Learning Perfect Coordination with Minimal Feedback in Wireless Multi-Access Communications

William Zame
Departments of Economics and Mathametics
University of California, Los Angeles
Los Angeles, CA 90095
Email: zame@econ.ucla.edu

Jie Xu and Mihaela van der Schaar
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90095
Email: jiexu@ucla.edu, mihaela@ee.ucla.edu

***Abstract*—Coordination is a central problem whenever stations (or nodes or users) share resources across a network. In the absence of coordination, there will be collision, congestion or interference, with concomitant loss of performance. This paper proposes new protocols, which we call *perfect coordination (PC) protocols*, that solve the coordination problem. PC protocols are completely distributed (requiring neither central control nor the exchange of any control messages), fast (with speeds comparable to those of any existing protocols), fully efficient (achieving perfect coordination, with no collisions and no gaps) and require minimal feedback. PC protocols rely heavily on *learning*, exploiting the possibility to use both actions and silence as messages and the ability of stations to learn from their own histories while simultaneously enabling the learning of other stations. PC protocols can be formulated as finite automata and implemented using currently existing technology (e.g., wireless cards). Simulations show that, in a variety of deployment scenarios, PC protocols outperform existing state-of-the-art protocols – despite requiring much less feedback.**

I. Introduction

In networks where a number of stations share common network resources (e.g., wireless spectrum), if more than one station attempts to access the same resource in the same slot/period, collision, congestion or interference, with concomitant loss of performance, will typically occur. Coordination among stations has the potential to avoid conflict and improve utilization of network resources – especially to increase throughput and decrease delay. Because central control and the exchange of control messages between stations are typically wasteful of resources and often impossible, so the first *desideratum* for a coordination protocol is that it be *distributed.* Because gathering information (sensing) also uses resources, a second *desideratum* is that the protocol use *minimal feedback.* A third *desideratum* is that it achieve *maximal goodput*. The search for such protocols is by now quite old (e.g. slotted Aloha [1][2] and the distributed coordination function [3][4]). These protocols are widely studied and used, but both fail the last two *desiderata*: they require substantial feedback and, because they do not eliminate collisions and empty slots, do not achieve maximal goodput.

Financial support was provided by the National Science Foundation under grants 0617027 and 1218136, by the UCLA Academic Senate Committee on Research and by the Einaudi Institute for Economics and Finance (Rome). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agency.

This paper proposes and analyzes new protocols that meet all three of these *desiderata*; we call them *Perfect Coordination protocols*. We propose two protocols, the first designed for settings in which the number of stations is known, the second designed for settings in which the number of stations is not known (but an upper bound is known); each of these settings represents a realistic set of environments. The protocols we describe are perfectly distributed and require no central control, no exchange of control messages between stations and minimal feedback: stations that transmit learn whether or not their transmission is successful; stations that are idle can not/do not sense the channel and hence learn *nothing*. The assumption of minimal feedback – which distinguishes the present work from all of the literature of which we are aware – makes coordination much more difficult but is important: sensing activity on the channel when not transmitting requires the expenditure of energy and is prone to serious errors (because of the difficulty of distinguishing the traffic of other stations from ambient noise) which increase the fragility of protocols. The protocols we introduce converge as rapidly and with probability as high as previous protocols, and achieve perfect coordination, not merely zero collision – hence greater goodput and smaller delay – *despite requiring much less feedback.*

Our protocols lead stations to learn about the evolving state of the system, to condition their pattern of actions on what is learned, and to enable the learning of other stations. In comparison with previous protocols, stations learn more and use more of what they have learned – especially about the pattern of actions of *other* stations. Remarkably, stations can learn all of this *solely on the basis of their own histories of successful and unsuccessful transmissions*. This is possible because stations learn *cooperatively*. This learning exploits two opposite facets of the environment: first, that actions of stations can be used as implicit messages, and second that silence can also be a message.[1] In addition to cooperative learning, actions as signals and silence as a signal, our protocols make extensive use of the idea that stations that are indistinguishable *ex ante* and randomize in an identical fashion may still experience different realizations of that randomization and hence become distinguishable *ex post*. We emphasize that the protocols we

[1] That the apparent absence of information can in fact *be* information has been well-known at least since Sherlock Holmes. Recall the exchange between Inspector Gregory of Scotland Yard and Sherlock Holmes in Arthur Conan Doyle's story 'Silver Blaze' [5]: Gregory: "Is there any other point to which you would wish to draw my attention?" Holmes: "To the curious incident of the dog in the night-time." Gregory: "The dog did nothing in the night-time." Holmes: "That was the curious incident."

