A New Approach to MAC Protocol Optimization

A. Faragó, A. D. Myers, V. R. Syrotiuk, and G. V. Záruba
Erik Jonsson School of Engineering and Computer Science
Center for Advanced Telecommunications Systems and Services (CATSS)
The University of Texas at Dallas
{farago, amyers, syrotiuk, zaruba}@utdallas.edu

Abstract—**A systematic and automatic method for dynamically optimizing medium access control (MAC) protocol parameters is presented. Our** *meta-protocol* **approach is capable of performing on-line optimization of critical MAC parameters without knowing in advance what network conditions will arise or how they may fluctuate over time. Furthermore, this dynamic optimization is achieved without any centralized control or exchange of control messages between nodes. The power of the new technique is demonstrated by two examples. In a LAN environment, it outperforms traditional contention based MAC protocols that adjust retransmission probabilities, e.g., employing binary exponential backoff. In a synchronous multi-hop environment, it automatically converges to the proper transmission schedule assignments for the actual node density.**

INTRODUCTION

A multiaccess medium is the foundation for communication in a number of networks, such as local area networks (LANs), metropolitan area networks (MANs), satellite networks, and radio networks. The medium in these networks is a shared resource, and a medium access control (MAC) protocol coordinates all packet transmissions. Consequently, the performance of the MAC protocol has an immediate and fundamental impact on the overall efficiency of a multiaccess network.

When the network conditions are unknown or changing, the choice of the MAC protocol to use is not obvious. The usual approach in this situation is to include some kind of adaptivity in order to adjust the protocol parameters to the actual network conditions. There are a number of well known adaptation techniques, a few typical examples are briefly reviewed below. While these techniques are capable of successfully adjusting the parameters of the specific protocol for which they are designed, none of them extends into a general method that can be applied in practically any situation. Our contribution in this paper is a principally new, systematic approach to achieve MAC protocol adaptivity at a *general* level, not restricted to a specific protocol.

One well known form of adaptation via parameter adjustment is the class of backoff algorithms used by contention protocols. Two characteristic backoff solutions are the pseudo-Bayesian algorithm and binary exponential backoff.

The pseudo-Bayesian algorithm dynamically manipulates the transmission probability of each node by maintaining an estimate of the number of backlogged nodes, i.e., nodes with packets to send [2], [9]. An increase in the number of backlogged stations reduces the transmission probability, and vice versa. Thus, the pseudo-Bayesian algorithm can adjust its operation to match the contention level in a multiaccess channel. Unfortunately, each node must have an accurate estimate of the overall arrival rate of incoming traffic which is generally unknown and time varying. Each node must also know the outcome of every network transmission which may not be possible in every type of multiaccess network. Nonetheless, the pseudo-Bayesian algorithm is used in a number of MAC protocols (see [6], [10], [11] for examples).

Unlike the pseudo-Bayesian algorithm, the binary exponential backoff algorithm does not require global feedback or an estimate of the network arrival rate. Instead, each node adjusts its transmission probability based on the number of unsuccessful transmission attempts. However, exponential backoff was proven unstable (infinitely growing delays) under quite general modeling assumptions [1]. Furthermore, a large class of acknowledgment based backoff mechanisms were also proven to be unstable [8]. Thus, finding the optimal dynamic adjustment of the transmission probability is far from trivial.

Another important class of protocols is the family of *spatial reuse* TDMA protocols used in mobile multi-hop networks. Here adaptivity can be achieved by dynamically recomputing transmission schedules based on the local network topology (see [4], [11] for examples). One way to adjust in this situation is that the nodes alternate between a contention protocol and a TDMA allocation protocol. Nodes use the contention protocol to determine the proper schedule length and slot assignments through the exchange of small control packets. Once the schedules are fixed, the operation switches over to the TDMA protocol and the nodes use their computed schedules. When node mobility results in a topology change, the contention protocol must be run again. However, these spatial reuse TDMA protocols can become unstable if the rate of mobility outpaces the rate at which the transmission schedules can be updated. More-

0-7803-6451-1/00/$10.00 © 2000 IEEE

over, scarce bandwidth resources are lost to the contention protocol.

In [5], we proposed a principally new approach to create adaptive and scalable MAC solutions. The novel approach, using a paradigm from computational learning theory [3], develops a *meta-protocol framework* that implements a higher layer of adaptivity on top of the existing MAC protocols. Specifically, we introduced a technique that automatically combines any set of existing MAC protocols into a single protocol, resulting in an *aggregated protocol* that has provable optimality properties, as shown in [5]. Thus, we assume that a number of existing MAC protocols are available as *components* and our "meta-MAC" protocol works on top of them, optimally combining their individual transmission decisions into a final decision. The combination is continually updated according to the *local* feedback information available at each node.

