Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing

Alberto Cerpa and Jennifer L. Wong and Miodrag Potkonjak and Deborah Estrin
Center for Embedded Networked Sensing, University of California, Los Angeles

January 2005

Abstract

Recently, several studies have analyzed the statistical properties of low power wireless links in real environments, clearly demonstrating the differences between experimentally observed communication properties and widely used simulation models. However, most of these studies have not performed in depth analysis of the temporal properties of wireless links. These properties have high impact on the performance of routing algorithms.

Our first goal is to study the statistical temporal properties of links in low power wireless communications. We study short term temporal issues, like lagged autocorrelation of individual links, lagged correlation of reverse links, and consecutive same path links. We also study long term temporal aspects, gaining insight on the length of time the channel needs to be measured and how often we should update our models.

Our second objective is to explore how statistical temporal properties impact routing protocols. We studied one-to-one routing schemes and developed new routing algorithms that consider autocorrelation, and reverse link and consecutive same path link lagged correlations. We have developed two new routing algorithms for the cost link model: (i) a generalized Dijkstra algorithm with centralized execution, and (ii)a localized distributed probabilistic algorithm.

1 Introduction

Recent studies indicate profound differences between experimentally observed properties of low power communication links and widely used simulation models [1, 2, 3, 4, 5]. Nevertheless, most of these studies have not performed in depth analysis of the temporal properties of wireless links. These properties have a strong impact on the performance of many protocols and localized algorithms used in low power networks, in particular, routing algorithms.

Our starting point is a study of statistical temporal properties of links in low power wireless communication systems. We emphasize on time dependent properties, which have strong ramifications on routing protocols. The results of the study are used to analyze how statistical temporal properties impact routing protocols. We studied one-to-one routing protocols and provided several suggestions for protocol designers using the insight gained from our analysis. We have also developed new routing algorithms that consider autocorrelation, reverse link and consecutive same path link lagged correlations. The first algorithm is a generalized Dijkstra algorithm with centralized execution. The second algorithm is a localized probabilistic algorithm with distributed execution.

In our study, we do not consider packet losses introduced by traffic synchronization (concurrent traffic, contention based MAC). Nevertheless, our results are useful for two reasons. First, they apply directly when using contention free MAC protocols, like pure TDMA or pseudo-TDMA schemes [6]. Second, they provide a *tight* upper bound as to what is achievable when using contention-based MAC schemes.

2 Related Work

There is a large body of literature on temporal models of radio propagation that have influenced this work. The emphasis has been on the variability of signal strength in proximity to a particular location [7]. Small scale fading models based on Rayleigh and Rice distributions are used for modeling localized time durations (a few microseconds) and space locations (usually one meter) changes [7].

The differences between the classical models and our approach are numerous and include different modeling objectives (reception rate of packets vs. signal strength), our radios have different features (e.g. communication range in meters instead of km),

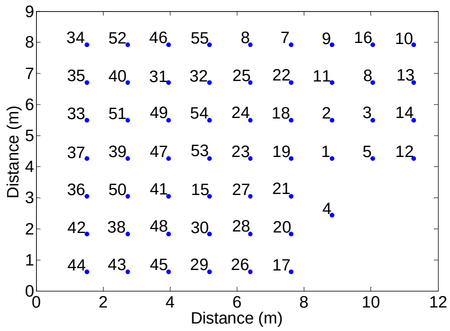


Figure 1: Layout of the nodes

we capture phenomena that is not addressed by the classical channel models (asymmetry, correlations between reception rate of links), we use different modeling techniques (free of assumptions, non-parametric vs. parametric), and we use unique evaluation techniques (evaluation of multi-hop routing).

More recently there have been many empirical studies with deployments in several environments using low-power RF radios [1, 2, 3, 4, 5]. The majority of these studies used the TR1000 [8] and CC1100 [9] low power RF transceivers (used by the Mica 1 [10] and Mica 2 [11] motes respectively). However, most of these studies concentrate on analyzing the spatial characteristics of the radio channel and do not analyze the temporal variability of link quality over extended periods of time. Zhao et al. [2] performed some temporal analysis using an array of nodes placed in a straight line with two hour experiments. They demonstrated heavy variability in packet reception rate for a wide range of distances between a transmitter and receiver. Furthermore, Cerpa et al. [4] used heterogeneous hardware platforms consisting of Mica 1 and Mica 2 motes in three different environments to collect comprehensive data about the dependency of reception rates over time with respect to a variety of parameters.

There are also several important empirical studies using medium-power RF radios in indoor and outdoor environments [12, 13]. These studies use 802.11 wireless radios. Aguayo et al. [12] measured the temporal variability of the packet reception rate for very short time scales (from 10 ms to 10 seconds) showing that the packet loss rate for all links could be considered independent for very small time scales (10 to 100 ms), and then it begins deviating for some percentage of links at larger time scales. Draves et al. [13] compared different link quality estimation metrics, showing that the ETX metric [14] performs significantly better than hop-count and packet pair in static wireless environments.

There are three major differences between the analysis and evaluation developed in this paper and the previous work. The first is that we study the impact of a significantly large number of factors that impact the quality of wireless links over time and attempt to model not only isolated pairs of transmitters and receivers, but also the correlation between different pairs and different subsets of links with significantly larger time scales (several days). The second major difference is that we have developed a new link quality metric that more accurately measures the impact of the temporal variations of the wireless channels. Finally, using the knowledge built with our analysis, we implemented two new routing algorithms that take advantage of our findings.

All of our techniques use non-parametric procedures. In particular, we directly leverage on smoothing and density kernel estimators [15, 16, 17].

3 Experimental Methodology

We performed experiments using the SCALE wireless measuring tool [4]. The basic data collection experiments work as follows. Either a single designated node or a group of nodes transmit a certain number of packet probes (one transmitter at a time in the case of multiple transmitters). Each probe packet contains the sender's node id and a sequence number. The rest of the nodes record the packets received from each neighbor and keep updated connectivity statistics, using the sequence numbers to detect packet losses.

