International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

XPRESS for Wireless Multi hop Networks

[1] M.Damodaram & [2] B.Rupadevi
[1]M.Tech (Computer Science), AITS, Tirupati, damodaram0503@gmail.com
[2] Ph.D Associate Professor, Dept of CSE, AITS, Tirupati, singhrupa23@gmail.com

Abstract—

A holistic approach to the protocol architecture design in multihop wireless networks. Our goal is to integrate various protocol layers into a rigorous framework, by regarding them as distributed computations over the network to solve some optimization problem. Different layers carry out distributed computation on different subsets of the decision variables using local information to achieve individual optimality. Taken together, these local algorithms (with respect to different layers) achieve a global optimality. Our current theory integrates three functions—congestion control, routing and scheduling in transport, network and link layers into a coherent framework. Within this context, this model allows us to systematically derive the layering structure of the various mechanisms of different protocol layers, their interfaces, and the control information that must cross these interfaces to achieve a certain performance and robustness. Transmissions over the network are scheduled using a throughput-optimal backpressure algorithm. Realizing this theoretical concept entails several challenges, which we identify and address with a cross-layer design and implementation on top of our wireless hardware platform. In contrast to previous work, we implement and evaluate backpressure scheduling over a TDMA MAC protocol, as it was originally proposed in theory. Our experiments in an indoor test bed show that XPRESS can yield up to 128% throughput gains over 802.11.

I INTRODUCTION

The success of communication networks has largely been a result of adopting a layered architecture. With this architecture, its design and implementation is divided into simpler modules that are separately designed and implemented and then interconnected. A protocol stack typically has five layers, application, transport (TCP), network (IP), data link (in-clude MAC) and physical layer. Each layer controls a subset of the decision variables, hides the complexity of the layer below and provides well-defined services to the layer above. Together, they allocate networked resources to provide a reliable and usually best-effort communication service to a large pool of competing users. We also show that XPRESS accurately emulates the optimal backpressure schedule and delivers relatively low delays when operating close to capacity. Finally, we provide an analysis of the communication and computation overhead of XPRESS and identify different system design choices and limitations

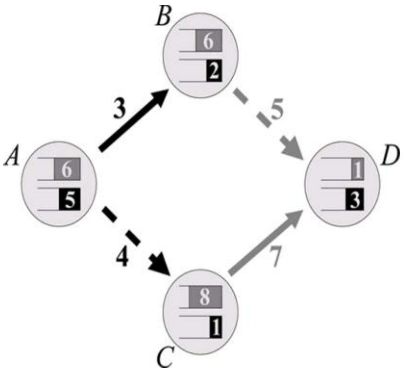

Fig. 1. Backpressure scheduling in a network with two flows, black and gray, from to . Links in sets {(A,B),(C,D)} (continuous) and {(A,C),(B,D)} (dashed) can be scheduled in the same slot.

II BACKPRESSURE SCHEDULING

The backpressure algorithm was introduced as a scheduling policy that maximizes the throughput of wireless multichip networks. Assuming slotted time, the basic idea of backpressure scheduling is to select the “best” set of no interfering links for transmission at each slot. We now describe this idea in a 4-node network with two flows, black and gray, from node A to D, depicted in Fig. 1. Each node maintains a separate queue for each flow. For each queue, the number of backlogged packets is shown. Assume that we have two link sets, {(A,B),(C,D)} and {(A,C),(B,D)}, shown as continuous and dashed lines, respectively. The links in each set do not interfere and can transmit in the same time slot. The scheduler executes the following three steps at each slot. First, for each link, it finds the flow with the maximum differential queue backlog. For example, for link (A,B) , the gray flow has a difference of 0 packets and the black flow has a difference of 3 packets. The maximum value is then assigned as the weight of the link (see Fig. 1). Second, the scheduler selects the set of no interfering links with the maximum sum of weights for transmission. This requires to compute the sum of link weights for each possible set. In the example, {(A,B)(C,D)} set sums to 3+7=10 and {(A,C)C(B,D)} set sums to 4+5=9 . The scheduler then selects the set with the maximum sum of weights, i.e.{(A,B),(C,D)} to transmit at this slot. Finally, packets from the selected flows are transmitted on the selected links, i.e., black flow on link and gray flow on link . The same computation is then performed at every slot.

