Reliability and Acknowledgements in Low-Power Wireless Sensor Networks

by Author George William Shaw

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Kristofer S. J. Pister
Research Advisor

(Date)

* * * * * * *

Professor Alberto L. Sangiovanni-Vincentelli
Second Reader

(Date)

Contents

I. Introduction.......1
- A. Related Work.......1
- B. Reliability.......2
- C. Metrics.......2

II. Terms and Assumptions.......3

III. Basic Acknowledgements.......4
- A. No Acknowledgement Protocol.......4
  - 1) Reliability-Case Tx-Rx Actions.......4
  - 2) Mean Tx-Rx Actions.......4
  - 3) Route Congestion.......4
  - 4) Reliability-Case Tx-Rx Events.......4
  - 5) Mean Tx-Rx Events.......4
- B. Only End-To-End Acknowledgement.......4
  - 1) Reliability-Case Tx-Rx Actions.......5
  - 2) Mean Tx-Rx Actions.......5
  - 3) Route Congestion.......5
  - 4) Reliability-Case Tx-Rx Events.......5
  - 5) Mean Tx-Rx Events.......5
- C. Link-Layer Acknowledgement.......6
- D. Implicit Acknowledgment.......7
  - 1) Reliability-Case Tx-Rx Actions.......7
  - 2) Mean Tx-Rx Actions.......7
  - 3) Route Congestion.......7
  - 4) Reliability-Case Tx-Rx Events.......7
  - 5) Mean Tx-Rx Events.......7
- E. Immediate Acknowledgment.......8
  - 1) Reliability-Case Tx-Rx Actions.......8
  - 2) Mean Tx-Rx Actions.......8
  - 3) Route Congestion.......8
  - 4) Reliability-Case Tx-Rx Events.......8
  - 5) Mean Tx-Rx Events.......8
- F. Separate Acknowledgment.......8
  - 1) Reliability-Case Tx-Rx Actions.......8
  - 2) Mean Tx-Rx Actions.......8
  - 3) Route Congestion.......8
  - 4) Reliability-Case Tx-Rx Events.......8
  - 5) Mean Tx-Rx Events.......9

IV. Comparison of Basic Acknowledgements Protocols9
- A. Bounding Values.......10
- B. Bandwidth.......11
- C. Node and Network Energy.......11
- D. Resource Usage.......12
- E. Reliability.......12
- F. Latency.......13

V. Comparison of End-to-End Acknowledgements.......14
- A. Extraordinary Events.......14
- B. When to Use End-to-End Acknowledgments.......14
- C. End-To-End Acknowledgement with Link-Layer Acknowledgements.......15
  - 1) Calculations.......15
  - 2) Bandwidth.......15
  - 3) Node and Network Energy.......15
  - 4) Resource Usage.......15
  - 5) Reliability.......16
  - 6) Latency.......16

VI. Cumulative or Selective Acknowledgments.......16

VII. Conclusions.......16

References.......33

Reliability and Acknowledgements in Low-Power Wireless Sensor Networks

George William Shaw

***Abstract*— All wireless sensor networks with greater than 0% packet delivery rate (PDR) can be made 100% reliable—when given unbounded time to achieve successful packet delivery. Real systems, however, don't have unbounded time or resources. Reliability is the probability that a packet will be successfully delivered within the required time bound. The delivery reliability requirement and time bound are determined from the application-dependent delivery MTBF at which exceptional reliability measures can be tolerated. Reliability is generally achieved through packet delivery acknowledgements, and conventional wisdom says that reliable operation requires these acknowledgements be end-to-end. However, reliability meeting the application requirement can often be achieved without the overhead of an end-to-end acknowledgment. This report contributes a fundamental comparison of the common underlying mechanisms used to acknowledge packet delivery, their behaviors under changing conditions, and the environments in which they perform best. It suggests a method for evaluating actual packet delivery reliability to estimate the packet delivery rates used for planning the network, and a method for determining the time bounds that must be accommodated based on the estimated packet delivery rates so as to achieve the desired delivery reliability. It finds that by changing acknowledgment protocols, the reliability of the system can be increased by several orders of magnitude (to the limit of the inherent network reliability), that immediate acknowledgments perform generally best overall, and that if an end-to-end acknowledgment must be used, implementing it with a link-layer acknowledgment is the most efficient.**

