimal operation guarantee [3], [20]. The strengths of backpressure have recently increased the interest on practical implementation of backpressure over wireless networks. Backpressure has been implemented over sensor networks [21] and wireless multi-hop networks [22]. The multi-receiver diversity has been explored in wireless networks using backpressure in [23]. The 802.11 compliant version of enhanced backpressure is evaluated in [24]. Backpressure routing and rate control for intermittently connected networks was developed in [24].

VILCONCLUSION

A receiver-based flow control to cope with network overload and achieve optimal utility is developed. The proposed scheme is robust to uncooperative users who do not employ source-end flow control and to malicious users that intentionally overload the network. A novel feature of the policy is a receiver-based backpressure/push-back mechanism that regulates data flows at the granularity of traffic classes, where packets can be classified based on their types. This is in contrast to source-based schemes that can only differentiate between source-destination pairs. The developed control policy may be useful to handle different types of service requests in Internet application servers under overload conditions, or manage multicommodity flows in finite-buffer networks with performance guarantees. The receiver based flow control scheme has a wide range of potential applications, including preventing denial-of-service. The future work is a closely related problem that involves the interaction between TCP-based flow control and the receiver-based flow control scheme that is TCP's response to the receiver flow control and optimal network utility. The future work is backpressure-based implementations with queue prioritization and congestion window size adjustment. The interaction of TCP with backpressure is handled by updating the TCP congestion window evolution mechanism in the extension of receiver based flow control.

REFERENCES

- [1] L. Tassiulas, A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *IEEE Trans. on Auto. Control*, vol. 37(12), Dec. 1992.
- [2] L. Tassiulas, A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," in *IEEE ToIT*, vol. 39(2), March 1993.
- [3] M. J. Neely, E. Modiano, C. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *IEEE/ACM ToN*, vol. 16(2), April 2008.
- [4] K. Tan, J. Song, Q. Zhang, M. Sridharan, "A compound TCP approach for high-speed and long distance networks," in *Proc. of IEEE INFOCOM*, Barcelona, Spain, April 2006.
- [5] S. Ha, I. Rhee, L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," in *SIGOPS Oper. Syst. Rev.*, vol. 42(5), July 2008.
- [6] The Network Simulator - ns-2, Version 2.35, available at www.isi.edu/nsnam/ns/.
- [7] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," Morgan & Claypool, 2010.
- [8] M. Chiang, S. T. Low, A. R. Calderbank, J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," in *Proceedings of the IEEE*, vol. 95(1), Jan. 2007.
- [9] X. Lin, N. B. Schroff, R. Srikant, "A tutorial on cross-layer optimization in wireless networks," in *IEEE JSAC*, vol. 24(8), Aug. 2006. [10] H. Seferoglu, E. Modiano, "Diff-Max: separation of routing and scheduling in backpressure-based wireless Networks," in *Proc. Of IEEE INFOCOM*, Turin, Italy, April, 2013.
- [11] L. X. Bui, R. Srikant, A. Stolyar, "A novel architecture for reduction of delay and queueing structure complexity in the back-pressure algorithm," in *IEEE/ACM Transactions on Networking*, vol. 19(6), Dec. 2011.
- [12] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, J. Barros, "Network coding meets TCP," in *Proc. of IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
- [13] S. Gheorghiu, A. L. Toledo, P. Rodriguez, "Multi-path TCP with network coding for wireless mesh networks," in *Proc. of IEEE ICC*, Cape Town, South Africa, May 2010.
- [14] H. Seferoglu, A. Markopoulou, K. K. Ramakrishnan, "I2NC: intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. of IEEE INFOCOM*, Shanghai, China, April 2011.
- [15] P. A. Chou, Y. Wu, "Network coding for the Internet and wireless networks," in *IEEE Signal Proc. Magazine*, vol. 24(5), Sept. 2007.
- [16] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proc. of ACM SIGCOMM*, Vancouver, Canada, Sep. 1998.
- [17] S. Low, "A duality model of TCP and queue management algorithms," in *IEEE/ACM Transactions on Networking*, vol. 11(4), Aug. 2003.
- [18] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc on-demand distance vector (AODV) routing," *RFC 3561, IETF*, July 2003.
- [19] R. K. Jain, "The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling," John Wiley & Sons, April 1991.
- [20] A. L. Stolyar, "Greedy primal dual algorithm for dynamic resource allocation in complex networks," in *Queueing Systems*, vol. 54, 2006.
- [21] S. Moeller, A. Sridharan, B. Krishnamachari, O. Gnawali, "Routing without routes: the backpressure collection protocol," in *Proc. of ACM IPSN*, Stockholm, Sweden, April 2010.
- [22] R. Lauffer, T. Salonidis, H. Lundgren, P. L. Guyader, "XPRESS: a crosslayer backpressure architecture for wireless multi-hop networks," in *Proc. of ACM MobiCom*, Las Vegas, NV, Sep. 2011.
- [23] A. A. Bhorkar, T. Javidi, A. C. Snoereny, "Achieving congestion diversity in wireless ad-hoc networks," in *Proc. of IEEE INFOCOM*, Shanghai, China, April 2011.
- [24] K. Choumas, T. Korakis, I. Koutsopoulos, L. Tassiulas, "Implementation and end-to-end throughput evaluation of an IEEE 802.11 compliant version of the enhanced-backpressure algorithm," in *Proc. of TridentCom*, Thessaloniki, Greece, June 2012.
- [25] J. Ryu, V. Bhargava, N. Paine, S. Shakkottai, "Backpressure routing and rate control for ICNs," in *Proc. of ACM MobiCom*, Chicago, IL, Sep. 2010.
- [26] A. Warrier, S. Janakiraman, S. Ha, I. Rhee, "DiffQ: practical differential backlog congestion control for wireless networks," in *Proc. of IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.
- [27] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, A. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc networks," in *Proc. Of IEEE INFOCOM*, Phoenix, AZ, April 2008.
- [28] B. Radunovic, C. Gkantsidis, D. Gunawardena, P. Key, "Horizon:

balancing TCP over multiple paths in wireless mesh network,” in *Proc. of ACM MobiCom*, San Francisco, CA, Sep. 2008.

- [29] J. Ghaderi, T. Ji, R. Srikant, “Connection-level scheduling in wireless networks using only MAC-layer information,” in *Proc. of IEEE INFOCOM*, Orlando, FL, March 2012.
- [30] A. Shpiner, I. Keslassy, “Modeling the interactions of congestion control and switch scheduling,” in *Computer Networks*, vol. 55(6), April 2011.
- [31] P. Giaccone, E. Leonardi, F. Neri, “On the interaction between TCP-like sources and throughput-efficient scheduling policies,” in *Elsevier*, 2013.
- [32] Y. Yi, A. Proutière, and M. Chiang, “Complexity in wireless scheduling: impact and tradeoffs,” in *Proc. of ACM MobiHoc*, Hong Kong, China, May 2008.