Routing in the Soccer Field for Real-Time Athlete Monitoring

Ashay Dhamdhere†, Alex Kurusingal†, Sarthak Grover†, Vijay Sivaraman† and Alison Burdett⋆
†School of EE&T, University of New South Wales, Sydney, NSW 2052, Australia
Emails: {ashay@unsw.edu.au, alex@ee.unsw.edu.au, sarthak@ee.unsw.edu.au, vijay@unsw.edu.au}
⋆Toumaz Technology Limited, Abingdon, Oxfordshire, UK. Email: {alison.burdett@toumaz.com}

***Abstract*—Real-time physiological monitoring of athletes during sporting events has tremendous potential for maximizing player performance while preventing burn-out and injury, while also enabling exciting new applications such as referee-assist services and enhanced television broadcast. Emerging advanced monitoring devices have the right combination of light weight and unobtrusive size to allow truly non-intrusive monitoring during competition; however their small battery capacities (and consequently limited wireless ranges) make real-time data extraction a challenge, particularly in sports with a large playing area. We focus on real-time data extraction in a soccer game. In a companion paper we have profiled the wireless connectivity between players and shown that the network thus formed is highly dynamic, exhibiting sparse connectivity and correlation between links. To the best of our knowledge routing protocols for such a scenario have not been evaluated before. This paper makes three contributions. First, we show that multi-hop routing is necessary to extract data in real time, and that the optimal (though infeasible) delay is well within what is acceptable for real-time data delivery. Next we consider practical classes of routing protocols, and present routing protocols with successively better delay performance (at the expense of increased energy consumption), culminating in a novel tunable scheme that allows us to trade off these conflicting requirements. Finally, we show that suitable parameters exist which allow our proposed scheme to achieve the required combination of a high delivery ratio, low delays and acceptable resource consumption, enabling real-time athlete monitoring in the soccer field.**

I. INTRODUCTION

Sports today is becoming increasingly scientific, and elite athletes rely as much on technology as on training. With the consequent raising raising of the bar and narrower winning margins, maximizing performance while avoiding injury requires continuous physiological monitoring of athletes during both training and competition. Real-time information on athlete heart-rate, temperature, speed, distance, position and impact during a live game is tremendously useful and can be exploited in many ways. For example, it can help identify potentially dangerous situations such as when a player's heart rate or temperature become excessive, or when high fatigue levels affect body physiological and bio-mechanical processes to cause injury. Strategically, real-time information in field sports can assist the coach in optimising team performance by making more informed player substitution decisions, for example based on player vital signs or the distance run. Finally, there is a growing interest from sports organizers in using real-time position and impact information for referee-assist services in field sports such as soccer, rugby and hockey (akin to the third umpire in cricket), as also from television channels in augmenting live broadcast with player parameters (heart rate during clutch events, impact levels during collision, speed and acceleration, etc.) so as to heighten the level of engagement for audiences. The wide spectrum of applications and the promise of far-reaching benefits for athletes, coaches and the sporting audience has spurred tremendous interest in the field of athlete monitoring.

There are challenges to the deployment of athlete monitoring systems from both the device as well as the operating environment. From a device perspective the device size, battery capacity and communication range serve as impediments. Commercial devices in use today, such as the SPI Elite from GPSports [1] and CSIRO's radio tracking device have sizes resembling a mobile phone. These devices are clearly useful during training and in sports such as cycling where they do not impede athlete movement. However, they are too intrusive to be used in field sports such as soccer where athletes need to be totally unfettered; these sports need devices with small form factors and light weight allowing unintrusive monitoring and wireless relaying of data. In turn, the small form factors for the device necessitate small batteries, which correspondingly reduces the communication range, making real time data extraction challenging.

Further, the body-worn nature of the device means that the athlete's body acts as an attenuator, severely impeding the signal in certain directions. Correspondingly, the propagation of the signal shows strong dependence on the player's orientation. As a player's orientation in a soccer match is constantly changing, the radio connectivity between players exhibits a choppy, on-again, off-again pattern. The granularity of these changes distinguish the player monitoring scenario from other applications of wireless networks, resulting in significant challenges for any routing protocol which attempts real-time data delivery. To the best of our knowledge, there is no known feasibility study of multi-hop routing in such a highly dynamic network, nor any known data on the topology or connectivity in an athlete-monitoring network.

In a companion paper [2] we have extensively characterized the radio environment during soccer games. We gathered experimental data from a first-division University team, and profiled the wireless connectivity between players and from players to base stations around the field. We quantified the contact and inter-contact durations, and established for the first time the operating conditions in a field sports environment.

Further, we investigated the radio propagation characteristics of the body-worn device, and showed how the human body affects radio coverage at different orientations.

Our contributions in this paper are three-fold. First we establish the feasibility of real-time data extraction in this environment using off-line trace analysis, and evaluate the minimum delay achievable using *any* routing algorithm. We show that the delays achievable therein lie well within what is acceptable for real-time applications, and that multi-hop routing is essential to achieve acceptable delays. Next, we develop a class of practical routing algorithms which approach the performance of our offline optimal, with different consumptions of energy and transmission bandwidth, culminating in a novel tunable scheme that allows us to trade off resource consumption for delay performance. Finally, we show that suitable parameters exist which allow the proposed scheme to be tuned to achieve the required combination of high delivery ratios, low delays and acceptable resource consumption, enabling real-time athlete monitoring in the soccer field.

II. RELATED WORK

As we shall show, the topology and connectivity characteristics observed in the soccer game fall in the category of *delay tolerant networks*, wherein a connected path between a source-destination pair may not always exist [3]. The extensive research in this area can be broadly grouped in three categories based on the connectivity in the network. We outline existing work and point out the pitfalls in applying it to the unique athlete monitoring scenario.