In this paper, we show that our meta-protocol technique provides a general way to automatically *optimize* critical MAC protocol parameters without any prior knowledge of the actual network conditions (e.g., traffic load or topology), or how they will change over time. Furthermore, this optimization is completely distributed and requires no control information to be exchanged between nodes. In one of our examples we use the approach to find the proper transmission probability for nodes operating in a LAN, and demonstrate that our method outperforms the more traditional techniques described above. We also apply the meta-protocol approach to the scheduling update problem of spatial reuse TDMA protocols, and show that our method automatically finds near optimal transmission schedules, without any exchange of control messages.

THE META-PROTOCOL FRAMEWORK

To simplify our discussion, we restrict our attention to slotted time and assume immediate perfect feedback is available at the end of each slot (preliminary results on the effect of imperfect channel feedback can be found in [5]). The actual way of computing the combined transmission decision in each time slot by our meta-MAC protocol is based on a weighted average of the decisions made by each component protocol. The final decision to transmit in a slot is then made using randomization based on the weighted sum. Based on the channel feedback available at the end of a slot, the individual component protocol weights are adjusted according to the outcome of the slot. A component protocol that contributed to a wrong decision will undergo a reduction in its weight. We call the method the *Randomized Weighted Majority (RWM) Meta-protocol.*

Figure 1 illustrates M MAC protocols, $P_1, \dots, P_M$ to be combined at all nodes. The final decision, $\tilde{D}_t \in \{0,1\}$, whether to transmit in a slot is determined by appropriately combining the decisions of the M component protocols.

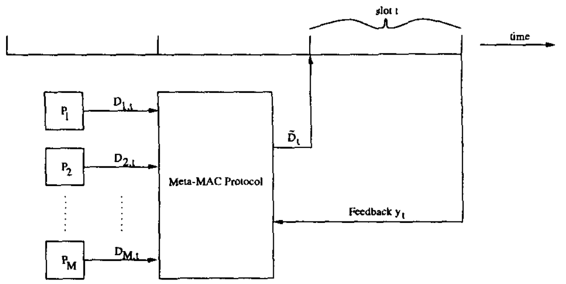

Fig. 1. Meta-MAC protocol model.

Each protocol P_i runs independently, producing a decision $D_{i,t}$, $1 \leq i \leq M$, in each slot t. The value of $D_{i,t}$ is a real number between 0 and 1, where $D_{i,t} = p$ is interpreted to mean that protocol P_i would transmit in slot t with probability p. No assumptions are made concerning how each component protocol reaches its decision. Instead, the final decision D_t is computed as a function of the weighted average of the $D_{i,t}$ values:

$$D_t = F\left(\frac{\sum_{i=1}^{M} w_{i,t} D_{i,t}}{\sum_{i=1}^{M} w_{i,t}}\right). \tag{1}$$

The function F can be chosen in several ways. For this paper, we chose $F(x) = x$. In other words, D_t is simply the weighted average of the $D_{i,t}$ values, which is then rounded, using randomization, to 0 or 1 to obtain the final *binary* decision $\tilde{D}_t$ for slot t.

The meta-protocol at each node maintains the weights used in (1). The positive number $w_{i,t}$ is the weight of protocol P_i for slot t. At the end of each slot the weights at the given node are updated using the channel feedback. We do not restrict the nature of the feedback, we only assume that from the feedback we can conclude whether the decision was right or wrong. For example, if collision occurs, then the decision to transmit was wrong. On the other hand, a successful transmission implies the decision was correct. The weight update algorithm works as follows. Let y_t denote the feedback at the end of slot t:

$$y_t = \begin{cases} 1 & \text{if the decision in slot } t \text{ was correct} \\ 0 & \text{if the decision in slot } t \text{ was incorrect} \end{cases}$$

Then the correct decision, z_t, can be retrospectively computed from the feedback as:

$$z_t = \tilde{D}_t y_t + (1 - \tilde{D}_t)(1 - y_t). \tag{2}$$

That is, $z_t = \tilde{D}_t$ if $y_t = 1$; otherwise $z_t = 1 - \tilde{D}_t$. Of course, we cannot simply set the decision for slot t according to z_t, since z_t becomes known only at the end of the slot. Using z_t, the weights are updated according to the following simple exponential rule:

$$w_{i,t+1} = w_{i,t} \cdot \mathrm{e}^{-\eta|D_{i,t}-z_t|}. \tag{3}$$

The term $|D_{i,t} - z_t|$ in the exponent represents the deviation of protocol i from the correct decision. If this deviation is zero, then the weight of P_i remains unchanged. Otherwise the weight of P_i decreases with increasing deviation. Due to the normalization in (1), the relative weight of those protocols that made a correct decision will grow, while of those which made a mistake will shrink. The constant $\eta > 0$ controls the rapidity of the weight change and thus has a great influence on the stability and convergence speed of the meta-protocol. The effect of η is still being studied, however some preliminary results are published in [5].