.A. Backpressure Algorithm

More formally, the backpressure scheduling algorithm consists of the following steps executed for each time slot.

Flow Scheduling and Routing: For each link , select the flow with the maximum queue differential backlog where and are the queue backlogs for flow at nodes and , respectively, and is the set of flows. The maximization in implicitly performs routing by selecting the link that each flow may use during the slot. The weight of each link is then selected as the weight of flow.

Link Scheduling: Select the optimal link capacity vector that satisfies where are the link capacity vectors. The capacity of each link is the maximum rate in bits per second that the link can transmit subject to the channel state and the interference due to the other links in the vector. The set of all feasible link capacity vectors define the capacity region A.

Transmission: During the time slot, a selected link transmits a packet from flow using rate

B Congestion Control

The backpressure algorithm is throughput-optimal when the flow rates are within the capacity region. This issue can be addressed by combining the backpressure algorithm with the network utility maximization (NUM) framework, originally proposed for wire line networks [5]. This framework leads to a simple distributed congestion control algorithm where the source of each flow adjusts the flow rate as where is the queue backlog for flow at the source and is the inverse of the first derivative of the utility function at the point . In [6]–[8], it is proven that the congestion control scheme of (4) regulates the flow rates to be within the capacity region and cooperates with the backpressure scheduler to maximize throughput.

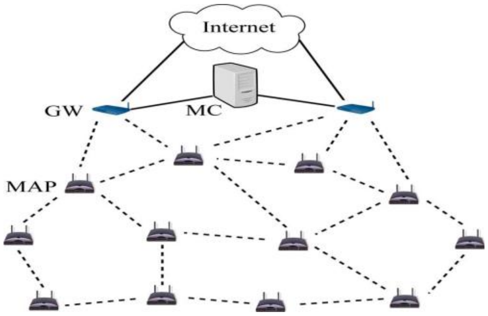

Fig. 2. XPRESS architecture, composed of MAPs to provide wireless coverage to mobile clients, GWs to provide Internet connectivity, and an MC for wireless scheduling.

III XPRESS DESIGN

This section presents the XPRESS system, a cross-layer backpressure architecture for wireless multihop networks. To our knowledge, XPRESS is the first system to implement backpressure scheduling over a time-slotted MAC, as it was originally proposed in theory. We first provide a high-level system overview, and then we detail the data plane and control plane designs. Finally, we describe the design of our backpressure scheduler with speculative scheduling.

A. Overview

I implement and evaluate backpressure scheduling over a TDMA MAC protocol, as it was originally proposed in theory. Our experiments in an indoor test bed show that XPRESS can yield up to 128% throughput gains over 802.11. ropose a passive technique to learn about transmission conflicts. This approach exploits several exposed terminals unused by 802.11, but it does not address hidden interferers. The propose an online approach where APs periodically silence their clients and run a quick interference tests. ropose centralized architectures to schedule AP transmissions. XPRESS shares the centralized philosophy of these architectures, but also has fundamental differences. propose 802.11 backpressure implementations to reduce the end-to-end delay. This issue can be addressed by combining the backpressure algorithm with the network utility maximization (NUM) framework, originally proposed for wire line network. In XPRESS, the wireless network is composed of several mesh access points (MAPs), a few gateways (GWs), and a mesh controller (MC), as depicted in Fig. 2. We use the term “node” to refer to a mesh node that can be either anMAP or a GW. The MAPs provide wireless connectivity to mobile clients and also operate as wireless routers, interconnecting with each other in a multihop fashion to forward user traffic. Mobile clients communicate with MAPs over a different channel, and thus are not required to run the XPRESS protocol stack. The GWs are connected to both the wireless network and the wired infrastructure and provide a bridge between the two. The MC is responsible for the coordination of the wireless transmissions in the network, and it is analogous to a switching control module. In our design, the MC is deployed in a dedicated node in the wired infrastructure and connects to the gateways through high-speed links. In an alternative design, the MC could be implemented within one of the gateways, if necessary.