A. Stable Routes

Protocols which fall into this class require that a stable route be present before forwarding of any message takes place. Such protocols attempt to forward a single copy of the message between a source-destination pair in the network and tend to assume that a continous end to end path will exist once the message has been forwarded. Examples of such protocols include AODV, DSDV [4] and DSR [5]. Under AODV, for example, a node that wants to send a message to a particular destination broadcasts a Route Request (RREQ) packet to its neighbours [6]. If a node receiving the RREQ packet is the destination, or knows a route to the destination, it sends a Route Reply packet back to the source, which contains the forward path to the destination.

Clearly, these schemes require routes to remain stable for some time in order to provide good throughput. It has been seen that DSDV does not perform well compared to other single copy schemes such as DSR and AODV [7], [8] in the presence of mobility. Das et al. [9] evaluate AODV and DSR under a Random Waypoint mobility model, with speed distributed between 0 and $20ms^{-1}$. They showed that high delivery ratios (above 95%) are possible when nodes pause at each waypoint for more than 100 seconds, while low delays necessitate pausing for more than 300 seconds. While the athlete speeds in a soccer game are up to a factor of 4 lower, we show in [2] that the soccer environment entails significantly higher mobility, making this class of protocols unsuitable.

B. Long Meeting Durations and Unstable Routes

Protocols in this class are designed for scenarios wherein a connected path between source and destination may not persist, but where each interaction between nodes is sufficiently long to allow a dialog to take place. These protocols do not wait for a complete route to be discovered. When two nodes meet, they exchange information regarding their *utility* in reaching other nodes. Forwarding decisions are made based on these utility metrics. A single-copy example is the Seek and Focus protocol [10], while multi-copy schemes include PROPHET (Probabilistic ROuting Protocol using History of Encounters and Transitivity) [11] and MAXPROP [12].

A unifying feature of this class of protocols is the exchange of utility metrics between nodes during every encounter. In MAXPROP and PROPHET, node i delivers a copy of a message to node j if node j has a higher utility for reaching the destination than node i. In Seek and Focus, a message is randomly forwarded until a custodian with a sufficiently high utility value for the destination is encountered. This custodian now performs utility-based routing, forwarding the packet only to nodes with even higher utility values for the destination.

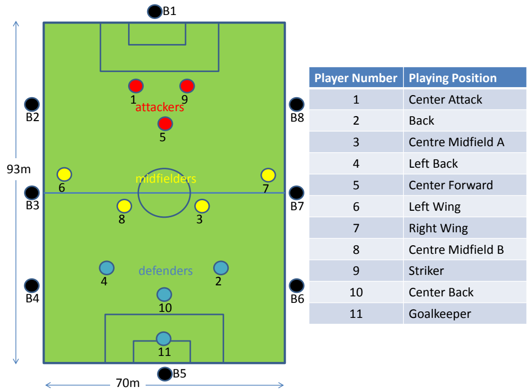
The exchange of utility information calls for encounters between nodes to be long enough for this handshake procedure to be carried out. These schemes are ideally suited for sparse networks (where a node may spend sustained periods being disconnected with other nodes) with long encounter times (allowing utility information to be exchanged, forwarding decisions to be made, and data exchanged before the nodes move out of contact). However, as we show in [2], the connectivity in soccer games often changes on a second-by-second basis, which is too rapid for this class of schemes.

C. Short Contact Duration

This class of protocols do not disseminate routing table updates or utility vectors, and are thus better suited to situations where the encounter durations between nodes are very short. Examples of this protocol include *direct transmission*, *Randomized Routing* and *Spray and Wait*. In direct transmission, for example, the node generating a message is the one delivering it to the destination. The delivery delay depends on the node mobility patterns, and bounds on delay can be established [10], [13], [14]. These protocols do not require a lengthy exchange of data between nodes during an encounter, and are well suited to the connectivity patterns experienced in a soccer match.

III. EXPERIMENTAL SETUP

As typifies an emerging application, very little is known about the wireless connectivity between players and from players to base stations in a soccer field. We experimentally characterized and quantified the connectivity in conjunction with a University football club. We outfitted each player in the squad with a MicaZ mote, which was programmed to send out a *hello* packet every second. This was done in a time-slotted manner, such that only one node was transmitting at any given time. All other nodes listened, and kept track of all nodes they heard during the experiment. Further, we located 8 base stations around the field, which also listened to player transmissions. During the course of the experiment, two of the player motes were lost (due to physical contact between players), and a base station was lost (as it was hit by the soccer ball), leaving 9 functional player motes and 7 base stations. We shall refer to the *aggregate base station* to encapsulate all bases; thus a player can reach the aggregate base if it can reach *any* of the bases. The experimental setup is illustrated in fig. 1.



Player Number	Playing Position
1	Center Attack
2	Back
3	Centre Midfield A
4	Left Back
5	Center Forward
6	Left Wing
7	Right Wing
8	Centre Midfield B
9	Striker
10	Center Back
11	Goalkeeper

Fig. 1. Experimental setup for characterizing wireless connectivity

Effectively, this data allowed us to construct, in an offline manner, the connectivity patterns between players (and from players to base stations) on a per-second basis. We found the following,

1) The encounter durations are typically very short, with 90% of the encounter durations being 4 seconds or less.
2) At each instant of time the topology is sparsely connected. Of the 81 possible unidirectional links (including 72 links between players and 9 links from each player to the aggregated base station), only a small fraction are on at a given time.
3) The inter-encounter durations exhibit a power-law distribution (as opposed to exponential) in the interval $10-100$ seconds, leading to heavy-tailed inter-encounter durations in this range. The long inter-meeting times between players are a fundamental limitation that routing protocols must contend with.

Our experiments, profiling and modeling of the connectivity are described in detail in a companion paper [2], and set the stage for the routing protocols to be presented herein.