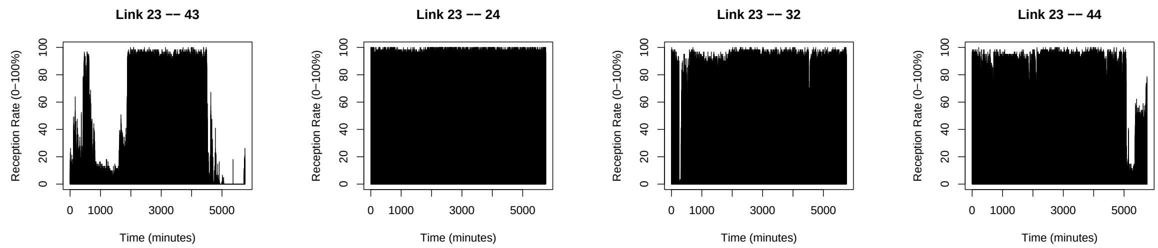
All experiments were conducted in an indoor office-like setting of approximately 20m by 20m. A total of 55 Mica 1 mote [10] nodes, which uses the RFM TR1000 radio chip [8], were placed in a grid structure in the environment at approximately 1m distances. The layout of the nodes is shown in Figure 1.

We collected four types of data sets:

Data set A. A single node broadcasts a packet every second, all other nodes record the received packets for a period of 24 hours. The purpose of this data set was to establish the behavior of links over an extended period of time. Four different nodes (10, 23, 44, and 54) were selected as the broadcasting node.

Data set B. Node 23 broadcasts a packet every second, all other nodes record received packets for a period of 96 hours. This data set is used to determine if there is cyclical patterns in the link quality over multiple days.

Data set C. Node 23 broadcasts a packet each second to all other nodes in the network for a period of 30 hours, where the packet size rotates in 10, 15, 20, 30, 40, 80, 120, 135, 145, 150, 155, 170 and 190 bytes. A packet of each size is sent every 13 seconds. The set is used to determine the impact of the packet



(a) RR: 48.02% RNP: 1189.6

(b) RR: 95.36% RNP: 1.0491

(c) RR: 86.39% RNP: 4.5673

(d) RR: 79.86% RNP: 1.9175

Figure 2: Aggregate of reception rate by minute.

size on link quality.

Data set D. All nodes broadcast a packet one at a time in round robin fashion with one second intervals between nodes for 48 hours. A packet is broadcasted by each node every 55 seconds. We collected two of these traces, the first one with round robin using node id sequence of $1, 2, \ldots 55$, and the second one with a random node id sequence.

4 Temporal Properties

In this section we analyze several individual and group link properties of wireless links in our experiments. Our goal is to improve our qualitative and quantitative understanding of the link temporal properties, providing intuition for network design and operation, as well as statistically sound conclusions.

4.1 Single Link Autocorrelation

The most common measure for the quality of links is the percentage of received packets over a certain period of time, *reception rate (RR)*. We will see that a better measure is to consider the average number of packets that must be sent before a packet is received. We will refer to this value as the *required number of packets (RNP)*. Commonly, it is assumed there is a reverse relationship between RNP and RR. However, temporal correlations often invalidates this.

For example, consider the four links shown in Figure 2. In this case, we show the aggregated reception rate (by minute) of the data from set B. In Figure 2(a) we see a link with an average reception rate of 48.02%. This link is highly unreliable and the required number of packets (with constant back-off) will be high (1189.65), even though there are minutes where the link is reliable. In Figure 2(b)) we show a link that has very high reception rate (95.36%). In this case, while a few messages were not received the link was completely reliable with very low required number of packets (1.05). Consider the medium quality links shown in Figures 2(c) and 2(d). The first link has a RR of 86.39% and the second link has a RR of 79.86%. When using RR as a quality metric, clearly the first link is better than the second. Surprisingly, when using RNP, the second link is much better than the first. The reason of this counterintuitive result is due to the fact that the RR metric does not take into account the *underlying distribution of the losses*; short periods of zero RR in any particular time interval will trigger the RNP to higher values, even though the average RR might still be higher than the other link in the same time interval. As a result of this inconsistent behavior, the required number of packets provides a better picture of the usefulness of the link.

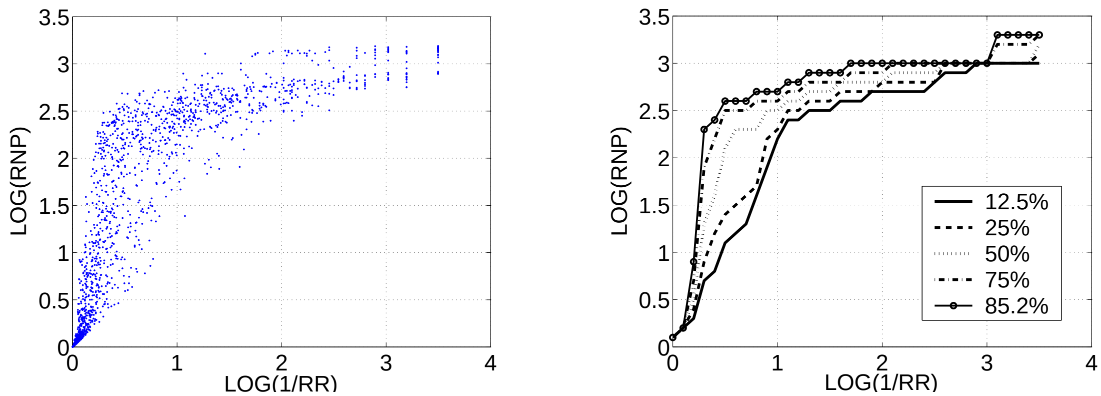
RR^{-1}	1-1.1	1.1-1.2	1.2-1.5	1.5-2	2-5	5-10
Cons.	0.970	0.695	0.661	0.658	0.607	0.555

Table 1: 1/RR and RNP Consistency

We statistically analyze the relationship between the reception rate and the required number of packets using data sets B and D to fully characterize the usefulness of moderate links. Fig. 3 shows the relationship between RNP and RR in log scale. If the underlying distribution of packet losses corresponds to a random uniform distribution, we would expect a one-to-one relationship between RNP and RR. From the figure we clearly see this is not the case. Perhaps more importantly, using RR as the main evaluation of link quality estimation can lead to gross overestimation of the quality of certain links. It is clear that there exist many pairs of links i and j where one link i has both lower RR and lower RNP than link j.