Note that the direct use of (3) can cause underflow in the number representation, since the weights decrease monotonically. This problem is easily solved in practice, by renormalizing the weights after each update. One can also set a minimum value below which no weight can drop. Renormalization does not change the *relative* sizes of the weights and since they are only used in a normalized way in computing the combined value D_t by (1), therefore, only their relative sizes matter.

Having introduced the needed concepts, we can now summarize our meta-protocol.

RWM Meta-protocol

Initialization: set all weights to 1.

For each slot t do:

- *At the beginning of slot t*:
  - Collect the component decisions $D_{1,t}, \dots, D_{M,t}$.
  - If there is no packet to send, then set $\tilde{D}_t = 0$; Else compute D_t according to (1).
  - Probabilistically generate the final binary decision $\tilde{D}_t$,, where $\Pr(\tilde{D}_t = 1) = D_t$ and $\Pr(\tilde{D}_t = 0) = 1 - D_t$.
  - If $\tilde{D}_t = 1$, then transmit in slot t; Otherwise refrain from transmission.
- *At the end of slot t*:
  - Compute $z_t = \tilde{D}_t y_t + (1 - \tilde{D}_t)(1 - y_t)$ using the decision $\tilde{D}_t$ and the feedback y_t.
  - Update all of the weights according to (3).

EXAMPLES OF MAC LAYER OPTIMIZATION

Although designed to combine different sets of MAC protocols, the meta-protocol can also combine several versions of the same protocol using different parameters. In this way, our meta-protocol framework is capable of automatically optimizing critical MAC protocol parameters. In this section, we investigate the application of the meta-protocol framework to MAC layer optimization. Specifically, we will find the optimal transmission probability in a LAN using a combination of several p-persistent slotted Aloha protocols. In addition, we will use our meta-protocol approach to find the optimal transmission schedules in a static multi-hop wireless network by combining a large number of simple TDMA transmission schedules.

Optimizing Transmission Probabilities

In this example, each node combines the same M p-persistent slotted Aloha protocols, each with a different transmission probability. Thus, in a given slot t, each component protocol P_i simply returns a decision $D_{i,t}$ equal to its transmission probability. The meta-protocol combines these decisions, and adjusts the relative weights based on the outcome of slot t. Those component protocols whose transmission probabilities are far from the optimum will have their weights for future contribution to the overall decision $\tilde{D}_t$ reduced. Likewise, component protocols that have near-optimal transmission probabilities will have their relative weights increased. In this way, the meta-protocol should find the best transmission probability p for a given traffic load.

To validate this claim, we simulated a local area network consisting of $N = 100$ nodes. When the network is saturated, i.e., each node has always packets to send, the probability of a successful transmission is given by $p(1-p)^{N-1}$. In such a scenario, the value of p must be $1/N$ to obtain optimal throughput. Consequently, the number of component protocols should be $M > \log(N)$. For this experiment, we combine $M = 15$ p-persistent slotted Aloha protocols where each protocol P_i has transmission probability $p_i = 1/2^i, i = 1, \dots, M$.

For comparison purposes, we also simulated the stabilized slotted Aloha (SSA) protocol presented in [7]. The SSA protocol relies on a collision resolution technique similar to the binary exponential backoff algorithm. However, the transmission probability of the SSA protocol is not reset to one when a packet has been transmitted successfully. Instead, the transmission probability is multiplied and divided by a constant $q > 0$ whenever a collision or empty slot is detected, respectively. In this way, the SSA protocol increases fairness and avoids insta-

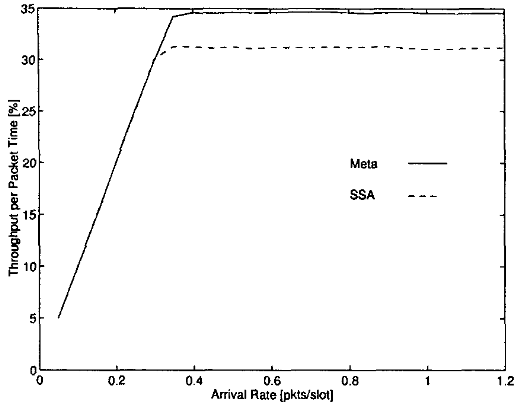

Fig. 2. Throughput comparison of the meta-protocol and SSA in a LAN.