	Lee+Walrand (ZC)	*Fang et al (L-ZC)*	*Park + van der Schaar*	*He et al (SRB)*	*This paper (PC)*
Channel Sensing When Not Transmitting	Yes	Yes	Yes	Yes	No
Fully Coordinated	No	No	No	No	Yes
Learn # of Stations	No	No	No	No	Yes
Number of Stations	Unknown	Unknown	Known/ Unknown	Unknown	Known/ Unknown
Convergence (Small Networks)***	Very Fast	Very Fast	Fast	Medium	Very Fast
Convergence (Large Networks)***	Fast	Fast	Medium	Slow	Fast

* Convergence speed is the number of slots required such that the system enters the steady state (perfect coordination or zero collision) with probability 0.999. (Very Fast: < 100 slots; Fast : 100 - 500 slots; Medium: 500 – 5000 slots; Slow: > 5000 slots.)
** Small Networks: number of stations from 1 to 10; Large Networks: number of stations from 20 to 50.

TABLE I. COMPARISON OF ASSUMPTIONS, RESULTS

propose require only finite memory, can be formulated as finite automata, and can be implemented using current wireless cards without requiring the development or deployment of any new hardware; see [6] for instance. We do require that local clocks of the stations be synchronized; this could be accomplished by time stamping from the access point (as in [13]), by GPS coordination (as in [12]) or by various other methods.

The idea of utilizing collisions as a coordination device can be found in various existing network protocols, most notably in various versions of slotted Aloha protocols and in exponential backoff protocols. By and large, these protocols, are designed in an *ad hoc* manner and utilize available past information in a limited way, yielding only limited performance improvements. Coordination protocols [7]-[14] have received substantial attention in the literature; [7]-[11] are closest to the present work.[2] Table I below provides some comparisons between the current work and those papers. We defer more detailed comparisons until we present simulations in Section VI, but for now we would like to highlight a few points. First, all of these other papers assume that stations can sense the state of the channel when they are not transmitting; we do not. Second, all of these papers propose protocols that *do not* learn the number of stations and *do not* achieve perfect coordination, hence yield sub-optimal (possibly highly-suboptimal) goodput; our Perfect Coordination protocols *do* learn the number of stations and *do* achieve perfect coordination, hence yield optimal goodput. Third, the speed of our Perfect Coordination protocols is comparable to that of these other protocols, despite the fact that we assume much less feedback.[3]

In what follows, Section II presents our protocols, establishes analytic estimates of convergence probabilities and speeds, discusses robustness. Section III uses simulations to provide performance analysis of our protocols and comparisons with other protocols. Section VI concludes.

II. PERFECT COORDINATION PROTOCOLS

A. The number of stations N is known

For a known number of stations N, we propose a family $\Phi(N,K,p)$ of Perfect Coordination protocols, depending on a length parameter K and a vector p of probabilities (strictly between $0,1$) to be chosen by the administrator. (Different choices of K,p lead to different probabilities and speeds of convergence.) Each protocol is divided into *phases*, each phase is divided into *cycles*, each cycle is divided into *slots*. In the *Learning* phase, stations learn their place in an endogenously determined sequencing of all the stations; in the *Transmission* phase, stations transmit in the sequence determined in the first stage.

By definition a protocol specifies random actions conditional on personal history; however $\Phi(N,K,p)$ uses only some of the information contained in those histories (in addition to the parameters K,p): (1) the current phase; (2) the location of the current cycle within the current phase; (3) the location of the current slot within the current cycle; (4) personal information about the previous slot; (5) a summary statistic of personal history: the station's *index*, an integer between 0 and N.

A station's index is a convenient way for the station to keep track of whether it has at some point won a lottery (as described below) and, if so, when. Each station knows only its *own* index. At any slot, a station's index will depend on its index in the previous slot and the station's personal observation about the previous slot. We initialize so that $index(z)=0$ for all stations z.

Fix a positive integer K and an N-vector $p=(p_1,\dots,p_N)$ of probabilities strictly between $0,1$. The protocol $\Phi(N,K,p)$ begins in the Learning phase.