Fig. 3(b) shows the CDF of RNP as a function of RR for different percentages of population below each curve. We observe that most of the values tend to converge at very high or very low RR values, but the values of 10% < RR < 90% tend to be quite spread.

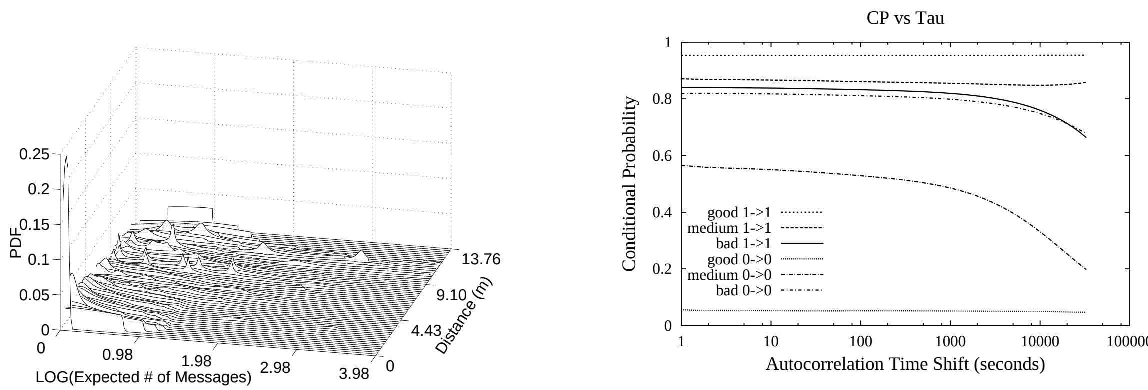
These figures show that RR is not a precise estimator for absolute RNP values in a significant range.



(a) Links characterized by their 1/RR and RNP values

(b) CDF of RNP as function of 1/RR (double logarithmic scale)

Figure 3: RNP as a function of 1/RR



(a) RNP and link length PDF

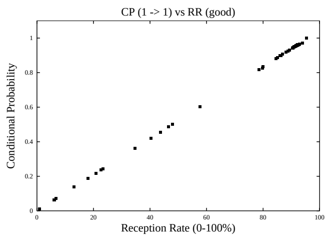
(b) CP as a function of τ

Figure 4: RNP and Autocorrelation

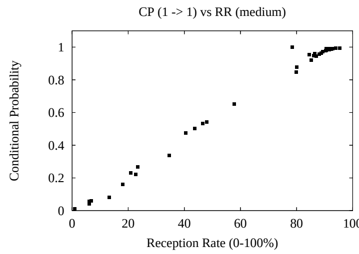
Table 1 shows the *consistency* levels based on different 1/RR value groups. In our study, our criteria of consistency is defined as the probability of the 1/RR estimator to rank the links in the same ranking order as their RNP values within each group, or in other words, the relative difference with respect to other links (not the absolute value). The consistency for all links is 0.959. However, from the table we observe that the RR measure is only a consistent RNP estimator for the higher quality link (RR $>$ 90%), and the consistency decreases rapidly after that. Note that a 50% consistency estimator is as useful as flipping a coin.

The likelihood of the required number of packets for a given distance is presented in Figure 4(a). We used data sets A and B for analysis. The figure illustrates that when the distance between the transmitter and receiver is low, there is a high likelihood that the RNP will be low. However, as distance increases, we do not see a corresponding increasing trend in RNP as would be expected if there exists a strong relationship between the two factors. Often for any given distance there is a non-trivial likelihood that the RNP will be fairly low. As a result, the distance between two radios is not the determining factor in the quality of the link.

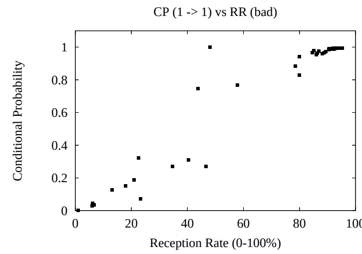
Next, we analyze the autocorrelation of three qualitatively different types of links (good, medium, bad) by using conditional probabilities (CP) of two events, a packet reception after a packet reception (1→1) and a packet loss after a packet loss (0→0) for different time intervals ($\tau : 2^n$ seconds with $n : 1, 2, \ldots 15$). We observe in Figure 4(b) that good links tend to have a very high and very low CP for the 1→1 and 0→0, respectively over very long periods of time. This indicates that good links are quite stable. On the other hand, bad links are stable as well for shorter periods of times, but their properties tend to disappear once



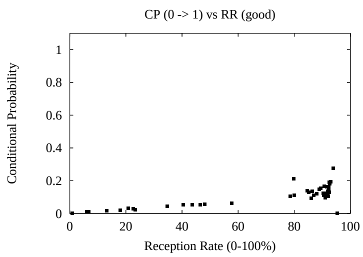
(a) good



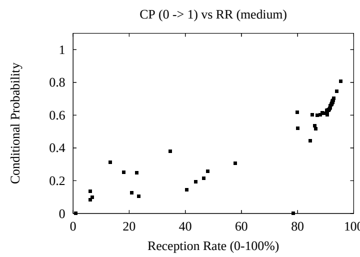
(b) medium



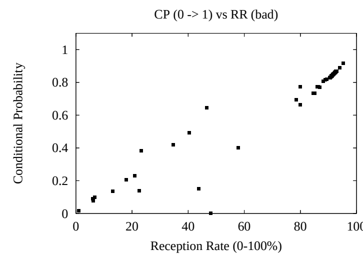
(c) bad



(d) good



(e) medium



(f) bad

Figure 6: Conditional Probability (1→1 and 0→1) as a function of Reception Rate

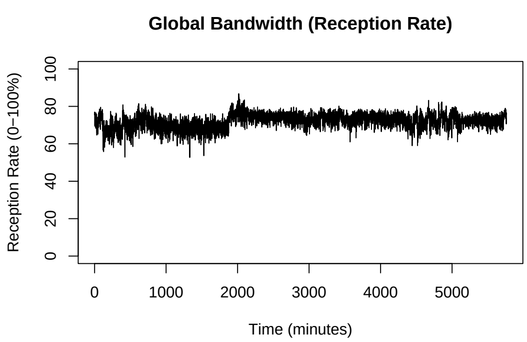


Figure 5: RR as a function of time

we go into longer periods of time (in the order of ∼9 hrs). Medium links tend to have higher correlation for successful packet reception (1→1) than for packet losses (0→0). In the latter case, the autocorrelation drops for the larger time intervals.