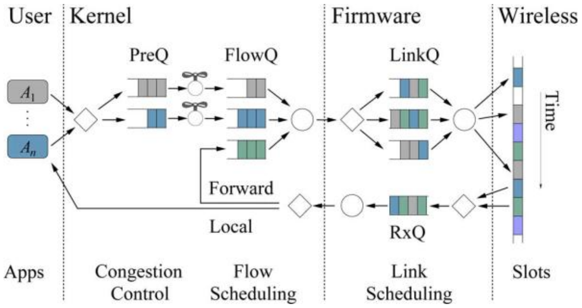

Fig. 3. Data plane at XPRESS nodes. Diamonds are packet classifiers, while circles are packet schedulers. Rate control and flow scheduling occur at the OS kernel, whereas link scheduling occurs at the network card firmware.

B. Data Plane

The XPRESS data plane spans the transport, network, and MAC layers of the protocol stack, as depicted in Fig. 3. The transport and network layers implement congestion control and flow scheduling, respectively. The MAC layer implements link scheduling and a TDMA MAC protocol. Fig.3 depicts this organization on our test bed hardware devices where the full MAC firmware resides on the wireless cards, while the upper layers reside in the host OS kernel. In the figure, diamonds represent packet classifiers, while circles represent packet schedulers. The data flow from left to right is outgoing packets originating from the applications to the wireless medium; the data flow in the opposite direction is incoming packets that are routed or delivered to the applications. Packets in the slotted wireless medium (far right), which are neither incoming nor outgoing, represent transmissions between two other nodes in the network.

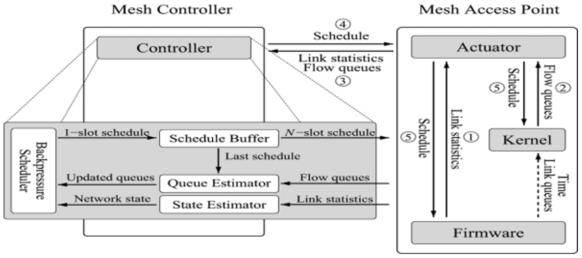

Fig. 4 depicts the XPRESS control plane with its different components at the MC and an MAP.

1.XPRESS Flows: In XPRESS, a flow is defined at the IP layer by its source and destination mesh nodes. Our design is general and can easily accommodate other flow definitions. However, compared to the usual 5-tuple flow definition of source and destination ,IP addresses, source and destination transport ports, and transport protocol (i.e., TCP or UDP), this design decision reduces processing and communication overhead in XPRESS at the expense of flow granularity.

2. Link Scheduling: The MAC protocol keeps an individual queue for each neighbor in order to enable *link scheduling*, which allows a higher spatial reuse than *node scheduling*. As packets dequeued from the FlowQ arrive at the link-level packet classifier, they are classified according to the destination MAC address and inserted into the appropriate link queues (LinkQ). The slottedMAC, realized by a TDMAMAC protocol maintains network-wide node synchronization and ensures that transmissions occur strictly within slot boundaries. When a transmission slot starts, the MAC protocol dequeues a packet from the scheduled Link Q and transmits it over the air. If the transmission fails and the retransmission limit is not reached, the packet remains in the appropriate Link Q until the next slot for the same neighbor.