I. Introduction

All multi-hop wireless sensor networks (WSN) with greater than 0% packet delivery rate (PDR) can be made 100% reliable—when given an unbounded amount of time to achieve successful packet delivery. Real systems, however, don't have unbounded time or resources, so achieving a reliability of less than 100% must be tolerated. Thus, reliability is the probability that a packet will be successfully delivered within the required time bound. Acknowledgement protocols can significantly affect delivery reliability, but a desired level of reliability cannot be "created" by the protocol; the level of reliability can only be reached or not. Reliability is determined by the packet delivery rate and the number of successful packet transmissions the protocol can achieve within the time bound. Conventional wisdom says that end-to-end acknowledgements are required for reliable operation. They have their uses, but their requirement depends on several specific factors. Using end-to-end acknowledgements adds network workload that can lower, rather than raise, the possible time-bounded delivery reliability for the route and for the network as a whole.

A. Related Work

Most papers look at acknowledgments as part of some other function, such as routing algorithms or network layer protocols. The author did not find any references that simply analyzed the fundamental characteristics of the basic available mechanisms. In [5], the authors define the metric ETX (expected transmissions) that uses live data for p_{data} and p_{ACK} to compute expected transmissions at the next hop and thus select the best route. They look at only one hop, however. In [17], the authors improve on ETX with MT, which looks ahead a hop and also considers the variance of the packet delivery rate in making its routing decision. While both of these have the live data available, and while they both attempt to pick a good route, they do not determine if the path they chose meets any outside criteria, such as a specific reliability or time bound. Many papers looks at various mesh routing systems, including braided diffusion, directed diffusion, disjoint multipath, braided multipath and others to some quality metric, but none determine or evaluate time bounded reliability.

We limit our analysis to the fundamental behavior of the underlying mechanisms on a linear route. Many papers look at some metric of energy consumption or efficiency, or time-bounded access to the wireless medium, but not bounded-time packet delivery. The exception is in [15], the author uses collected or live packet delivery rate data to evaluate the reliability of a route of links given time-varying packet delivery rate information and determines hard time bounds for packet delivery. We look at the fundamental operators on the time-bounded reliability of delivery: the work load of a protocol and the primary factor to which the work load is sensitive, the packet delivery rate. The conclusions suggest that by all measures immediate acknowledgments are generally the most robust and efficient; they are even built into the hardware of many radios. However, current work still uses other protocols and ignores the primary driver of delivery reliability, the packet delivery rate.

Messina in [10] proposes a network-layer protocol called PERLA that uses implicit acknowledgements rather than link-layer acknowledgements with packet caching and retransmission from alternate local motes, and finds a higher reliability. However, their gains appear to come entirely from scheduled retransmissions that avoid collisions compared to the competing protocols for which they do not consider scheduling. Using scheduled acknowledgement as in TSCH might alone eliminate that advantage. Further, their analysis assumes an ideal channel, i.e. $PDR = 1.0$, which is unrealistic for WSNs. A realistic packet delivery rate causes implicit acknowledgements to start competitively but to degrade more quickly as the packet delivery rate drops compared to immediate acknowledgements.

This report contributes a fundamental comparison of the common underlying mechanisms used to acknowledge packet delivery, their behaviors under changing conditions, and the environments in which they perform best. It suggests the method for evaluating actual packet delivery reliability from [15] for estimating the packet delivery rates used for planning the network, and a method for determining the time bounds that must be accommodated based on the estimated packet delivery rates so as to achieve a desired delivery reliability.

B. Reliability

Achieving high reliability is desirable, but high reliability cannot guarantee delivery, it only makes successful delivery more likely. A packet can be acknowledged at every hop, yet not reach the destination due to a failure of the packet to be forwarded at any hop. An end-to-end acknowledgement is the only way to guarantee that a packet, or group of packets, did arrive at the destination, but it is still impossible to *guarantee* that this acknowledgement can be received within the time bound. Thus, the highest reliability solution for a particular route is generally an efficient packet acknowledgement protocol to move packets to the destination and an efficient end-to-end acknowledgement to verify that they were received at the destination. However, the last measure of reliability in guaranteeing packet delivery, the end-to-end acknowledgment, comes at a high resource cost, can lower the reliability of the network as a whole, and is frequently not needed to achieve the required application-level reliability.