- In slot 1, all stations randomize: they transmit with probability p_1 and remain silent with the complementary probability $(1-p_1)$. This randomization creates an *endogenous lottery*.
- In slot 2 each station conditions on what it observes about slot 1 (its personal history). A station z that transmitted in slot 1 and was successful – *won the lottery* – sets its $index(z)=1$ and transmits with probability 1 in the second slot (and in every succeeding slot in the current cycle); a station that did transmit but was unsuccessful – did not win the lottery – or did not transmit (and hence observed nothing) randomizes again in slot 2, with the same probabilities $p_1, 1-p_1$.
- The process continues through slot K (one cycle of the Learning phase).
- At the end of the first through N-th cycles of the Learning phase, the protocol repeats the process above, with three changes:
  - stations that won a lottery (those with $index(z)>0$) remain silent (inactive) throughout the current cycle;
  - stations that have not won a lottery (those with $index(z)=0$) randomize with probabilities $p_n, 1-p_n$;
  - a station z that wins the lottery in cycle n sets $index(z)=n$[4]

  At any point in time, $index(z)>0$ if and only if station z won the lottery in cycle $index(z)$ of the Learning phase; if $index(z)=0$ then station z has not won a lottery in any cycle.

[2][12] approach the problem somewhat differently and assume a very different information structure: stations can observe whether or not there was a successful transmission but cannot distinguish between a collision and ambient noise on an idle channel. This seems an unrealistic assumption, especially because such observation would almost surely entail a large error rate, which would be incompatible with the protocols they propose.

[3]In fact, if we consider duration rather than number of slots, Table I underestimates the speed of our protocols because our protocols do not require full slots. We will discuss this is more detail later.

[4]A station that wins the lottery in some slot during cycle n transmits in every succeeding slot in that cycle, so at most one station can transmit successfully in any given cycle and so no two stations can have the same index.

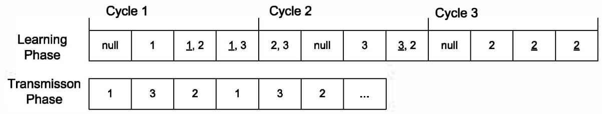

Fig. 1. Illustration of PC protocol when N is known. ($N = 3, K = 4$)

- At the end of N cycles of the Learning phase, stations enter the Transmission phase. If $index(z) = n > 0$ then station z transmits in slot n and only in slot n; if $index(z) = 0$, then station z transmits in *every* slot of the Transmission phase.
- After N slots (one cycle) of the Transmission phase *all* stations will have had the *same* experience: either at least one collision or success whenever transmitting. If all stations experienced at least one collision, then stations are not perfectly coordinated; in that case, all stations set *index* $= 0$ and return to the beginning of the protocol. If all stations experienced success whenever transmitting, then stations are perfectly co-ordination; in that case, all stations return to the beginning of the Transmission phase and repeat the Transmission phase indefinitely.

Both individual and cooperative learning play important roles in this protocol. In the Learning phase, stations are creating an endogenous lottery: creating the lottery is cooperative (although it requires no communication, only conformity to the protocol); winning the lottery is individualistic. A station that wins the lottery in the current cycle transmits in every remaining slot of the current cycle, and stations that have won lotteries in previous cycles do not participate in the current cycle; both of these cooperative activities avoid interference with the learning of other stations. Learning of a different sort occurs in the Transmission phase: all stations learn either that perfect coordination has been achieved, so that they should remain in the Transmission phase indefinitely, or that perfect coordination has not been achieved, so that they should re-initialize and begin the Learning phase again. Figure 1 illustrates the operation of this protocol with $N = 3$, $K = 4$. Station 1 wins the lottery in Cycle 1, Slot 2; Station 3 wins the lottery in Cycle 2, Slot 3; Station 2 wins the lottery in Cycle 3, Slot 2.

1) The Probability of Convergence: It is straightforward to calculate the probability that perfect coordination is achieved in one round (N cycles) of the Learning phase. From this we can calculate the probability that the protocol converges to perfect coordination within a given number of rounds and hence in a given number of slots. (Translation from rounds into slots is straightforward: if perfect coordination is achieved after R rounds of the Learning Phase then $RNK+(R-1)N$ slots will have elapsed.) As a consequence, it follows that, for every N and every choice of K, p, convergence to perfect coordination in finite time occurs with probability 1; i.e., for each $\varepsilon > 0$ there is some T so that the probability of convergence to perfect coordination in T slots or less is at least $1 - \varepsilon$.