IV. NEED FOR MULTI-HOP ROUTING

Based on our connectivity profiling, we seek to answer two questions. First, we seek to establish the delays achievable using the simplest routing strategy, direct delivery. The delays achievable herein would help us determine whether more sophisticated strategies are required. Next, we ask what would be the *lowest achievable* delay using any protocol. With this end, we implement off-line, an impractical (in terms of resource consumption), yet optimal (in terms of delay performance) flooding-based protocol. This allows us to bound the delays achievable by any practical scheme and determine whether real time data extraction is possible.

A. Unsuitability of Single-Hop Routing

The simplest single-copy forwarding scheme simply transmits a sample at every time instant until the sample reaches the base; since the player has no way of knowing whether a base is within listening range of its transmission, it continually transmits hoping to reach a base. We assume here and in all subsequent schemes that the base is able to inform a player when its transmission is received, and further, that transmissions take place every second. We refer to this strategy as Direct Delivery. Performance would depend heavily on player position; players which are normally located close to a base (such as the goalkeeper) will see good performance, as opposed to players located in the center of the field (such as a midfielder).

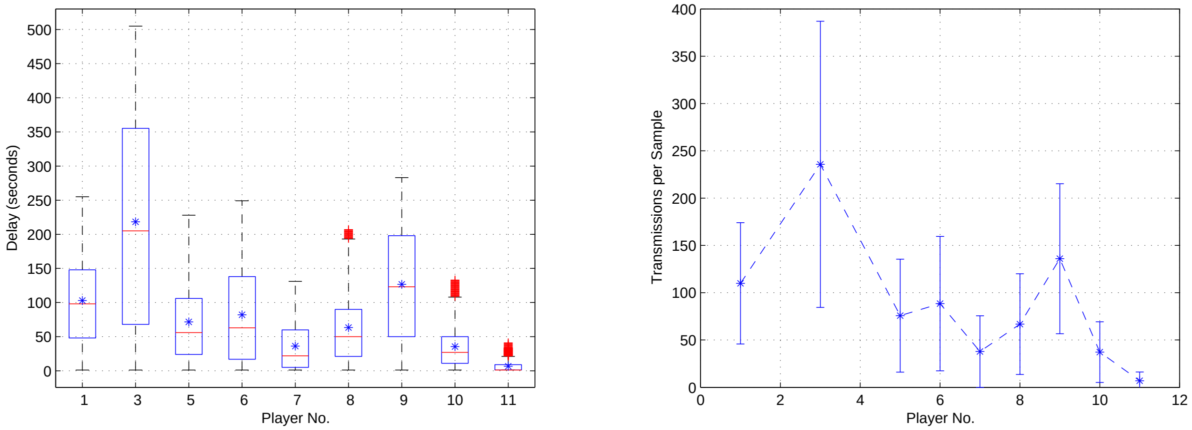
We evaluate the delay experienced by individual samples. Data samples which cannot be immediately delivered to the base are assumed to be buffered for later delivery. Thus the only packets which would remain undelivered are those generated towards the end of the game period. We assume that these samples are delivered to the base at the end of the game, and their delays are calculated accordingly.

Fig. 2 presents the delay performance (in the form of a boxplot, with the mean delay superimposed as a blue star) as well as the resource consumption for Direct Delivery. The scheme is relatively frugal in terms of resource consumption, with each sample being transmitted less than 100 times on average for a majority of the players. However the mean delays lie above 50 seconds for a majority of the players, and can reach as high as 200 seconds; only the goalkeeper, who is well connected with the base, has a small delay. Clearly, such high delays are unsuitable for real-time applications, thus establishing the necessity for multi-hop routing.

B. Off-line Optimal Routing

The offline optimal provides a measure of the best achievable delay. We assume that once a data sample is generated, it is continually transmitted in every slot. Similarly, when a neighbour picks up the sample, it is re-transmitted by the neighbour. The only provision we make to reduce the number of transmissions in the network is to assume that, once a player transmits to the base, the base is able to inform the player that the transmission was successfully received. The player's device then erases its entire window of samples, and the process re-starts. This results in a complete *flooding based* approach to data dissemination, which is intensive in terms of memory usage as well as bandwidth.

However, this approach is useful in benchmarking the lowest achievable delay and the highest achievable delivery ratio. Since every available node participates in the forwarding process, this mechanism is guaranteed to provide the lowest delay. Similarly, since all samples are transmitted until they



(a) Boxplot of Delay

(b) Mean Number of Transmissions per Sample

Fig. 2. Delay Performance and Resource Consumption of Direct Delivery

reach a base station, it provides every data sample with the highest possible chance of being delivered.

We plot the delay performance from time 900 to 1400 (in the form of a boxplot with the mean superimposed as a blue star), as well as the resource consumption (the mean number of transmissions per sample) in figure 3. Note that the mean delays are between 7 and 18 seconds, well within the range of what is acceptable for real-time applications. However, good delay performance comes at the cost of increased resource consumption; the mean number of transmissions per sample is above 2000 for all players except the goalkeeper, between 10 and 60 times higher than what is required for direct delivery. Though impractical, this scheme allows us to explore certain characteristics of the playing environment.

1) Effect of Player Position: The player positions have an important role in determining the nature of delay. For the goalkeeper (player 11), 48% of the packets are received immediately; during these times the goalkeeper is connected to the base station directly behind the goal. In comparison, only 8% of the midfielder's packets are delivered immediately, since the midfielder spends most of his time away from the boundaries of the field. The result is a larger median value for the delay (10 seconds as compared to 1 second for the goalkeeper). This is a fundamental limitation brought upon by the nature of player mobility as well as the radio characteristics of the devices, which any routing algorithm must contend with.

2) Effect of Intermittent Connectivity: We note that even for the goalkeeper (the best connected player in the network), the sample delays have a long tail that extends to more than 40 seconds. This is because even the goalkeeper is out of touch with the base station (located directly behind the goal) for significant periods of time, leading to periods where *none* of the players are able to connecte to a base. During this period, data gets flooded thru the network, awaiting a path to the base thru *any* player. This is a fundamental limitation on the delays brought upon by the nature of connectivity in field sports.