The delivery reliability requirement and time bound are determined from the application-dependent delivery MTBF at which exceptional reliability measures, such as retrying the entire packet sending process to as much as human intervention, can be tolerated. If the application sends one packet per minute and can tolerate a delivery MTBF of one delayed or failed delivery per day, then the required delivery reliability is

$$R = 1 - \frac{1}{60\, minutes * 24\, hours} = 99.93\%.$$

For non-critical packets the time bound may be the network timeout, which may be as long as several minutes or more, while for time-critical packets the time bound may be only a few hundred milliseconds. The time bound directly affects the delivery reliability since it limits the amount of work that the network may perform in attempting to deliver the packet. The work performed is counted in Tx-Rx actions, and the temporal character of this activity is counted in Tx-Rx events. Sufficient Tx-Rx events must be accommodated to achieve the desired reliability R. Only if the specified time bound is larger than the latency required by that number of Tx-Rx events can the protocol achieve this reliability. We refer to values related to this level of work as *reliability-case* values. In general, the less work a protocol must perform to achieve a given reliability, the more reliable the network can be for a given time bound. In some cases, the value of Tx-Rx events and Tx-Rx actions are the same. We will use the term Tx-Rx events when considering latency and otherwise use the term Tx-Rx actions to refer to activity involved. Using one term often implies information about the other.

We assume packets transmissions are Bernoulli trials that fail or succeed with some probability based on the packet delivery rates. This creates a distribution of the probability of a given number of successes in a given number of trials (Tx-Rx actions). The median of the distribution tells us the number of Tx-Rx actions at which 50% of the total Tx-Rx actions are successful, or a reliability R of 0.5. The mean of the distribution tells us the average number of Tx-Rx actions expected to achieve the given number of successful receptions. Determining the reliability-case number of Tx-Rx actions to achieve reliability R is generally a matter of finding the number of Tx-Rx actions (the point on the distribution) that the probability is R that the network can achieve the given number or more of successful receptions. Thus if R is 0.999, we want to find the number of Tx-Rx actions to be accommodated to ensure that 99.9% of the time we will have at least the required number of successful receptions. Reliability-case values specify the minimum resources required to achieve the desired reliability and are used where worst-cast resources must be evaluated, such as in packet buffers and delivery latency, where exceeding the value causes the failure of a packet to be delivered within the required time bound. The mean value, conversely, is useful for values which are averaged over time, such as goodput or energy consumption.

C. Metrics

In this analysis we are concerned with motes that are power-constrained (battery operated), resource constrained, and latency sensitive. They have limited battery life, memory, processing power and radio bandwidth. The longer a mote is awake or the longer the receiver or transmitter is on, the more energy it consumes. The longer a mote must hold a packet for possible retransmission, the more memory it consumes. The more frequently a mote transmits the more radio bandwidth it consumes. The transmission and reception of packets is generally the largest resource consumer on a mote. Reducing the number of Tx-Rx actions can not only increase the packet delivery reliability, but by reducing the resources used, it can increase the battery life of the mote, the battery life of the network and the reliability of the network as a whole. Tx-Rx Actions are similar to expected transmissions, ETX, described by [5], and to minimum transmissions, MT, described by [17] as routing metrics, except they are calculated to greater detail and rather than using dynamically measured values per link to plan routes, we are using predicted or statically measured values for design and planning.

We will primarily compare per-packet acknowledgements (ACK) protocols. As a group, we will look at protocols that transmit no acknowledgement, only end-to-end acknowledgement, and three link-layer acknowledgements: implicit acknowledgement, immediate acknowledgement, and separate acknowledgement. Then, as a group, we will also look at the case for end-to-end acknowledgments and review link-layer protocols that include an end-to-end acknowledgement per data packet. These selections describe limiting behavior for other protocols and set the expected bounds of the system.

II. Terms and Assumptions

Originate is used to refer to a link-layer transmission of a packet from the source mote into the network toward the destination mote that may entail many transmissions en route. A single packet sent from the network level to the link level and into the network may originate the same packet several times if required by the acknowledgement protocol.

Source is the mote that originates a packet and is thus the first mote in the sequence of motes on the route that transfers the packet to the destination.

Destination is the last mote in the sequence of motes that receives a packet that originates at the source.

Route refers to the entire sequence of motes that transfers a packet from the source mote to the destination mote.

Hop is a counting measure referring to the number of wireless links required to transfer a packet from one mote, through neighboring motes, to another mote.

(Re)Transmission refers to data transfers between motes on the route to the destination. Transmission implies a receiver will attempt reception.

Reliability R is defined as the probability that a packet will arrive at its destination within a specified time bound, or latency.

Packet delivery rate (PDR) is the probability that a packet will be successfully transferred from one mote to an immediate neighbor. For the most accurate estimation of Tx-Rx events, the packet delivery rate should consider CCA failures and scheduling delays as packet delivery failures that lower the packer delivery rate.