3. Packet Reception and Forwarding: Once a packet is received, it is first filtered based on the destination MAC address and then inserted into a single receive queue (RxQ) at the firmware. The packet is delivered to the network layer at the kernel, where it is routed and tagged for local delivery or forwarding. In the latter case, the packet is inserted into the respective FlowQ and waits to be scheduled, just like a locally generated packet after passing congestion control.

C. Control Plane

The different system components must exchange control information to coordinate the network-wide transmissions. Fig. 4 depicts the XPRESS control plane with its different components at the MC and an MAP. The uplink control channel is represented by the arrows labeled from 1 to 3, while the downlink control channel is represented by arrows 4 and 5..

Uplink Channel: Each node runs an actuator application that communicates with the controller application at the MC. A schedule computation cycle begins at the start of a new frame. At this point, the TDMA MAC protocol notifies the actuator about the new frame and piggybacks in the same message the
, such as received signal strength and delivery ratio.
The actuator then collects the FlowQ lengths. combines all information, and sends it to the MC using the uplink channel.

Mesh Controller (MC): The zoom of the MC in Fig. 4 shows the different steps taken for the calculation of a TDMA frame schedule composed of slots. First, the link statistics are used to estimate the network state, namely, the interference relation sand link loss rates... The scheduler then uses the FlowQ lengths and network state to compute the network schedule, using the backpressure algorithm our speculative scheduling technique

Downlink Channel: When the schedule computation is finished, the MC disseminates the new schedule using the downlink control channel. The actuator receives this packet and forwards the new schedule both to the OS kernel as well as to the MAC layer. The TDMA MAC starts using this new schedule for data transmission in the next frame. Packets will then be dequeued from the FlowQs to the LinkQs in accordance with this new schedule.

IV. XPRESS IMPLEMENTATION

The XPRESS design is general and can be realized on a wide range of platforms. In this section, we describe the main components of our cross-layer implementation in the Linux OS and the firmware of our WiFi cards. We follow a top–down approach
and describe these components in the order of the outgoing data path in Fig. 3

***A Queues and Scheduler Flow Queues**:* Outgoing packets are intercepted using the Net filter post-routing hook in the Linux kernel. Intercepted incoming packets that have been routed, and thus are ready to be forwarded, are classified and put into the corresponding FlowQ. We pass the FlowQ backlog information to the actuator module through the Linux interface. The actuator in turn forwards this information over the uplink control channel to the MC for schedule computation

B Backpressure Scheduler

In our speculative scheduling, the schedule for each slot is computed at the MC using the backpressure algorithm.

Kernel Packet Scheduler: The packet scheduler, in turn, is implemented in the MAPs as a kernel thread that waits for the computed schedule. The schedule contains information about which flows to transmit in each slot, as well as the next hop to be used by each flow.

C. TDMA MAC Protocol

Implement a time-slotted MAC protocol in the firmware of our 802.11a/b/g cards and disable some inherent CSMA functionality, including carrier sensing, back off, RTS/CTS, and NAV. This gives us full control of transmission times and allow us to divide the MAC operation into frames and time slots. In our implementation, each TDMA frame is composed of a group of slots, which are divided into a control subframe (CS) followed by a data subframe (DS).. Each data slot is assigned to a particular set of non interfering links based on the schedule computed by the MC. The MAC packet scheduler receives this schedule from the actuator, and then takes care of transmitting packets according to the schedule and within slot boundaries.

V. INTERFERENCE ESTIMATION

Nodes in XPRESS measure the RSS of each neighbor from the synchronization beacons they transmit in the control subframe of each TDMA frame. Each node has an assigned slot in this subframe, at which the node transmits its beacon using the lowest 802.11a PHY rate (i.e., 6 Mb.s) without interference. This results in one RSS measurement per frame for each link. Based on these measurements, we estimate the signal-to-interference ratio (SIR) of a link under the interference of a transmitting node as the difference , between , the RSS of link , and , the RSS of link , both in dBm.