V. FORWARDING BASED ROUTING SCHEMES

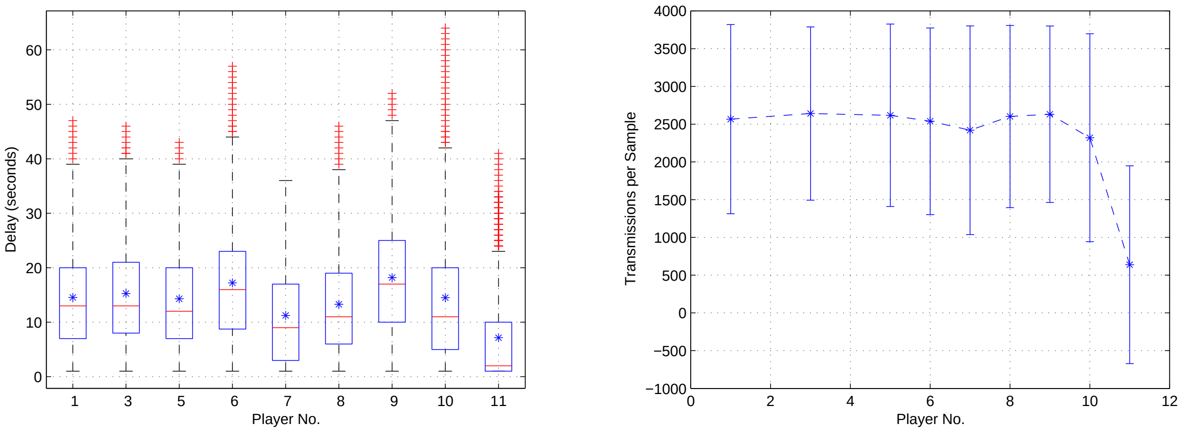
The routing scheme proposed in the previous section is too resource-intensive for practical implementation. We first consider schemes that are extremely frugal in terms of their resource consumption, of course, at the cost of increased delay. The uniting feature of these schemes is that they only allow a small number of copies of each data sample to be in circulation at any given time. By doing so we inherently limit the number of transmissions necessary to convey the data to the base.

A. Random Forwarding

One way of making use of player mobility is to probabilistically forward a data sample to other randomly encountered players. For players who see the base relatively infrequently this has the desirable effect of disseminating data to players who may have more frequent interactions with the base. If a player cannot deliver data to the base in a given time slot, it explores whether other players are able to hear its transmissions. If it is able to hear n other players, it tosses a fair $(n+1)$-faced die. With probability $1/(n+1)$ it keeps the packet to itself, while the probability of forwarding to any of its neighbours is $1/(n+1)$ each. If the player has no neighbours in a given slot, it keeps the data sample to itself.

In this way, the *custodian* of the data sample is chosen at every time step. The above procedure is now repeated by each custodian of the sample, until the sample hits the base station. This simple forwarding scheme results in a *decrease* in the mean delay for sparsely connected players, as it affords them a chance to make use of better connected players. In [10] the authors show that, for mobility models where the expected meeting time is a concave function of distance, randomized forwarding results in a reduction of the expected delivery time of the message. The assumptions for this result call for every node to move *independently* according to a mobility model which has this property; this does not happen in a soccer game, which exhibits correlation between player movements [2]. However, for players who hold more central positions in the field, random forwarding results in the sample reaching more peripherally located players, who have smaller expected times to hit the base.

The delay performance and resource consumption of Random Forwarding is illustrated in fig. 4. The players who suffer in this scheme are those that have a very good connection



(a) Boxplot of Delay

(b) Mean Number of Transmissions per Sample

Fig. 3. Delay Performance and Resource Consumption of Full Flooding

with the base station, e.g. players 7, 10 and 11 (right wing, center back and goalkeeper), which are located peripherally and close to a base. Random forwarding for these players results in samples traveling inwards into the center of the field; e.g. the sample sent at time 1168 by the goalkeeper traverses the path $11 \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow 8 \rightarrow 6 \rightarrow 11$, *returning* to the goalkeeper to be finally forwarded to the base, incurring a delay of 81 seconds. The direct delivery delay on the other hand is only 20 seconds.

B. Two-copy Routing Scheme

In order to make the Random Forwarding scheme amenable for players who have frequent contact with the base, we introduce a second copy of each sample that is always kept with the source. The first copy is randomly forwarded exactly as in the single copy case.

The delay for each sample through this scheme is effectively the minimum of the delays incurred by direct delivery and random forwarding. Clearly, this scheme will result in more transmissions per sample, while improving performance for well-connected players. We plot the delay performance as well as the resource consumption in fig. 5. As we would expect, the right wing, center back and goalkeeper all see their performance improve, while other players continue to experience the benefits of random forwarding.

C. Comparison of Forwarding Based Schemes

In fig. 6 we compare direct delivery, random forwarding and two-copy using the offline optimal as a reference. Note that there is still a significant gap in delay performance between the best-performing scheme (two-copy) and the offline optimal. While the frugal resource consumption is desirable, we require better delay performance for real-time applications. In the following section we consider flooding-based approaches to bridge this gap.