Acknowledgements (ACKs) are packets transmitted to confirm the arrival of a data packet and are only transmitted after the successful reception of a packet (positive acknowledgements).

Failure is the lack of or incorrect reception of a packet for any reason, such that the data in the packet cannot be considered valid by the available measures, or the arrival of a packet outside of the required time bound.

Reliability-case values are the minimum amount of an item to be accommodated to achieve the desired reliability R.

p_{data} is the packet delivery rate for a data packet.

p_{ACK} is the packet delivery rate for an acknowledgement packet.

Tx-Rx Actions count the occurrences of the operation of a transmitter and receiver on a pair of motes.

Tx-Rx Events count the periods of transmitter or receiver activity, regardless of how many transmitter or receiver are involved. It is a metric to evaluate latency and can be translated into seconds.

In most cases, one can expect $p_{ack} > p_{data}$ for several reasons. When sent as an explicit packet, acknowledgement packets are generally shorter than data packets, so the contribution to bit error rate due to Gaussian noise is reduced. If the acknowledgment is transmitted soon after the data packet then the channel coherence that made the data packet successful is maintained while CAA has suppressed the transmissions from local nodes [16]. While asymmetric packet delivery rates have been reported by [9],[4] has showed by physically swapping pairs of motes that asymmetries can be due to reception sensitivities of the different motes, i.e., inconsistency in the quality of the receiver hardware. Thus, we assume all packet delivery rates are symmetric.

We also make several simplifying assumptions:

$p_{AB}(f,t) = p_{data}$ or p_{ACK} and is a constant for all routes, times, and frequencies.

$p_{ACK} \geq p_{data}$, as discussed above.

All probability distributions are independent and identically distributed.

Each node has a single parent and is routed in a linear topology. Multiple parents in a mesh topology can improve the packet delivery rate but complicate evaluation of the resources used and are not treated in this report.

T_{hop}, the delay per successful hop, is a constant in the network and independent of the success or failure of previous hops or previous attempts to transmit the packet on this hop. This is consistent with some of the most popular MAC layers, such as B-MAC, SCP, TSCH, TSMP, but is not the case with epoch-based duty-cycling with epochs that are long compared to the length of a packet.

For the graphs in this report, a reliability of R = 0.999 is used unless labeled otherwise as this represents greater than one failed delivery per day using the example above of one transmission per minute. Several metrics are evaluated at a packet delivery rate of 1.0, 0.9, 0.6, and 0.3 to characterize the protocols. Trends for other lower and higher reliability are discussed in the text and in Table 5 and Table 6.

III. Basic Acknowledgements

A. No Acknowledgement Protocol

A packet can be reliably sent with a no-acknowledgement protocol to achieve some reliability $R > PDR$ by unconditionally originating the packet the reliability-case times to achieve a probability of delivery greater than R. If $R < PDR$ then the packet need only be originated once. With its simplicity, the technique might at first be considered to reduce resource usage, and is commonly used in single-hop home sensor networks where high reliability is not a concern.

A single data packet traveling k hops to its destination with no retransmissions at any hop will arrive with a probability of $p_{data}{}^{k}$. However, at a 90% packet delivery rate and ten hops, the probability that the single packet sent from the source mote arrives at the destination mote is $p_{AB} = 0.9^{10} \cong 35\%$. If we originate the packet some multiple b times, we can increase the delivery probability. The probability that all packets are lost is the overall error rate, ε.

$$\varepsilon = (1 - p_{data}{}^{k})^{b}$$

The probability of at least one successful reception p_{AB}, which is all that is required, is also the reliability R.

$$R = p_{AB} = 1 - \varepsilon = 1 - (1 - p_{data}{}^{k})^{b}$$

Thus, the packet originations b for reliability R is

$$b = \frac{\log(1-R)}{\log(1 - p_{data}{}^{k})}.$$

To achieve a reliability R of 0.999 over ten hops with $p_{data} = 0.9$, the packet must be originated sixteen times. Multiple copies of the same packet are originated and generally received to ensure reliability R is met and only in about 1 in 1000 cases will just one packet reach the destination. It is for the assurance of the 1 in 1000 case that the sixteen packets must be originated in this scenario.