Theorem 1: Fix $N, K, p = (p_1, \dots, p_N)$ (with $0 < p_n < 1$ for all n). The probability that the protocol $\Phi(N, K, p)$ achieves perfect coordination in no more than R rounds (no more than $RNK + (R-1)N$ slots) is exactly

$$1-\left(1-\prod_{n=1}^{N}\left\{1-[1-(N-n+1)p_n(1-p_n)^{N-n}]^K\right\}\right)^R \quad (1)$$

Proof: Omitted due to space limitation. See [15]. ■

Corollary 1: For every N and every choice of $K, p = (p_1, \dots, p_N)$ (with $0 < p_n < 1$ each n), the protocol $\Phi(N, K, p)$ converges to perfect coordination in finite time with probability 1.

Proof: For fixed N, K, p the probability in (1) tends to ∞ with R. ■

2) A Numerical Estimate: The expression (1) is somewhat intractable and does not provide a simple rule for an administrator who wishes to guarantee a given probability of convergence. However a judicious choice of the probability vector p and some simple manipulation yields (Theorem 2) both a tractable estimate and a useful procedure for choosing K, R to guarantee a given probability of convergence. We first isolate two simple lemmas.

Lemma 1: For $x > 1$, $1 - \left(\frac{x-1}{x}\right)^{x-1}$ is a decreasing function of x and

$$\lim_{x\to\infty}\left(\frac{x-1}{x}\right)^{x-1} = \frac{1}{e}$$

Proof: Take logarithms and differentiate; then use the fact that log is a concave function to derive the first assertion and L'Hopital's rule to derive the second assertion. ■

Lemma 2: For ℓ a strictly positive integer

$$\max_{0<p<1} \ell p(1-p)^{\ell-1} = \left(\frac{\ell-1}{\ell}\right)^{\ell-1} = \left(1-\frac{1}{\ell}\right)^{\ell-1}$$

Moreover, the maximum is attained when $p = 1/\ell$.

Proof: Apply the the first-order condition. ■

Theorem 2: Fix N and set $p^* = (p_1^*, \dots, p_N^*)$, $p_n^* = \frac{1}{N-n+1}$.

(i) The probability that the protocol $\Phi(N, K, p)$ converges to perfect coordination in at most R rounds of the learning phase ($RNK + (R-1)N$ slots) is at least

$$1-\left(1-\left\{1-\left[1-\left(\frac{1}{e}\right)\right]^K\right\}^N\right)^R$$

(ii) Fix $\varepsilon > 0$. In order that $\Phi(N, K, p)$ converges to perfect coordination in at most R rounds of the learning phase with probability at least $1 - \varepsilon$ it is sufficient that

$$K \geq \frac{\log N - (1/R)\log\varepsilon}{-\log[1-(1/e)]}$$

Proof: Omitted due to space limitation. See [15]. ■

To make Theorem 2 more concrete, take $N = 10$ and $\varepsilon = 10^{-4}$ so that we seek convergence with probability at least 0.9999. In view of (ii), it is sufficient to take $R = 1, K = 25$ or $R = 2, K = 15$. The former choice would seem to be superior to the latter (250 slots vs. 310 slots) but note that if $K = 15$ then convergence actually occurs in one round (150 slots) with probability greater than 0.99 – so using the latter choice leads to a much smaller *expected* time to convergence.

As the simulations in Section VI will make clear, these analytic estimates are not sharp; but they do have the virtue of simplicity, and the rate of convergence they provide is already quite good (especially when interpreted in terms of time rather than number of slots).

B. The number of stations N is Unknown

Now we assume that the true number N of stations is unknown but that an upper bound $N^{\max}$ is known. We could apply any of the protocols $\Phi(N^{\max}, K, p)$ in this setting, but the result might not be very satisfactory: If $N < N^{\max}$ then $\Phi(N^{\max}, K, p)$ will eventually reach the situation in which the N stations transmit in turn but in a cycle of length $N^{\max}$, so that there will be gaps and throughput will be less than optimal; moreover since $\Phi(N^{\max}, K, p)$ executes at least $N^{\max}$ cycles in the Learning phase, the protocol might take much longer than is necessary – much longer than necessary if N is much smaller than $N^{\max}$. We avoid both of these problems by modifying the Learning phase slightly and adding two intermediate phases so that the protocol moves into the Transmission phase as soon as all stations have learned their indices and closes gaps along the way.