VI. TUNABLE FLOODING SCHEME FOR SOCCER

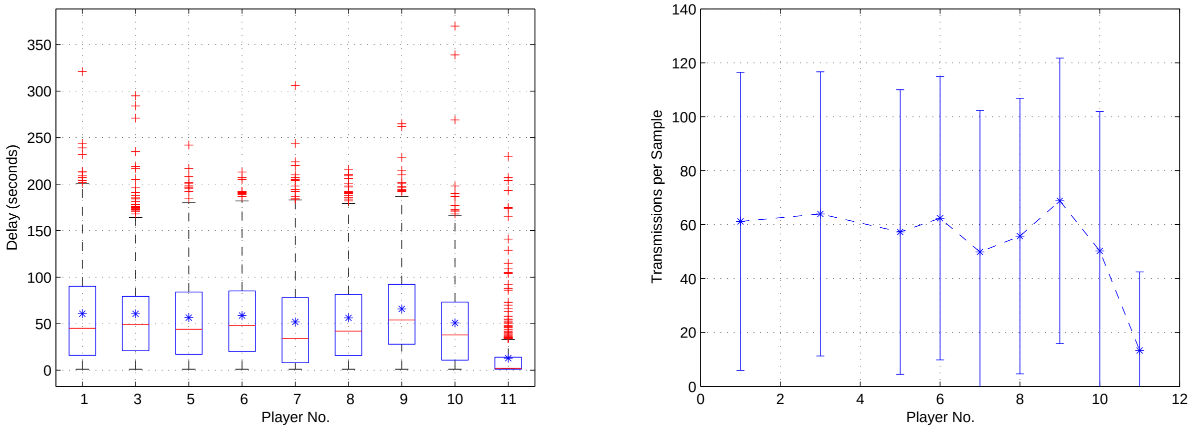
The schemes we have considered so far have fixed tradeoffs points between delay performance and resource consumption. The offline optimal is biased completely towards delay performance, resulting in very high resource consumption. On the other hand, random forwarding and direct delivery are biased towards low resource consumption, resulting in delays that are up to 16 times higher than the offline optimal. From an application viewpoint, we would like to be able to *tune* the performance of the scheme, either towards better delay performance or lower resource consumption. To allow for good delay performance, we consider a flooding-based approach. A practical flooding-based strategy must be able to limit both the memory requirements as well as the transmission bandwidth. The tunable scheme we propose allows us to control these parameters, and incorporates the following features.

A. Features of Proposed Scheme

1) Replication at the Source: In the proposed scheme, a player maintains a *window* of samples of length W for itself in a FIFO manner; thus, a newly generated sample pushes out the oldest sample in the window. During its slot, a player transmits its entire window of samples. Thus, every sample is *replicated* W times by the source. As in the offline optimal algorithm, this guards against the case where a sample may be lost because no other player (or base station) is able to hear the transmitter at that time.

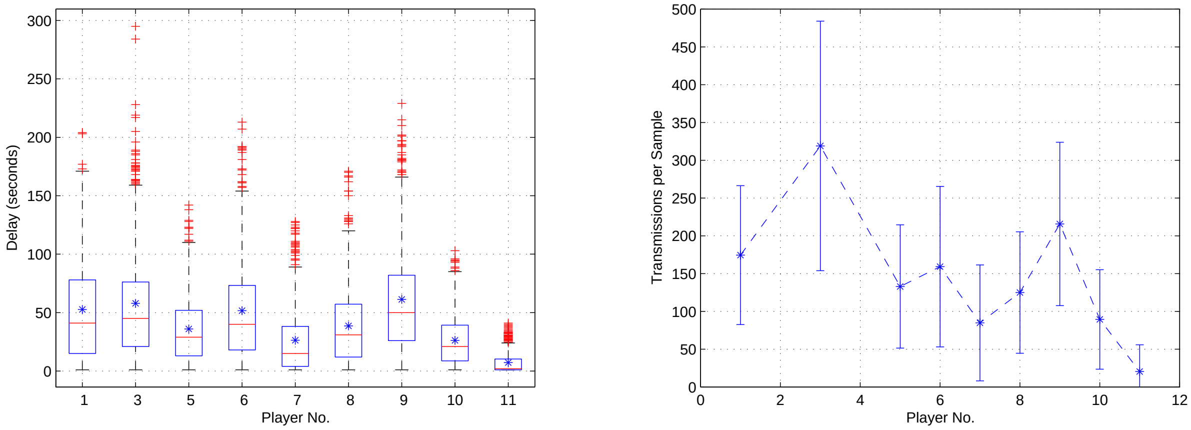
2) Replication at Intermediate Nodes: Further, every player also maintains a window of W samples for *every other player*. In effect, this allows replication of the samples at *every intermediate hop*. When player i is not transmitting, it listens to other players' transmissions. Packets heard from other players are then analyzed for data. If a received packet contains a more recent window of samples for player j, this data replaces the contents of the window for player j held by player i.

3) Data Freshness: The above two steps limit the amount of data transmitted in any packet to a $N * W$ matrix of samples (where N is the number of players). However, we still have the issue of old data being forwarded within the network. This may arise, for instance, when player i forwards its data to player j, and then becomes disconnected from other players for an ensuing interval. In this case, the data from player i



(a) Boxplot of Delay (b) Mean Number of Transmissions per Sample

Fig. 4. Delay Performance and Resource Consumption of Random Forwarding



(a) Boxplot of Delay (b) Mean Number of Transmissions per Sample

Fig. 5. Delay Performance and Resource Consumption of Two Copy

may become obsolete (from an application point of view), yet continue to be forwarded by player j and its neighbours.

To ensure *data freshness*, player j forwards data for player i only if the *most recent* sample in its window for player i is less than A samples old. Thus, each intermediate node filters the data it receives from surrounding players before forwarding it on. The quantity A can be selected based on the requirements of the application. For instance, for television broadcast enhancement where players' heart rate is streamed in real time, samples which are more than 10 seconds old may become irrelevant, and A can be set to 10.