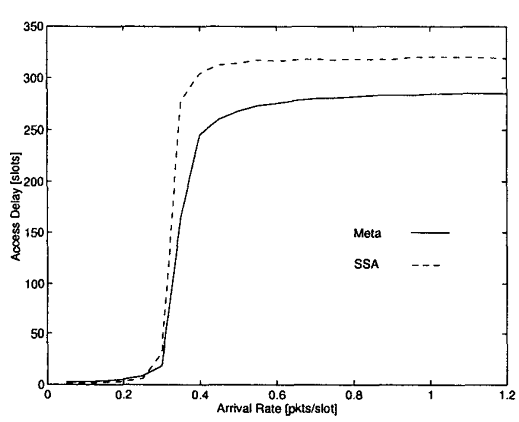

Fig. 3. Access delay comparison of the meta-protocol and SSA in a LAN.

bility at high loads. In our examples the multiplier constant is set to $q = 1/2$.

For each protocol, we collected numerical data pertaining to two key performance metrics — throughput and average access delay. The throughput measures how effectively the network transmits packets between nodes. The average access delay measures the average time needed to successfully transmit a packet. The results are depicted in Figures 2 and 3 as a function of the overall arrival rate (i.e., traffic load).

As it can be observed, the meta-protocol outperforms the SSA protocol. This can be attributed to the fact that the transmission probability of the SSA protocol is limited to powers of $1/2$. Specifically, this protocol "jumps" between probabilities $1/2^i$ and $1/2^{i+1}$ which may bound the optimal transmission probability, p_{opt}, for the current slot, i.e., $1/2^i < p_{opt} < 1/2^{i+1}$. On the other hand, the weight distribution of the meta-protocol allows it to fine tune its transmission probability in a much more continuous manner. Consequently, the meta-protocol can converge to the correct p_{opt}, giving rise to improved performance.

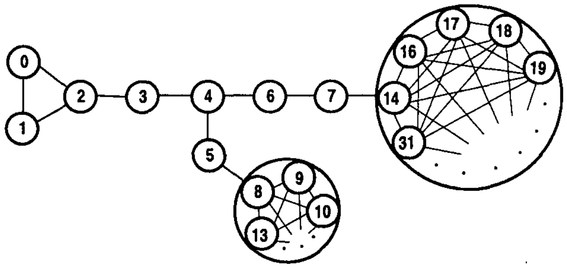

Fig. 4. Static multi-hop network topology.

According to [7], the choice of q affects the best throughput the protocol can achieve, and how quickly it can converge to that value. If q is very small, then the best throughput of the SSA protocol will be that of the meta-protocol. The meta-protocol can adapt more quickly since the adaptation only depends on the η value that, unlike q, does not introduce forced granularity.

Thus when combining protocols of the same type with varied parameters, this example shows that the meta-protocol automatically selects the protocol with the best parameters for the given network load.

Optimizing Transmission Schedules

In this example, each node has the same combination of M simple TDMA transmission schedules, each with different frame lengths (measured in slots) and slot assignments. The goal of this experiment is to show that the meta-protocol will automatically converge to the frame length closest to the density in each part of the network, and make non-conflicting slot assignments to each node.

To validate this claim, we simulated a static multi-hop network consisting of 32 nodes using the topology illustrated in Figure 4. Here, the small circles represent nodes (with the given node identifier) and the lines connecting the circles represent bi-directional wireless links. The large circles represent fully connected sub-networks of size 6 and 18, involving nodes 8–13 and 14–31, respectively.

In this experiment, the frame lengths will be powers of two since it is essential that the different schedules of neighboring nodes interact correctly to achieve non-conflicting schedules, using frame lengths that are powers of two ensures that smaller frames can be embedded in larger ones. Thus, there are at most

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frame Size	4	4	4	4	4	4	4	4	8	8	8	8	8	8	16	32
Assigned Slot	2	3	0	1	2	3	0	2	5	1	2	6	4	0	3	22

Node	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Frame Size	32	32	16	32	32	16	16	16	32	16	32	32	32	16	32	32
Assigned Slot	21	17	2	19	5	13	15	7	10	14	1	9	25	11	27	26

TABLE I

FRAME LENGTH AND SLOT ASSIGNMENTS MADE BY THE META-MAC PROTOCOL.