There are several conclusions we can draw from our analysis. First, the RNP metric is a better metric than RR to estimate the quality of the links because it takes into account the underlying distribution of losses. Second, RNP is not directly proportional to 1/RR. Third, similarly to RR, RNP is not correlated with distance. Finally, it is useful to initially measure the channel aggressively using a constant backoff strategy in a learning phase to fully characterize all the links, and then later switch to a very sparse sampling strategy.

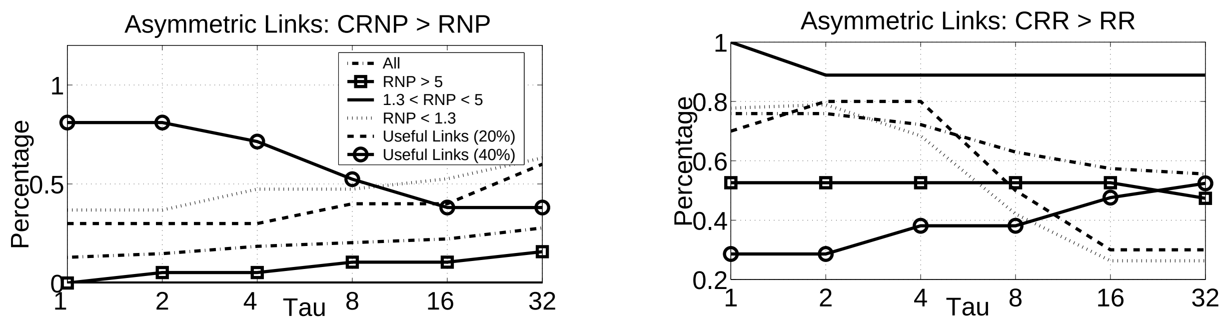
4.2 Global Quality Bandwidth

It is interesting to try to identify the existence of a particular time during the day where the average quality of all links from a transmitter is significantly better. The confirmation of the existence of this special time would affect the design of algorithms that could detect and opportunistically make use of it by transferring large amounts of data during that time interval.

We used data set B for this analysis. Figure 5 shows the average reception rate of all links in the system at the granularity of minutes. We can conclude that the global quality bandwidth of the system has very little oscillations over time and for all practical purposes it remains constant. There is no magic moment where we could opportunistically try to send more data to maximize chances of being delivered.

4.3 Covariance of Same Source Links

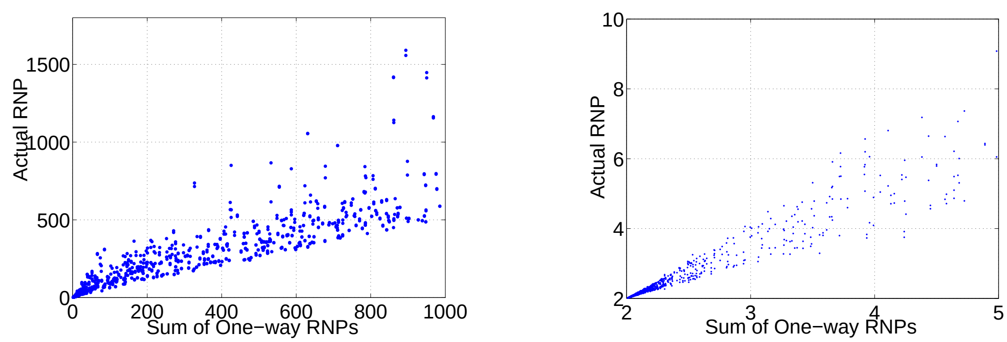
We also study properties of two links starting from the same source with the goal to verify whether routing strategies that explore multiple paths in parallel



(a) Required Number of Packets

(b) Reception Rate

Figure 7: Percentage of inconsistent links vs time



(a) Sum of One-way RNPs < 1000

(b) Sum of One-way RNPs < 5

Figure 8: Actual RNP cost as a function of the sum of one-way RNP estimator

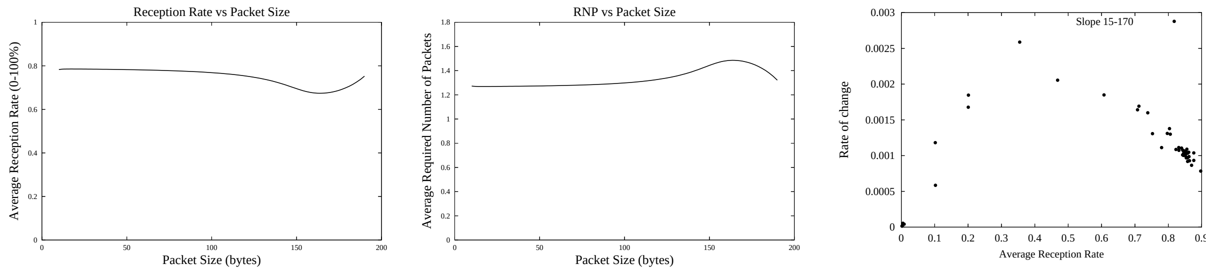
can significantly improve their chances of delivering the packet to the final destination.

We used data sets A and B for this task. Figure 6 shows the conditional probability of successfully receiving a packet if a node in another link also received it (1→1), and the conditional probability of successfully receiving a packet if a node in another link didn't receive it (0→1). We used three qualitatively different links (good, bad, and medium) to establish a comparison. Figure 6(a) shows that when a high quality link received a packet, most likely all other links will receive the same packet almost directly proportionally to the reception rate. Figures 6(b) and 6(c) show that when medium and bad quality links receive a packet from the source, the high quality links will also receive the same packet with very high probability. Figure 6(d) shows that when a good quality link does not receive a packet, with very high probability none of the other links will receive that packet either, even other high quality links. Therefore, there is very little incentive in spending more energy and bandwidth by trying to exploit multiple paths concurrently because chances are that none of the nodes will receive the packet. Figures 6(f) and 6(e) show that when bad and medium quality links do not receive the packet from the source, high quality links still have high chances of receiving it (as intuition would indicate).