B. Behaviour of Proposed Scheme

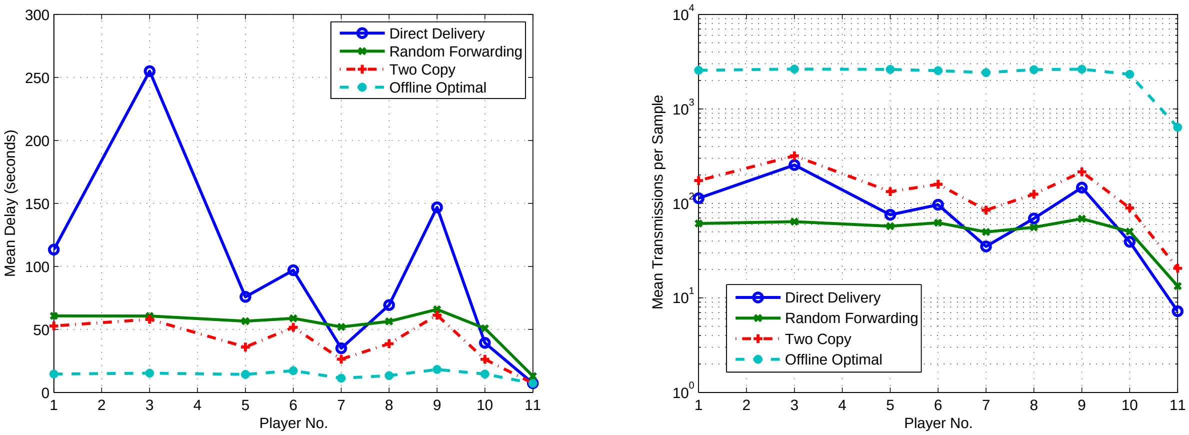
As opposed to the forwarding-based schemes, our flooding-based scheme will actively *drop* a window of samples as they age beyond A. As a result, certain samples may never get delivered to the base. Therefore, as compared to schemes so far, we need to evaluate an additional parameter, the *delivery ratio* of this scheme.

Further, note that W controls the amount of replication at the source, while A controls the amount of replication in the network. A smaller value of A means that a given window of samples is likely forwarded through a smaller number of hops, as it would hit the age threshold faster. Hence small values of A result in lower delivery ratios (for players who are not well connected with the base station) as the samples are dropped within the network before they hit the base.

Finally, note that the *maximum* time a sample can spend in the network is $W + A$. Note that we have chosen to implement our algorithm in this form (rather than take a more traditional approach in terms of hop counts) for two reasons:

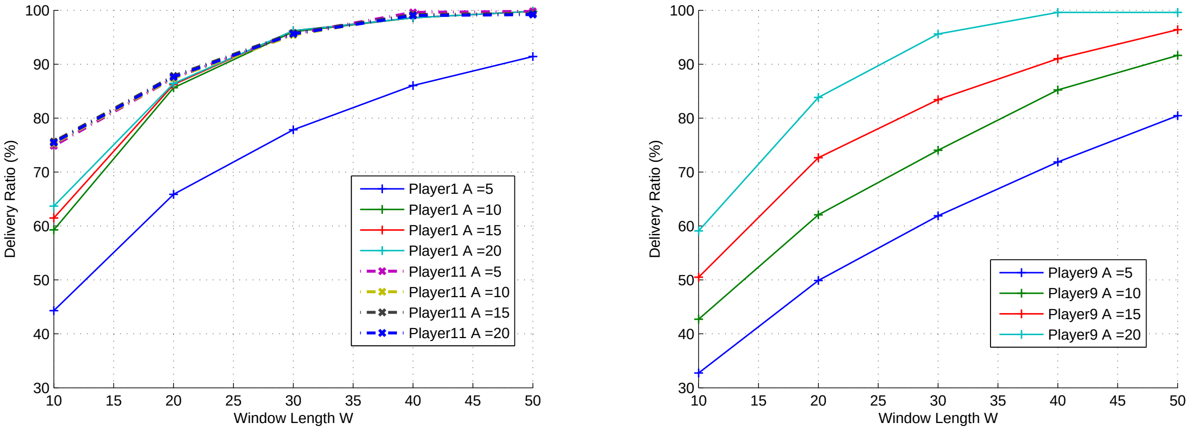
1) Using a window of samples for each player allows us to naturally control another important parameter, the *memory requirements* of this scheme, to a buffer of at most $N \times W$ samples per player.
2) Given that every player maintains a buffer of W samples for every other player, it is artificial to impose a hop count on every sample in that buffer. Rather, it is simpler and more intuitive to impose a maximum age A on the buffer, and to discard the entire buffer if the data is too old.

Finally, as with other schemes we assume that the base is able to inform a player when its transmission is received, upon which the player erases its entire window of samples.



(a) Mean Delays of Forwarding-based Schemes (b) Resource Consumption of Forwarding-based Schemes

Fig. 6. Comparing Delay Performance and Resource Consumption of Forwarding-based Schemes



(a) Player 1 (Center Attack) and Player 11 (Goalkeeper) (b) Player 9 (Striker)

Fig. 7. Delivery Ratios for Center Attack, Goalkeeper and Striker as a function of W and A for Tunable Scheme

C. Characterizing Dependence on Parameters

We evaluate the delivery ratio, delay performance and energy consumption of our tunable scheme. The results are best understood by categorizing players into three classes: a) those that are well-connected with the base (e.g. the Goalkeeper), b) those that see the base infrequently but are well-connected with other players (e.g. the Center Attack), c) those that see both the base as well as other players infrequently, (e.g. the Striker).