1) Reliability-Case Tx-Rx Actions

An originated packet fails along its route following a geometric distribution. The probability of $x = k$ Tx-Rx actions traversing the route is the same distribution plus the probability of success reaching the last mote, mote n. Since these probabilities sum to one, the distribution is inherently normalized. Thus, the probability of x Tx-Rx actions is

$$p_{\#tx}(x) = \begin{cases} p^{x-1} \cdot q, & x < n \\ p^{x-1} \cdot q + p^{x}, & x = n \end{cases}.$$

The expectation value for Tx-Rx actions for each packet sent along the route is

$$EX_{NA}(n) = \sum_{x=1}^{n} p_{\#tx}(x) \cdot x.$$

For the no-acknowledgement protocol, the reliability-case Tx-Rx actions to be accommodated to ensure reliability R for hops k is the packet originations multiplied by the expected Tx-Rx actions for each packet.

$$RCTxRxActions = b \cdot EX_{NA}(k)$$

2) Mean Tx-Rx Actions

While the packet will on average be received at the destination after some smaller number than packet originations b, with no acknowledgement to confirm delivery, all b packets must *always* be transmitted to assure reliability R is achieved. As a result, the mean packet originations is also b, so the reliability-case and mean values are the same. The majority of the packets *are assumed to be lost* somewhere in route.

$$MeanTxRxActions = RCTxRxActions$$

3) Route Congestion

The route congestion is computed using the distribution produced by $p_{\#tx}$, and accumulated towards the source as if to create the cumulative distribution function (CDF). Since the source only transmits and the destination only receives, the CDF is summed to itself shifted one hop toward the destination and then renormalized. Multiplying by $MeanTxRxActions$ creates the Tx-Rx actions distributed across the route.

4) Reliability-Case Tx-Rx Events

Since there is no acknowledgement to wait for, packets can be originated with no delay. Once all b packets have been transmitted the resources on the source mote are freed for other activity. However, the time bound for reliability R requires that the data packet be delivered, so we must include the time for the last packet to trickle through the route to the destination. There are many overlapping Tx-Rx events as the packets move through the network.

$$RCTxRxEvents = b + (k - 1)$$

5) Mean Tx-Rx Events

As with the mean Tx-Rx actions, with no acknowledgment to confirm delivery, all b packets are always transmitted. The reliability-case and mean Tx-Rx events have the same value.

$$MeanTxRxEvents = RCTxRxEvents$$

B. Only End-To-End Acknowledgement

A packet can be reliably sent with an only end-to-end acknowledgement protocol to achieve some reliability R by conditionally or unconditionally resending the packet some zero or more times until delivery is verified by reception of the acknowledgement packet. Conditionally resending the packet has the advantage that only the required Tx-Rx actions occur, but suffers in delivery latency because a failed-acknowledgement timeout must occur before the packet is originated again. Unconditionally originating the packet has the advantage of reducing the delivery latency because the packet can be transmitted without waiting for a failed-acknowledgement timeout, but suffers from the possibility of unneeded transmitting of the packet, unneeded Tx-Rx actions across the network, and a higher probability of multiple copies of the packet reaching the destination.

When the sender of a data packet requests an end-to-end acknowledgement then the destination must create and transmit an acknowledgement packet back through the network. The end-to-end acknowledgement packet is independent in scheduling, routing, delivery probability and transit time from the initiating data packet. The probability that a data packet originated from source mote A arrives at destination mote B is ${p_{data}}^{k}$, and the probability that the end-to-end acknowledgement packet transmitted from destination mote B arrives at source mote A is ${p_{ACK}}^{k} \leq {p_{data}}^{k}$, for a packet-acknowledgement round-trip probability of $p_{ABA} \leq {p_{data}}^{2k}$. Using the prior no-acknowledgement protocol scenario of a ten-hop distant route and 90% packet delivery rate, the total hops for the packet-acknowledgement round-trip is twenty hops. The probability that the data packet originated from source mote A arrives at destination mote B is again about 35%, but the probability that the acknowledgement packet originated from destination mote B is also received at source mote A is as low as about 12%. Even if the data packet is received correctly at destination mote B, there is still a probability $1 - {p_{data}}^{k}$ (about 65%) that the data packet will be originated one or more additional times because the acknowledgement packet did not complete the trip from destination mote B to source mote A. In general, whether conditionally or unconditionally originated, multiple copies of the data packet are received at the destination and by this measure more packets will be originated than necessary.