Our main conclusion from this analysis is that when routing data using high quality links, we should use single path routing strategies and abandon multiple concurrent paths strategies because they provide very little benefit for the increased cost in terms of energy consumption and contention for resources.

4.4 Forward and Reverse Link Correlation

In this subsection, we study the forward and reverse link properties between a pair of nodes. These properties are important for any protocol that uses two-way communication between source and destination, and in particular, when using hop-by-hop reliable schemes with acknowledgements (acks). The



(a) Average Reception Rate (b) Average RNP (c) Rate of Change

Figure 9: Variation as a function of the Packet Size

key questions we try to answer are when are the best times to forward and acknowledge reception of a packet and what is the best metric to measure the combined data forward and reverse acknowledgment total cost.

For this analysis we use data set D. The first question we address is whether there is any advantages in immediately sending an ack from the 1-hop destination to the source for a packet just received. We call the *conditional required number of packets* (CRNP) and the *conditional reception rate* (CRR) the cost in the reverse direction (acknowledgment from destination to source), conditional upon successfully receiving a packet in the forward direction. Figs. 7(a) and 7(b) show the percentage of links that have higher CRNP and CRR in the reverse direction in comparison with their overall RNP and RR in the reverse direction when a packet was successfully received in the forward direction as a function of the time waited to send the ack back (τ, measured in seconds). We analyzed six qualitatively different classes of links that are of potential interest for routing purposes: all links, high quality links (RNP < 1.3), medium quality links ($1.3 <$ RNP < 5), low quality links(RR > 5) and links with high distance over RNP ratios classified into two groups (top 20% and top 40%). In Fig. 7(a) we see that for all links in general and for almost all classes of links (except for the top 40% of useful links), the percentage of links with CRNP larger than RNP is quite small, and it is smaller when the ack is sent back as soon as possible (the smaller the RNP the better). Similarly, in Fig. 7(b) we see that the percentage of links with CRR larger than RR is higher for all links in general and almost all classes of links (except the top 40% and bad links) when the ack is sent back immediately. Therefore, there is a significant advantage to send ack signals in the first few seconds after reception. The advantage diminishes as waiting time increases.

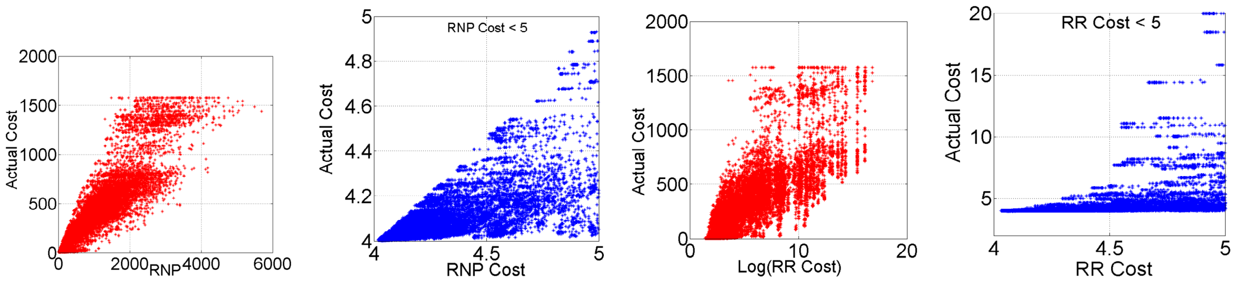
The other aspect we are interested is to determine the actual cost of a link when using hop-by-hop acks, and whether this cost could be estimated by the sum of the individual RNP in the forward and reverse direction, without requiring on-line measurement of the actual conditional cost. Considering only the RNPs in each link direction may not necessarily determine the best quality link if the forward and reverse direction present different levels of correlation for different links. Figure 8 shows the relationship between the sum of each individual RNP in each direction, and the actual RNP cost for all links when sending the acknowledgment immediately. In particular, we could observe from Fig. 8(b) that the sum of one-way RNPs is quite correlated with the actual link cost, with stronger correlation for smaller RNP values. In addition, the percentage of *consistency* (as defined in section 4.1) for this estimator is very high (96.2%). Therefore, the sum of RNPs is almost always a good indicator of overall quality of the link.

We conclude that there is significant benefit to acknowledge packets immediately because we increase the chances of the acknowledgement being received. We also conclude that an accurate cost metric could be well estimated using only the individual RNP values of each direction because the number of inconsistent links is not significant.

4.5 Packet Size

In this section, we analyze how the packet size affects the RR and the RNP. Each packet transmitted has a minimum fixed overhead provided by the starting symbol sequence, the radio header and the CRC. This cost is fixed and independent of the packet payload size. Our main motivation is to improve transmission efficiency by having proportionally less overhead per useful bit transmitted in the payload.

Using data set C, the first question we try to answer is to whether there is a fundamental trend or change that occurs for RR and/or RNP as a func-



(a) RNP Cost (b) RNP Cost < 5 (c) RR Cost (d) RR Cost < 5

Figure 11: Link Cost for 2-hop Forwarding Links

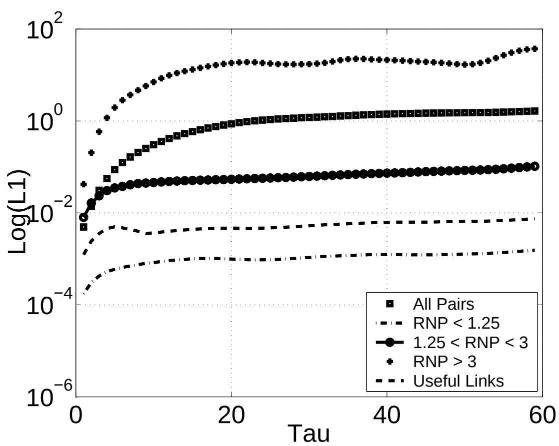


Figure 10: L1 RNP differences as a function of time

tion of packet size globally in the network. Figures 9(a) and 9(b) show the average RR and RNP as a function of different packet sizes for all links in the experiment. We see that the average RR slowly decrease and the RNP slowly increases until a packet size of 125 bytes.