As before, our protocols depend on $N^{\max}$ and on parameters that can be chosen by the administrator: a length parameter K (a positive integer) and a vector of probabilities parameters $q = (q_1, \ldots, q_{N^{\max}})$, each q_m strictly between 0, 1. The protocol $\Psi(N^{\max}, K, q)$ consists of four phases: *Learning-to-Win*, *Rectifying-the-Count*, *Learning-the-Losers*, *Coordinated Transmission*.

As before, $\Psi(N^{\max}, K, q)$ uses only some of the information contained in personal histories (in addition to K, q): (1) the current phase; (2) the location of the current cycle within the current phase; (3) the location of the current slot within the current cycle; (4) the station's personal information about the previous slot; (5) two summary statistics of the station's personal history: *stationcount* (an integer between 0 and $N^{\max}$), which will tell the station when to move to the Coordinated Transmission phase; and *index* (either $*$ or an integer between 0 and $N^{\max}$), which will tell the station in which slot of the Coordinated Transmission phase to transmit).

Each station knows its *own* statistics – *stationcount* and *index*. At any slot, a station's *stationcount* and *index* will depend on the values in the previous slot and the station's observation of the previous slot. It is convenient to refer to a station z as a *winner* if *index*$(z) > 0$, a *loser* if *index*$(z) = 0$ and as a *waiter* if *index*$(z) = *$. Winners are stations that have won a lottery *and* learned their indexed, waiters are stations that have won a lottery but have not yet learned their index, losers are stations that have not yet won a lottery. At various times, some of these categories will be empty (e.g., at the beginning of the protocol all stations are losers); at any moment in time there is at most one waiter. Because stations always know their own statistics, they know to which category they belong. The protocol begins in the Learning-to-Win phase.

- In the Learning-to-Win phase, waiters and winners remain silent throughout.[5] Losers randomize: if this is the M-th time the protocol has entered the Learning-to-Win phase, Losers transmit with probability q_m and remain silent with the complementary probability $(1 - q_m)$, where $m = \min\{N^{\max}, M\}$. (Stations can compute m without computing M because they only need to count as high as $m = N^{\max}$ – and then stop counting.) As before, this creates an endogenous lottery.
- A station z that wins the lottery in some slot in the current Learning-to-Win cycle sets *index*$(z) = *$ and transmits in every subsequent slot; losers continue to randomize.
- The process continues through slot K (one cycle of Learning-to-Win).
- At the end of each cycle of Learning-to-Win, the protocol enters the Rectifying-the-Count phase which consists of m slots. In each slot: losers remain silent; the waiter (if one exists) transmits in every slot; winners transmit in the slot corresponding to their index. An inductive argument shows that if there are $W < m$ winners at this stage then their indices are $1, \ldots, W$. Hence if there is a waiter, all winners experience a collision and the waiter experiences collisions in slots $1, \ldots, W$ and success in slot $W + 1$. The waiter sets its *index* equal to $W+1$; the waiter and all winners set their *stationcount* equal to $W + 1$. Note that at this point $W + 1$ is indeed the *correct* number of winners.
- At the end of one cycle of Rectifying-the-Count, the protocol enters the Learning-the-Losers phase which again consists of m slots. In every slot: losers transmit in every slot; winners transmit in the slot corresponding to their index. (There are no waiters at this point.) If there are no losers, the winners all experience success; if there are losers then both winners and losers experience collision. Hence after this cycle, all stations know whether or not losers remain.
- If no losers remain after Rectifying-the-Count and Learning-the-Losers, then every station moves to the Coordinated Transmission Phase and all stations transmit in sequence according to their indices, in a cycle of length *stationcount* (w) for any station w. This is perfect coordination and repeats indefinitely. If some loser or losers remain after Rectifying-the-Count and Learning-the-Losers then all stations return to Learning-to-Win and proceed as above.