VI CONCLUSION

We presented the design and implementation of XPRESS, a backpressure architecture for wireless multihop networks. Our design leverages a centralized controller for obtaining throughput optimal scheduling. In contrast to previous work, we integrated backpressure scheduling with a TDMA MAC protocol to allow precise timing in transmissions. Moreover, we introduced a novel interference estimation technique and an efficient speculative backpressure scheduler. In our future work, apart from the topics discussed , we also intend to evaluate XPRESS in larger networks. We believe our work opens up interesting avenues in wireless network system design, showing that optimal centralized routing and scheduling are feasible for small- to medium-sized wireless multi hop networks.

REFERENCES

[1] L. Tassiulas and A. Ephremides, “Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[2] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, “Joint scheduling and congestion control in mobile ad-hoc networks,” in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 619–627.

[3] B. Radunovic, C. Gkantsidis, D. Gunawardena, and P. Key, “Horizon: Balancing TCP overmultiple paths in wirelessmesh network,” in *Proc. ACM MobiCom*, Sep. 2008, pp. 247–258.

[4] A. Warrier, S. Janakiraman, S. Ha, and I. Rhee, “DiffQ: Practical differential backlog congestion control for wireless networks,” in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 262–270.

[5] F. Kelly, A. Maulloo, andD. Tan, “Rate control in communication networks: Shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, pp. 237–252, 1998.

[6] L. Chen, S. Low, M. Chiang, and J. Doyle, “Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks,” in *Proc. IEEE INFOCOM*, Apr. 2006, pp. .

[7] A. Eryilmaz and R. Srikant, “Joint congestion control, routing, and MAC for stability and fairness in wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[8] X. Lin and N. B. Shroff, “Joint rate control and scheduling in multihop wireless networks,” in *Proc. IEEE CDC*, Dec. 2004, vol. 2, pp. 1484–1489.

[9] B. Radunovic and J.-Y. L. Boudec, “Rate performance objectives of multihop wireless networks,” *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 334–349, Oct.–Dec. 2004.

[10] K. Makino and T. Uno, “New algorithms for enumerating all maximal cliques,” in *Proc. 9th Scand. Workshop Algor. Theory*, Jul. 2004, pp. 260–272.

[11] D. Koutsonikolas, T. Salonidis, H. Lundgren, P. LeGuyadec, C. Hu, and I. Sheriff, “TDM MAC protocol design and implementation for wireless mesh networks,” in *Proc. ACM CoNEXT*, Dec. 2008, p. 28.

[12] J. Lee, J. Ryu, S. Lee, and T. Kwon, “Improved modeling of IEEE 802.11a PHY through fine-grained measurements,” *Comput. Netw.*, vol. 54, no. 4, pp. 641–657, Mar. 2009.

[13] “iperf,” 2011 [Online]. Available: http://sourceforge.net/projects/iperf

[14] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Found. Trends Netw.*, vol. 1, no. 1, pp. 1–144, 2006.

[15] P. Wang, “Throughput optimization of urban wireless mesh network,” Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Delaware, Newark, DE, USA, 2009.

[16] X. Lin and N. B. Shroff, “The impact of imperfect scheduling on crosslayer congestion control in wireless networks,” *IEEE Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[17] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, “FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks,” in *Proc. Allerton Conf.*, Sep. 2010, pp. 514–521.

[18] A. Aziz, D. Starobinski, P. Thiran, and A. ElFawal, “EZ-Flow: Removing turbulence in IEEE 802.11 wireless mesh networks without message passing,” in *Proc. ACM CoNEXT*, Dec. 2009, pp. 73–84.

[19] J. Ryu, V. Bhargava, N. Paine, and S. Shakkottai, “Back-pressure routing and rate control for ICNs,” in *Proc. ACM MobiCom*, Sep. 2010, pp. 365–376.

[20] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, “Routing without routes: The backpressure collection protocol,” in *Proc. IEEE/ACM IPSN*, Apr. 2010, pp. 279–290.