The next aspect we would like to explore is to understand whether the change in packet sizes affect all links equally or if there is any relationship as a function of average RR. Figure 9(c) shows the relationship between the rate of change of the average RR as a function of the packet size for each pair of nodes in the experiments. The factor m on the y-axis corresponds to the slope of the linear interpolation of the average reception rates and different packet sizes for each link. All the different m coefficients are plotted as a function of the average reception rate for all packet sizes. We see that links with very low ($<$ 20%) or very high ($>$ 80%) RR are less affected by the change in packet size. Links with medium RR show a much larger variation. The quality variation of these group of links is strongly affected as we increase the packet size.

We can conclude that using larger packet sizes is certainly better in terms of efficiency by minimizing the overhead per useful bit transmitted without significantly increasing the RNP. Larger packet sizes could be use for opportunistic aggregation of data in intermediate nodes when multiple data packets are intended for the same destination, and when transmitting large amount of information.

4.6 Temporal Consistency of Links

It is also important to analyze long term correlations and the uniformity of links. The main motivation in this section is to determine how stable the links are and how often we have to update estimates about their quality.

Figure 10 shows the L1 norm of relative RNP differences for the five qualitatively different classes of links as a function of time (measured in hours). Note that y-axis follows a logarithmic scale and the fine structure of the differences is not completely visible. However, a number of conclusions is apparent and the most important features are well captured using logarithmic scale.

We see that the most stable links are the low RNP links with very small differences 10^-3 from the average over extended periods of time. An interesting result is that the variability of the high distance/RNP links (useful links) is also well below the average of all the links. Their change rate is less than 1% over 60 hours. Finally, we see that links with RNP $>$ 3 change at very rapid pace. However, since most of the links in our experiments were good links, the average variability for all links is skewed more heavily by the stability of the good links.

So, the stability of the links vary drastically depending on the quality of each link. Importantly, good quality links (high RR, low RNP) tend to be very stable over time. This is an additional reason

Algorithm 1 Two-Hop Correlation Shortest Path Algorithm

```
procedure CSP(W, i)    ▷ connectivity matrix and source node
    V = all n ∈ W − i                ▷ init candidate set
    InitState(W,i)          ▷ init all the state variables
    while V ≠ ∅ do          ▷ candidate nodes available
        u =RemoveBest(V)         ▷ based on shortest distance
        for all n ∈ W(u) do
            for all p ∈ W(u) do  ▷ all neighbors of u
                UPDATE(u, n, p)▷ update direct paths
            end for
            for all p ∈ W(n) do  ▷ all neighbors of n
                UPDATE(n, p, u)          ▷ update other
            end for
        end for
    end while
end procedure

procedure UPDATE(s, d, p)
    D(s, d, p) = BestDist(s, p) + W(s, d, p)
    P(s, d, p) = BestPath(s, p) + s
    if MD(d) > D(s, d, p) then          ▷ relaxation condition
        MD(d) = D(s, d, p)
        MP(d) = P(s, d, p)
    end if
end procedure
```

why those high quality links are exactly the ones that should be used in routing traffic.

4.7 Correlation Among Links on the Same Path

In this section, we study the link properties between two links in the same forwarding path. These properties are important since they can determine the total number of hops we need to gather information to make sound routing decisions.

We are interested in establishing the actual cost of a link when forwarding the packet in the path upon successful reception from the previous hop. Considering only the RNP of the previous hop link to choose the link for routing purposes may not necessarily determine the best link if the next hop link presents different levels of correlation for different previous hop links. Figure 11 shows the relationship between the sum of each individual RNP and RR in each hop, and the actual RNP cost for all links when forwarding the packet immediately to the next hop. We see from Fig. 11(b) that the sum of one-way RNPs in the forward direction is almost always greater than the actual cost of the forwarding path. The gains when using conditional RNP in the forwarding path are more significant than the one obtained in Section 4.4 and should be considered in order to establish the right quality metric for routing decisions.

We conclude that the link quality metric for routing should not only consider the RNP of individual links, but the actual conditional RNP based on the previous hop to get more accurate results with routing. Combining this result with the results of Section 4.4, means that when we are using a hop-by-hop acknowledgment mechanism it would be convenient to use an implicit acknowledgement strategy; i.e. forward the packet to the next hop immediately and combine it with a flag to indicate an ack to the previous hop from which we receive the packet.

5 Impact on Routing

In this section, we present some of the lessons learned in our previous analysis that could impact the design of routing algorithms and two new routing algorithms that could be used depending on the level of dynamics expected by the operation of the system.

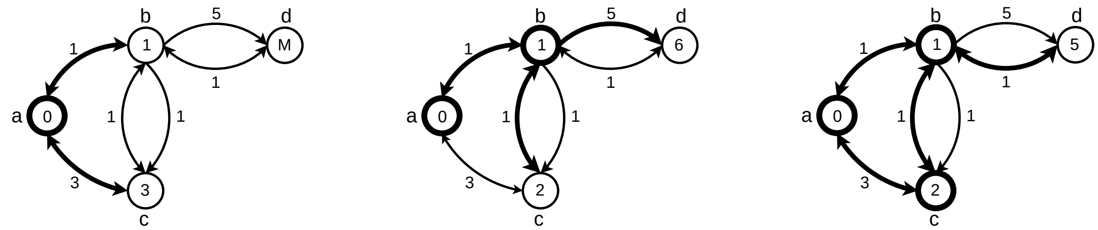
5.1 Lessons Learned

There are several lessons learned that can impact the way we do routing in sensor networks. While RR is the most intuitive metric to measure link quality, it is not always correct, and RNP should be used instead. High quality links are very good and quite stable over time. These links consistently rank among the top links over long periods of time. Therefore, centralized solutions are attractive because they allow to calculate an optimal solution to the shortest path routing problem and have low overhead in terms of the total number of control packets required to update the routing tables.

Nevertheless, there are cases when a centralized solution may not scale. For example, the network may not have enough number of good links to route all packets to the destinations. In addition, increasing levels of dynamics might be introduced by nodes failing, and/or by applying a sleeping scheduling mechanism on a subset of the available nodes [18, 19].

5.2 Centralized Routing Algorithm

Based on the analysis performed in Section 4, we showed that link correlations in the reverse-forward links and the consecutive links on the same path are significant and cannot be ignored. Therefore, any optimal solution of the shortest path routing problem should also consider the correlations among links.