Again, both individual and cooperative learning play important roles here. In the Learning-to-Win phase there is little to add to what we have already said about the Learning phase in the protocol $\Phi(N, K, p)$. In the Rectifying-the-Count phase, winners have learned the current total number of winners and their own place in the sequence; now winners and the waiter (if there is one) cooperate so that all can learn whether there is a waiter, in which case the waiter becomes a winner, all winners learn the new total number of winners, and all winners learn their places in the sequence. Note that waiters can count the number of current winners because each of them transmits in turn, so that the waiter experiences collisions, but then the current winners are silent, so that the waiter experiences success. In the Learning-the-Losers phase, all stations learn whether there are remaining losers; if so they return to the Learning-to-Win phase, if not they transit to the Transmission phase. Figure 2 provides an illustration of the operation of the protocol, with $N = 3$ and $N^{\max} = 4$. Each cycle of the Learning-to-Win phase consists of $K = 4$ slots. Station 1 wins the lottery in Cycle 1 Slot 2; Station 3 wins the lottery in Cycle 2 Slot 3; there are no winners in Cycles 3, 4; Station 2 wins

[5]In fact there will never be a waiter at this point.

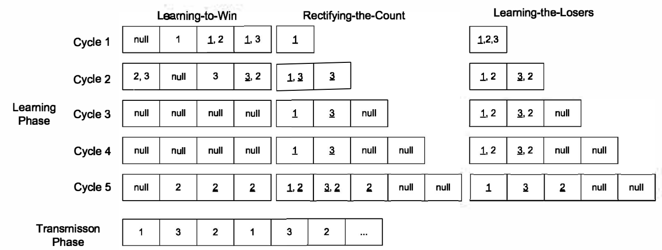

Fig. 2. Illustration of PC protocol when N is known. ($N = 3, K = 4, N^{max} = 4$)

the lottery in Cycle 5 Slot 2. The winners transmit in their corresponding slots in the Rectifying-the-Count and Learning-the-Losers phases.

1) The Probability of Convergence: At this point it is natural to ask for a parallel to Theorem 2 that provides, for each K, q, the precise probability of convergence to perfect coordination in a specified number of slots. However, a closed form expression for the precise probability seems difficult to provide. Instead, Theorem 4 provides convergence estimates for the particular vector of probabilities $q^* = (q_1^*, \ldots, q_{N^{\max}}^*)$ defined by

$$q_1^* = \frac{1}{N^{\max}}, q_2^* = \frac{1}{(N^{\max} - 1)}, \ldots, q_{N^{\max}-1}^* = q_{N^{\max}}^* = \frac{1}{2}$$

As our simulations will show these estimates are not at all sharp (much worse than the estimates from Theorem 3 when N is known), but they are good enough to establish (Corollary 2) convergence in finite time with probability 1. To state the estimate formally, write $B(N, R; \zeta)$ for the probability of getting at least N successes in $R \geq N$ independent trials each of which has probability of success ζ. (As before, we can translate rounds into slots although the translation is more complicated. Learning-to-Win requires K slots in every round; if the current round is r then Rectifying-the-Count and Learning-the-Losers each require $\min\{r, N^{\max}\}$ slots. Hence if $R \leq N^{\max}$ the total number of slots that have occurred in R rounds of the first three phases is $\sum_{r=1}^{R}(K + r + r) = RK + R(R+1)$; if $R > N^{\max}$ then – because there are $R - N^{\max}$ rounds in which Rectifying-the-Count and Learning-the-Losers require $N^{\max}$ slots – the total number of slots is $N^{\max}K + N^{\max}(N^{\max} + 1) + (R - N^{\max})(K + 2N^{\max})$.)

Theorem 3: If N is the true number of stations and $R \geq N$ set

$$\zeta = 1 - \left[1 - \left(\frac{1}{N^{\max} - N + 1}\right)\left(\frac{1}{e}\right)\right]^K$$

The probability that $\Psi(N^{\max}, K, q^*)$ converges to perfect coordination in no more than R rounds is at least $B(N, R; \zeta)$.

Proof: Omitted due to space limitation. See [15]. ■

Corollary 2: For every $N, N^{\max}$ and every choice of K, the protocol $\Psi(N, K, q^*)$ converges to perfect coordination in finite time with probability 1.

Proof: For all N and all $\zeta > 0$ the probability $B(N, R; \zeta)$ tends to ∞ with R. ■

We caution the reader that the estimate in Theorem 3 is very crude and that simulations show that the actual rate and probability of convergence are much better. For instance, suppose $N = 5, N^{\max} = 10, K = 20$. Theorem 3 estimates that the probability of convergence to perfect coordination in 10 rounds is at least .96 – but simulations show that the true probability of convergence to perfect coordination is greater than .9999.