$m = \lceil log_2 N \rceil$ different frame lengths employed. For each TDMA frame length, L_j, there are 2^j distinct schedules, each with a transmission right in a different slot. Thus, at each node i, there is a total of $M = \sum_{j=1}^{m} 2^j = 2^{m+1} - 1 = O(N)$ component TDMA protocols combined (i.e., in our example there are 63 component protocols contending at each node).

Initially, each node is assigned all slots from each length TDMA schedule, i.e., the weights for all component protocols are equal. The meta-protocol reaches a non-conflicting schedule if there is exactly one component TDMA protocol with a normalized weight nearly equal to one. This specific component protocol represents the optimal TDMA transmission schedule for the given node.

Table I shows the TDMA frame length and slot assignment determined by the meta-protocol for the network in Figure 4. The meta-protocol converged to near optimal frame lengths with no conflicts in the slot assignment. Since there are 18 nodes in the larger fully connected sub-network, some nodes must be assigned a frame size of 32 rather than 16.

Thus, by combining TDMA protocols with different frame sizes, the meta-protocol automatically converged to a near-optimal TDMA transmission schedule for each node in the example multi-hop network.

CONCLUSIONS

This paper presented a systematic and automatic method for dynamically optimizing MAC protocol parameters. By combining several versions of the same protocol, the meta-protocol is capable of performing on-line optimization of critical MAC parameters without advance knowledge of future network conditions or that of their fluctuation over time. Furthermore, this dynamic optimization is achieved without any centralized control or message exchanges between nodes. We applied our technique to the problem of finding the proper transmission probability for nodes operating in a LAN, and demonstrated the the resulting meta-protocol outperformed more traditional techniques. We also used our method in a synchronous multihop network, and showed that the meta-protocol found the optimal collision-free transmission schedules for the actual node density of the network. Thus, we conclude that our simple and practical combination algorithm can be greatly used for MAC protocol optimizations.

REFERENCES

[1] D. Aldous. "Ultimate Stability of Exponential Backoff Protocol for Acknowledgement Based Transmission Control of Random Access Communication Channels," *IEEE Transactions on Information Theory*, vol. 33, no. 2, 1987, pp. 219-223.

[2] D. Bertsekas and R. Gallager. "Data Networks," Prentice Hall, Inc., 1992.

[3] N. Cesa-Bianchi, Y. Freund, D. Helmbold, D. Haussler, R. Schapire and M. Warmuth. "How to Use Expert Advice," 25^{th} *Annual ACM Symposium on the Theory of Computing (STOC'93)*, San Diego, CA, May 1993.

[4] I. Chlamtac and S. Pinter. "Distributed Node Organization Algorithm for Channel Access in Multi-hop Packet Radio Networks," *IEEE Transactions on Computers*, vol. 36, no. 6, 1987.

[5] A. Faragó, A.D. Myers, V.R. Syrotiuk, and G.V. Záruba. "Meta-MAC Protocols: Automatic Combination of MAC Protocols to Optimize Performance for Unknown Conditions," in press for *IEEE Journal on Selected Areas in Communications*, Special Issue on Analysis and Synthesis of MAC Protocols, September, 2000.

[6] J. Frigon and V. Leung. "A pseudo-Bayesian Aloha Algorithm with Mixed Priorities for Wireless ATM," *Proceedings IEEE Personal, Indoor and Mobile Radio Communications*, vol. 1, 1998, pp. 45-49.

[7] D. Jeong and W. Jeon. "Performance of an Exponential Backoff Scheme for Slotted-ALOHA Protocol in Local Wireless Environment," *IEEE Transactions on Vehicular Technology*, Vol. 44, No. 3, 1995.

[8] F. Kelly. "Stochastic Models of Computer Communication Systems," *Journal of the Royal Statistical Society*, vol. 47, 1985, pp. 470-479.

[9] R. Rivest. "Network Control by Bayessian Broadcast," Report MIT/LCS/TM-285, MIT Laboratory for Computer Science, 1985.

[10] D. Sala, J. Limb, and S. Khaunte. "Adaptive Control Mechanism for Cable Modem MAC Protocols," *Proceedings of IEEE INFOCOM*, vol. 3, 1998, pp. 1392-1399.

[11] C. Zhu and S. Corson. "A Five-Phase Reservation Protocol (FPRP) for Mobile Ad Hoc Networks," *Proceedings of IEEE INFOCOM*, 1998.