(a) Step 1 (b) Step 2 (c) Step 3

Figure 12: Operation of the Two-Hop Correlation Shortest Path Algorithm

We have developed a generalized version of the Dijkstra algorithm that calculates the shortest path and considers correlations among links on the same path. The input of the algorithm consists of the source node id i and a 3-dimensional connectivity matrix indexed with variables source (s), destination (d), and previous hop (p) that have the following values:

$$W_{i,n,i} = \begin{cases} cRNP_{i,n} & \forall \text{ n neigh i,} \\ \infty & \text{otherwise.} \end{cases} \tag{1}$$

$$W_{s,d,p} = \begin{cases} sRNP_{p,s,d} & \forall \, \exists \text{ (s, d, p) tuples,} \\ \infty & \text{otherwise.} \end{cases} \tag{2}$$

The pseudo code is shown in Algorithm 1. The arguments needed are the W matrix and the source node i. The InitState call in line 3 initializes all the state variables. For the source i, $MD(i) = 0$, $MP(i) = \varnothing$, for the neighbors n of source i, $MD(n) = W(i, n, i)$, $MP(n) = i$, and for the rest of the (s, d, p) tuple $MD(s) = D(s, d, p) = \infty$, $MP(s) = P(s, d, p) = \varnothing$. The **while** loop that goes from lines 4 to 14 shows the main algorithm that is quite similar to standard SP Dijkstra. The differences are in the additional **for** loop in lines 10 to 12 and in the modified **UPDATE** procedure. Due to the correlations that may exist among links on the same path, the addition of a new node from the candidate set may affect previous routes to nodes that are not directly connected to the node being added to the covered set. In addition, we require to keep additional path state information in order to update the precedence matrix when a situation like this arises.

Figure 12 shows an example of the algorithm in action on a small four node network. The nodes are labeled with letters, the numbers inside the nodes indicate the shortest distance (MD) known at each step, and the edges show the different link costs depending on which previous node the packet comes from. For simplicity, several edges have the same cost in both directions. Note that there might be several edges between each pair of nodes depending on the previous hop (e.g. $b \rightarrow d$ cost is five if coming from a, and one if coming from c). Figure 12(a) shows the first step of the algorithm at initialization. The source node a is the only node in the defined set (bold), the shortest distance (MD) to the direct neighbors b and c is updated, and the distance of all the remaining nodes (d) is ∞ (M in the figure). Figure 12(b) shows the next step of the algorithm. Node b is removed from the candidate set, and all their neighbor nodes are updated. The results so far are identical to standard Dijkstra. Figure 12(c) shows the next step. When adding node c to the defined set (bold), this node does not have new direct neighbors from the candidate set. The standard Dijkstra algorithm would have stopped here. However, due to the high correlation with the link $b \rightarrow d$, a better shortest path to d that goes from $a \rightarrow c \rightarrow b$ is updated when using our algorithm. The running time of the algorithm is O(n^3) for dense and O($\frac{n^3}{logn}$ for sparse networks. It shows average improvements over the standard Dijkstra algorithm between 11% and 24% on set of networks that are formed by subset of nodes in our network.

5.3 Distributed Routing Algorithm

There are some scenarios where a centralized solution does not scale and distributed solutions are preferred. For example, there are several topology control algorithms [18, 19] that turn a subset of nodes off to save energy. These mechanisms change the underlying topology, forcing a recalculation of the optimal paths every time a node is turned off. Under these conditions, distributed solutions that depend on localized information may be more attractive.

We present a randomized distributed algorithm for solving the routing problem between two nodes in the network. We assume that nodes only know the RNP from other neighbors, and that packets include the aggregate RNP of the path (aRNP). This aRNP is the sum of all the RNPs in the path with respect to the original sender, and it is updated by each node in the packet forwarding path. Only directly connected nodes know when a neighbor is down (turned itself off and/or died). Source and destination nodes periodically send traffic to each other, enabling data traffic in both directions. The most important assumption we make is that *good* (i.e. low RNP) end-to-end paths are bidirectional as shown in section 4.4.

Each node maintains aRNP state for each (neighbor, sender) tuple. This state is updated every time a new packet from the sender is received via the neighbor. The aRNP estimated is calculated using an exponential weighted moving average (EWMA) with α factor of 0.01 (i.e. each new sample weighs 1% on the current estimate). Upon reception of a packet, each node unicasts the packet to the next hop. The next hop is determined probabilistically based on the aRNP estimates to the destination (calculated using previous traffic coming from the destination as explained above). The forwarding probability for each next hop candidate is given by the inverse of the aRNP for the (neighbor, destination) tuple divided by the sum of the aRNP inverses of all next hop candidate neighbors (neighbor, destination) tuples, all factors using a power factor δ. For instance, the probability of forwarding the packet to neighbor a, with final destination D, and $a \ldots n$ possible next hop neighbor candidates is given by:

$$P_a = \frac{(1/aRNP_{a,D})^{\delta}}{\sum_{a\leq i\leq n}(1/aRNP_{i,D})^{\delta}} \tag{3}$$

6 Simulation Results

We conducted simulations using EmStar [20] to measure the performance of our distributed algorithm in the presence of node dynamics. We set up a 60-node network for our simulations, with node degrees varying from 2 to 11, and with different link quality values. The best link had an RNP of 1.0, and the worst had an RNP of 6.387.

Each node in the simulation is initialized with the optimal routes calculated using the algorithm described in section 5.2 using all available nodes in the network. For each experiment, traffic is sent between source and destination with a constant bit rate. In each simulation run, we put 10% of the nodes to sleep (excluding source and destination), and then after a period of time (12 minutes), another 10% of the nodes go to sleep and the previous dormant nodes wake up. Each time nodes go to sleep, the optimal route between source and destination is re-calculated instantly (no overhead, ideal case).