C. Robustness

All protocols have some sensitivity to errors. The most obvious errors in the context we consider are the failure to detect an idle channel (not distinguishing a channel that is busy from a channel that is idle but noisy), and the failure to recognize a successful transmission (loss of acknowledgement). Because our protocols do not require stations that do not transmit to observe anything, our protocols are completely immune to the first of these errors – unlike other protocols that depend heavily on detection of idle channels. Failure to recognize a successful transmission in a given slot during the learning phases of either Φ or Ψ would have exactly the same effect as an unsuccessful transmission in that slot, so a small probability of loss of acknowledgement simply raises the probability of an unsuccessful transmission in a single slot by the same small amount. Failure to recognize a successful transmission in a given slot during the Transmission phase of Φ could result in confusion: some stations would experience success and believe perfect coordination had been achieved, hence repeat the transmission phase, while at least one station would experience collision and believe perfect coordination had not been achieved, hence return to the learning phase. However, the probability that confusion occurs could be made arbitrarily small simply by repeating the Transmission phase several times; confusion would occur only if some station experienced errors in each repetition. If loss of acknowledgement occurs with probability δ then the probability that confusion would occur in a single cycle of the Transmission phase would be $1 - (1-\delta)^N$ but k repetitions would reduce the probability to $1 - (1 - \delta^k)^N$ – and k repetitions require only kN slots. For instance, if $\delta = 10^{-2}$ and $N = 10$, $k = 3$ repetitions would reduce the probability of confusion below 10^{-5} at the cost of only 30 slots – less than 7ms. Repetition would be similarly effective with similarly low cost in the Rectifying-the-Count and Learning-the-Losers phases of the protocol Ψ.

III. Simulations

In this section, we provide simulation results to evaluate the performance of the proposed Perfect Coordination protocols and provide comparisons with existing protocols. For these simulations, we adopt the parameters specified by IEEE 802.11a, as in Table II.

TABLE II. Simulation configuration

Parameters	Values
Packet payload	1024 octets
MAC header	28 octets
ACK frame size	14 octets
Data rate	54 Mbps
PHY header time	20 μs
SIFS	16 μs
DIFS	34 μs

Table III documents the estimated and simulated speed of convergence of our protocols for $N = 4, 8, 16, 24, 32$: the number of slots required for the stations to achieve perfect coordination with pre-specified probabilities $0.99, 0.999, 0.9999$. For N known, we record the estimate implied by Theorem 3;

N is Known						
Prob.		N = 4	N = 8	N = 16	N = 24	N = 32
0.99	Sim.	40	104	240	384	544
	Est.	56	120	272	408	576
0.999	Sim	56	128	320	456	704
	Est.	72	160	336	528	736
0.9999	Sim.	64	160	368	552	768
	Est.	92	192	416	648	864
N is Unknown ($N^{max} = 32$)						
Prob.		N = 4	N = 8	N = 16	N = 24	N = 32
0.99	Sim.	288	442	740	1053	1483
	Est.	1224	1976	2850	2704	1516
0.999	Sim.	380	532	836	1120	1582
	Est.	1656	2511	3302	2998	1592
0.9999	Sim.	456	588	874	1176	1592
	Est.	2106	2998	3682	3226	1626

TABLE III. NUMBER OF SLOTS FOR PERFECT COORDINATION WITH PROBABILITIES .99, .999, .9999

for N unknown we assume $N^{\max} = 32$ and use the estimate implied by Theorem 4. In both cases the simulation results are generated from 10^4 Monte Carlo tests. There are two reasons why convergence is slower when the number of stations N is unknown. As we have noted earlier, when N is unknown, the protocol must go through the additional phases of *Rectifying-the-Count* and *Learning-the-Losers* at least N times; this requires at least $N \times (N+1)$ slots, which is substantial when N is large. More subtly, and more importantly, when N is not known we can tailor the probability parameter to the upper bound $N^{\max}$ – but we cannot tailor it to N itself. However, we note that actual convergence as measured by *time* – rather than number of slots – is very fast even if the number of slots is substantial, especially since during the coordination phase(s) we can use small slots in which stations only send a small payload (e.g. 100 bytes) in each slot. Allowing for packet overhead and signaling intervals, these small slots can be as short as 90 μs. (After perfect coordination is achieved, stations can revert to regular slots, typically 230 μs.) If we use these small slots, the actual simulated time to convergence in the worst case illustrated in Table III is only a few tenths of a second.