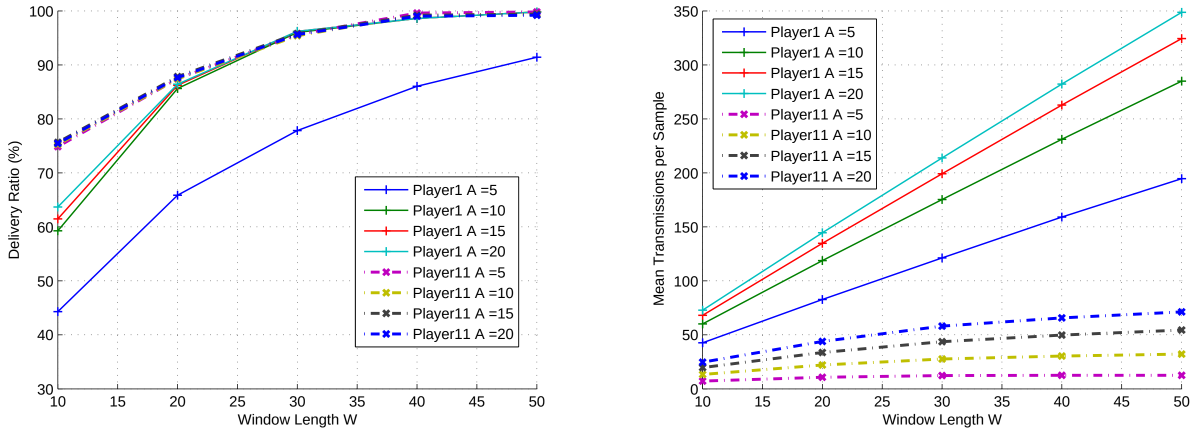
We plot the delivery ratios for representatives of each of the above three categories (the Goalkeeper, Center Attack and Striker) in fig. 7, and the mean delays and resource consumption for the Center Attack and Goalkeeper in fig. 8 (a) and (b) respectively (the delay performance of the Striker is found to be qualitatively similar to that of the Center Attack for the latter two parameters), and analyze each class of players.

Well-Connected to the Base: These players (e.g. Goalkeeper) need very little support from the network. Thus their delivery ratio is insensitive to A, and depends only on W. Similarly, their delay performance is insensitive to A since they delivers most of their packets directly to the base. Further, their resource consumption saturates with increasing W (as opposed to other classes where the resource consumption increases roughly linearly); since this class of players deliver most samples directly to the base, (and players flush their buffers on reaching the base), samples are typically not transmitted for the entire duration of the window.

Well-Connected with Other Players: These players need a certain minimum level of network support to deliver their packets. Thus, delivery ratio is poor for $A = 5$, but becomes insensitive to A for $A \geq 10$. Similarly, at small A values the burden of deliver falls on the source. Thus, we see higher mean delays for small A (occuring at higher values of W), while the mean delay decreases for higher A values (even as the delivery ratio increases). This shows that, as the forwarding load is balanced between the source and the network, both delivery ratio and mean delay can be improved.

The resource consumption for this class of players increases roughly linearly with W, with a slope determined by A (A determines the degree to which a sample sent by the source is forwarded in the network, which defines the slope).

Infrequently Connected with Base and Players: Players in



(a) Mean Delay (b) Resource Consumption)

Fig. 8. Mean Delay and Resource Consumption of Tunable Scheme as a function of W and A

this category (e.g. Striker) show similar characteristics as the previous class, except in their delivery ratio. Due to their isolation, they require substantial support from the network, and their delivery ratio rises steadily with A.

D. Tuning the Parameters

In the previous section we varied W and A independently. We now ask how we should divide our resources between W (the source) and A (the network). We plot the delivery ratio, mean delay and resource consumption for our tunable scheme for different W and A values, with $W + A = 45$, in figs. 9, 10 and 11.

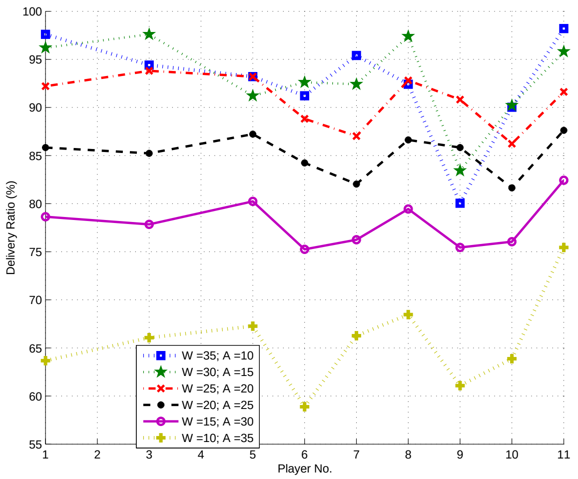


Fig. 9. Delivery performance of Tunable scheme with $W + A = 45$

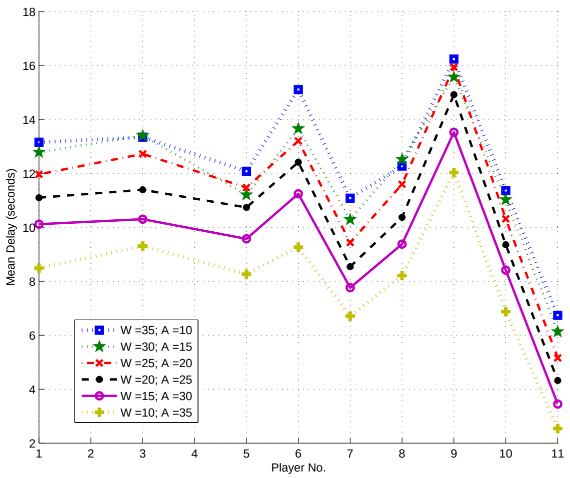


Fig. 10. Delay performance of Tunable scheme with $W + A = 45$

We note the following,

1) When low resource consumption is required, our scheme can be tuned to small W values (e.g. $W = 10, 15$); this limits the source transmissions which inherently limits resource consumption. Of course, this comes at the cost of a lower delivery ratio.
2) When higher resource consumption is accpetable, our scheme can be tuned towards higher W ($W = 25, 30$); the resulting increase in source transmissions (coupled with sufficient network support) increases the delivery ratio at the expense of higher resource consumption.
3) As before, choosing a very high value of W ($W = 35$) puts the burden of delivery on the source and is counter-productive

VII. DISCUSSION

The performance of all schemes is compared in figs. 12 and 13. The proposed tunable flooding-based scheme allows us to control the level of resource consumption to achieve desired delivery ratios. We see that at high delivery ratios ($> 90\%$) achieved with $[W = 30, A = 15]$, the scheme is able to achieve delay performance that is very close to that of the offline optimal, while keeping resource consumption within a factor of 4 from that of forwarding-based schemes; we note that the resource consumption is only a tenth of what is required by the offline optimal. The reader may be wondering why the delay performance of our scheme is better than the optimal; it must be noted that the optimal delivers *all* packets, while our scheme does not.