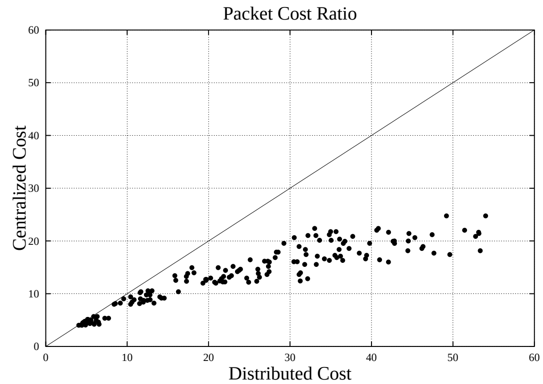


Figure 13: Total Packet Cost Centralized/Distributed

We implemented in Emstar a series of modules to simulate our distributed solution. LQE is the RNP link quality estimator implemented using the methodology described in the previous section. HBH, is a simple hop by hop reliability scheme that implements a stop and wait protocol. Finally, MIN-RNP implements the distributed routing algorithm described in section 5.3. We use a power factor δ of two for routing in our simulations.

The main performance metric we measured was the average total number of packets required to transmit one message back and forth between source and destination. Optimal routes tend to minimize the total number of packets required to transmit reliably between source and destination, and thus minimize energy consumption. Fig. 13 shows the relationship between the centralized optimal solution and our distributed solution. From the graph we see that when the cost of the path is small, the difference between the optimal and our distributed solution is not significant. When the path cost increases, the difference increases as well. The relative efficiency of the distributed case with respect to the centralized algorithm is 0.324 in the worst case. The convergence time of the algorithm is larger for higher cost paths. One of the reasons for this behavior is that the larger the cost path, the higher the probability of a path with a larger number of hops. The probability of our routing algorithm picking a sub-optimal path probabilistically increases with the path length and the average node degree.

It is interesting and instructive to analytically find the relationship between the total cost, including overhead, of the centralized algorithm and the cost of the distributed algorithm as a function of the frequency at which nodes switch their radios on/off. We assume that each time a node turns it's radio off, it floods a packet to all the nodes in the network informing it is disconnecting from the topology. No node floods the same packets twice, guaranteeing that each flooded packet will be forwarded at most N times.

We use the following parameters in our analysis. N is total number of nodes in the network (or routing domain); p is the percentage of nodes changing their radio status (on or off); f is the frequency of the status change (how frequent p% nodes change on/off); u is the update frequency (how frequent the nodes update their connectivity estimates to/from their neighbors); d is the data rate; α is the optimal cost of shortest path routing between source and destination (measured in total number of packets required per data packet sent); β: distributed routing algorithm cost between source and destination per data packet. This cost can be approximated to $\gamma * \alpha$, where γ is the efficiency factor).

The cost of the centralized algorithm (CA) is given by $CA = 2*(u+f*p)*N^2+\alpha*d$, being the first term the cost of sending (using flood) node and link quality estimation updates back and forth to the central node (running CA), and the last term the data forwarding cost. The cost of the distributed algorithm (DA) is given by $DA = \gamma * \alpha * d$.

In most practical cases of sensor networks, N will be O(100) in a single routing domain, $u << f$, p will be between 0.1 to 0.5 (depending on deployed density). Assuming $\gamma = 2$ (50% efficiency), $d = 0.1$ pkts/sec, and $\alpha = 10$, then the frequency of node status change should be less than ≈30 minutes in order for the distributed solution to compete with the centralized one.

7 Conclusion

We studied the statistical temporal properties of links used by low power wireless communication systems. We identified a set of properties that are the most relevant for the design of efficient routing protocols. For example, high temporal correlation implies the need to use the required number of packets instead of reception rate as the quality metric and implies the importance of using only high quality links. High variance in time lagged correlation of forward and reverse links implies the need for immediate sending of acknowledgments, while low short time variance of links favors communication using long packets. Correlations between links on the same multihop paths imply a need for the development of new types of shortest path algorithms, while high consistency of high quality links over time implies the rare need to update the models of communication links. Guided by the obtained insights, we have developed and analyzed two new routing algorithms: (i) a generalized Dijkstra algorithm with centralized execution that considers correlation of successive links in multihop communication, and (ii) a localized probabilistic algorithm that uses statistics about the reverse forwarding paths to establish probabilistic gradients on the forwarding path. We also performed simulations to analyze the overhead of the distributed solution with respect to the optimal solution in the scenario were a subset of nodes is powered down in order to save energy.

References

[1] D. Ganesan et al., "Complex behavior at scale: An experimental study of low-power wireless sensor networks," Tech. Rep. UCLA CSD-TR 02-0013, Feb. 2002.

[2] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *ACM Sensys*, 2003, pp. 1–13.

[3] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *ACM Sensys*, 2003, pp. 14–27.

[4] A. Cerpa, N. Busek, and D. Estrin, "SCALE: A tool for simple connectivity assessment in lossy environments," Tech. Rep. CENS Technical Report 0021, Sept. 2003.

[5] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *ACM Mobisys*, 2004, pp. 125–138.

[6] W. Ye, J. Heidemann, and . Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *IEEE Infocom*, 2002, pp. 1567–1576.

[7] T. S. Rappaport, *Wireless Communication: Principles and Practice*, Prentice Hall, 2000.

[8] RFM, "Tr1000 low power radio system, http://www.rfm.com," 2003.

[9] Chipcon, "Cc1000 low power radio tranciever, http://www.chipcon.com," 2003.

[10] J. Hill and D. Culler, "Mica: A wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov/Dec. 2002.

[11] Crossbow, "Mica2 wireless measurement system, http://www.xbow.com," 2003.

[12] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *ACM Sigcomm*, Sept. 2004, pp. 121–132.

[13] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *ACM Sigcomm*, Sept. 2004, pp. 133–144.

[14] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "High-througput path metric for multi-hop wireless routing," in *ACM Mobicom*, Sept. 2003, pp. 134–146.

[15] A. B. Owen, *Empirical Likelihood*, Chapman & Hall, 2001.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical learning; data mining, inference, and prediction*, Springer-Verlag, 2001.

[17] R. A. Thisted, *Elements of statistical computing*, Chapman & Hall, Ltd., 1986.

[18] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *ACM Mobicom*, July 2001, pp. 70–84.

[19] A. Cerpa and D. Estrin, "ASCENT: Adaptive self-configuring sensor networks topologies," in *IEEE INFOCOM*, Feb. 2002, pp. 24–60.

[20] L. Girod et al., "Emstar: a software environment for developing and deploying wireless sensor networks," in *USENIX*, June/July 2004, pp. 283–296.