Assuming $N^{\max}$ is known but N is unknown, we compare the goodput of our proposed Perfect Coordination protocol against the Zero Collision protocol (ZC) proposed in [7] and the modification L-ZC proposed in [8]. As above, we assume that our Perfect Coordination protocol uses small slots of 90 μs in the coordination phases and regular slots of 230 μs (including a 1024-Byte Payload and overhead) in the Transmission phase. To be generous to ZC, we assume it uses idle slots of 34 μs (i.e. a DIFS duration with a 9μs empty slot) when no stations are transmitting and regular slots of 230 μs when some station is transmitting. In both cases we assume the upper bound on the number of stations is $N^{\max} = 32$, which means that in ZC the contention window size (the number of slots in a cycle) is also $M = 32$. Table IV shows goodput achieved by PC, ZC and L-ZC as a function of the true number of stations N. For each protocol, we show goodput in Mbps and the fraction of theoretically optimal goodput (which in this case is 35.94 Mbps, after taking overhead into account). PC reaches perfect coordination rapidly and achieves optimal goodput independently of N; ZC and L-ZC both leave gaps, but leave fewer gaps – use the available slots more efficiently, hence achieve greater goodput – when N is large than when N is small. (If the size of the contention window is

		N = 4	N = 8	N = 16	N = 24	N = 32
ZC/L-ZC	Mbps	17.58	24.82	31.28	34.24	35.94
	Fraction	0.49	0.69	0.87	0.95	1.00
PC	Mbps	35.94	35.94	35.94	35.94	35.94
	Fraction	1.00	1.00	1.00	1.00	1.00

TABLE IV. GOODPUT COMPARISON FOR VARIOUS NUMBER OF STATIONS ($N^{\max} = 32$)

$M = N^{\max} = 32$ and the true number of stations is N, both ZC and L-ZC leave $M - N$ gaps.)

IV. CONCLUSION

In this paper, we have proposed a new class of MAC protocols that deploy sophisticated learning techniques to achieve perfect coordination. Our proposed protocols are completely distributed, requiring neither any central control nor any exchange of control messages between stations and use minimal feedback. Our results show that despite this minimal feedback the proposed protocols converge very quickly to perfect coordination and yield optimal throughput.

REFERENCES

[1] N. Abramson, "The Aloha system: another alternative for computer communications," in *Proc. the AFIPS Fall Joint Computer Communication*, vol. 37, pp. 281-285, 1970.

[2] L. G. Roberts, "Aloha packet system with and without slots and capture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, pp. 28-42, Apr. 1975.

[3] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[4] IEEE 802.11a, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Highspeed Physical Layer in the 5 GHz Bang*, Supplement to IEEE 802.11 Standard, Sep. 1999.

[5] A.C. Doyle, "Silver Blaze," *Strand Magazine* December 1892.

[6] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, "Wireless MAC processors: programming MAC protocols on commodity hardware," *IEEE INFOCOM 2012: 1269-1277*.

[7] J. Lee and J. Walrand, "Design and analysis of an asynchronous zero collision MAC protocol", Arxiv preprint 0806.3542, 2008.

[8] M. Fang, D. Malong, K. R. Duffy and D. J. Keith, "Decentralised learning MACs for collision-free access in WLANS," *Wireless Networks*, DOI: 10.1007/s11276-012-0452-1, May 2012.

[9] Y. He, R. Yuan, J. Sun, W. Gong, "Semi-random backoff: towards resource reservation for channel access in wireless LANs," *17th IEEE International Conference on Network Protocols (ICNP)*, 2009.

[10] J. Barcelo, C. Cano, B. Bellalta, M. Oliver, "Learning-BEB: avoiding collisions in WLAN," *Eunice Summer School*, 2008.

[11] J. Park and M. van der Schaar, "Medium access control protocols with memory," *IEEE/ACM Trans. Networking*, vol. 18, no. 6, pp. 1921-1934, Dec. 2010.

[12] K. Nakano and S. Olariu, "Randomized initialization protocols for ad hoc networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 7, Jul. 2000.

[13] K. Römer, "Time synchronization in ad hoc networks,"*ACM MobiHoc'01*, Long Beach, CA, Oct. 2001.

[14] R. T. Ma, V. Misra, and D. Rubenstein, "An analysis of generalized slotted-aloha protocols," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 936-949, Jun. 2009.

[15] Omitted Proofs online at http://ee.ucla.edu/∼jiexu/documents/GC-MAC-proofs.pdf.