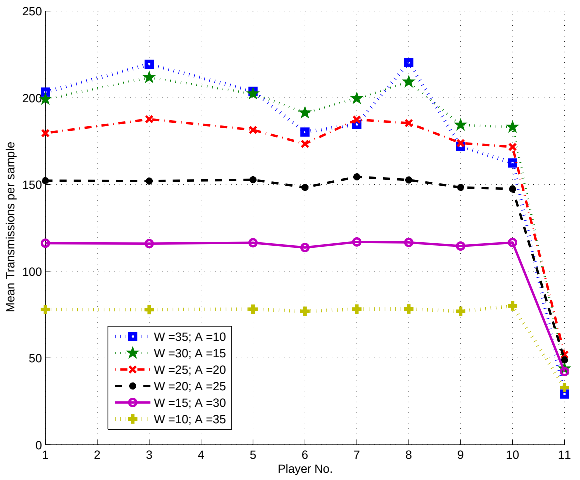


Fig. 11. Resource consumption of Tunable scheme with $W + A = 45$

On the other hand, when a high delivery ratio is *not* required, our scheme is tunable to reduce resource consumption even further.

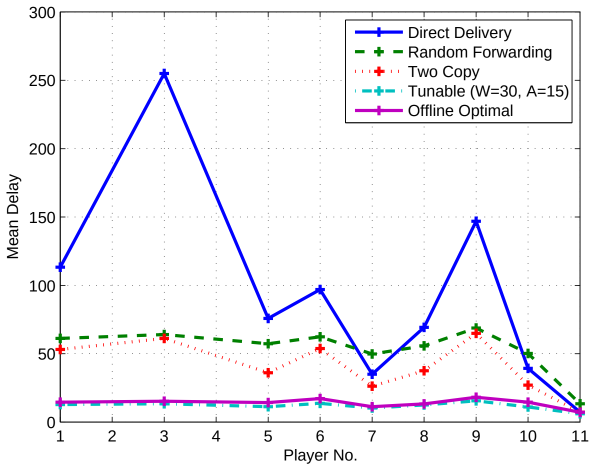


Fig. 12. Comparing Delay Performance of Different Schemes

VIII. CONCLUSION

We consider the problem of real-time monitoring of soccer players. In a companion paper, we have characterized the wireless connectivity between players through experimental data collection, modeling and analysis. In this work we explore routing strategies that yield good performance for real-time applications. We first show that multi-hop routing is essential to achieve good delay performance, and show that the delay achieved by an optimal (though impractical) routing algorithm is well within acceptable limits. Next, we show that forwarding based algorithms cannot meet delay requirements, and propose a tunable flooding-based scheme. We show that through the right choice of parameters, our tunable algorithm achieves a balance between high delivery ratios, low delays and low resource consumption. Further, the tunable nature of our scheme allows it to be used for a wide range of applications.

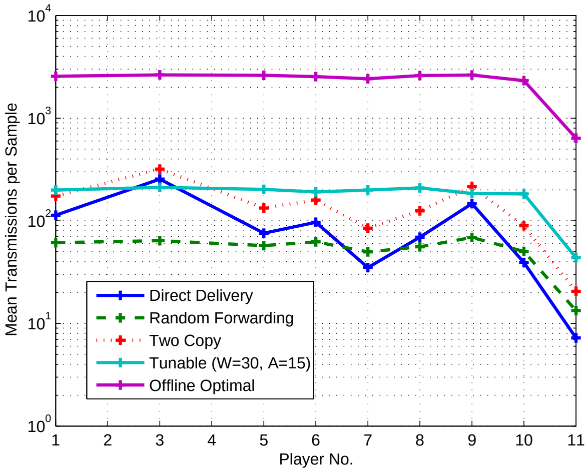


Fig. 13. Comparing Resource Consumption of Different Schemes

REFERENCES

[1] GP-Sports, “Spi elite.” [Online]. Available: http://gpsports.com

[2] V. Sivaraman, S. Grover, A. Kurusingal, A. Dhamdhere, D. Ostry, and A. Burdett, “Mobility in a soccer field: From empicical data collection to modeling correlated connectivity,” in *submitted to INFOCOM 2010*, March 2010.

[3] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *SIGCOMM ’04*. New York, NY, USA: ACM, 2004, pp. 145–158.

[4] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers,” *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 234–244, 1994.

[5] D. B. Johnson, D. A. Maltz, and J. Broch, *DSR: the dynamic source routing protocol for multihop wireless ad hoc networks*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, pp. 139–172.

[6] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, “Rfc 3561 - ad hoc on-demand distance vector (aodv) routing,” INTERNET ENGINEERING TASK FORCE, Jul 2003. [Online]. Available: http://tools.ietf.org/rfcmarkup?doc=3561

[7] F. Bai, N. Sadagopan, and A. Helmy, “Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks,” in *INFOCOM 2003*, 2003.

[8] T. Liu and K. Liu, “Improvements on dsdv in mobile ad hoc networks,” in *WiCom 2007*, Sept. 2007, pp. 1637–1640.

[9] S. R. Das, C. E. Perkins, and E. E. Royer, “Performance comparison of two on-demand routing protocols for ad hoc networks,” in *INFOCOM 2000*, 2000, pp. 3–12.

[10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: the single-copy case,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, 2008.

[11] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, 2003.

[12] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption-tolerant networks,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, pp. 1–11.

[13] M. Grossglauser and D. Tse, “Mobility increases the capacity of ad hoc wireless networks,” *Networking, IEEE/ACM Transactions on*, vol. 10, no. 4, pp. 477–486, Aug 2002.

[14] R. C. Shah, S. Roy, S. Jain, and W. Brunette, “Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks,” *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215 – 233, 2003, sensor Network Protocols and Applications.