The calculation of the reliability-case packet originations b that must be accommodated to achieve reliability R is the same as for the no-acknowledgement protocol, except that for a k hop destination $2k$ hops must be successfully traversed since success requires the acknowledgement packet to also be received. Thus, the packet originations b for reliability R is

$$b = \frac{\log(1-R)}{\log(1-{p_{data}}^{2k})}.$$

The expectation value for Tx-Rx actions is

$$EX_{EE}(n) = \sum_{x=1}^{n} p_{\#tx}(2x) \cdot 2x.$$

1) Reliability-Case Tx-Rx Actions

For the only end-to-end acknowledgement protocol, the number of reliability-case Tx-Rx actions to be accommodated to ensure reliability R for hops k is the packet originations b multiplied by the expected Tx-Rx actions for each packet.

$$RCTxRxActions = b \cdot EX_{EE}(k)$$

2) Mean Tx-Rx Actions

The mean Tx-Rx actions are the mean originations that occur multiplied by the expected Tx-Rx actions for each.

$$MeanTxRxActions = \frac{1}{{p_{data}}^{2k}} \cdot EX_{EE}(k)$$

3) Route Congestion

The route congestion is computed using the distribution produced by $p_{\#tx}$ for $2k$ hops and accumulating towards the source as if to create the cumulative distribution function (CDF). Since the source only transmits and the destination only receives, the CDF is summed to itself shifted one hop to toward the destination and then renormalized. However, since the returning acknowledgement is assumed to traverse the same route, the distribution is folded around the destination to create the backtracking. Then, multiplying by $MeanTxRxActions$ creates the Tx-Rx actions distributed across the route.

4) Reliability-Case Tx-Rx Events

The protocol can either conditionally or unconditionally originate a duplicate packet. If the data packet is conditional originated, then the source must wait for $2k$ Tx-Rx events to determine that the end-to-end acknowledgment has not been received and all reliability-case packet originations b must be accommodated. There are no retransmissions en route.

$$RCTxRxEvents = b \cdot 2k$$

If the data packet is unconditionally originated for all of the reliability-case packet originations, the minimum Tx-Rx events is that required to transmit all b packets and for the acknowledgement from the last one to be received at the source. There are many overlapping Tx-Rx events as the packets move through the network and the acknowledgments return.

$$RCTxRxEvents = b + (2k - 1)$$

This value is plotted in Figure 18.

5) Mean Tx-Rx Events

The protocol can either conditionally or unconditionally originate a duplicate packet. If the data packet is conditional originated, then the source must wait for $2k$ Tx-Rx events to determine that the end-to-end acknowledgment has not been received for the mean packet originations needed to achieve success. There are no retransmissions en route.

$$MeanTxRxEvents = \frac{2k}{{p_{data}}^{2k}}$$

If the data packet is unconditionally originated, the minimum Tx-Rx events is that required to transmit the mean originated packets and for the acknowledgement from last packet to be received at the source. There are many overlapping Tx-Rx events as the packets move through the network and the acknowledgments return.

$$MeanTxRxEvents = \frac{1}{{p_{data}}^{2k}} + (2k - 1)$$

This value is plotted only in Figure 17.

C. Link-Layer Acknowledgement

A data packet can be reliably sent with link-layer acknowledgements to achieve some reliability R by conditionally retransmitting the data packet at each mote in the route that does not receive, within the acknowledgement timeout, an acknowledgement packet from its neighbor indicating successful reception. We assume that the data packet is retransmitted at each mote as many times as required to achieve successful delivery, though many systems have limits. Since only one hop is involved for acknowledgement, the probability of successful data packet delivery is p_{data} and the probability of successful data packet and acknowledgement packet delivery is $p_{data} \cdot p_{ACK}$.

Link-layer acknowledgements thus have an advantage over no-acknowledgements or only end-to-end acknowledgements because only local motes are involved in the retransmission and thus fewer data packets are retransmitted due to failed acknowledgement packets. Because the only redundant packets ever transmitted are when the data packet has successfully traversed the one hop but the acknowledgement packet has not successfully returned the one hop, Tx-Rx actions are reduced with all of the subsequent advantages. Thus, even with an application-level verification mechanism, link-layer acknowledgments are justified because they improve performance [14].

Retries occur at each hop as the transmission succeeds or fails, and are thus Bernoulli trials, with the trials ending in success on the last trial at the destination mote B. The probability of success of the data packet being forwarded through the network at each hop is $p = p_{data}$ and is considered independent of the success of the acknowledgement packet. The probability of exactly retries b to achieve hops k is a negative binomial distribution given by

$$NB(b; k, p) = nchoosek(b + k - 1, k - 1) \cdot p^k(1 - p)^b.$$

The probability p_{AB} of sending a packet from source mote A to destination mote B and incurring up to retries b to achieve hops k is the cumulative distribution function (CDF) of the negative binomial distribution.

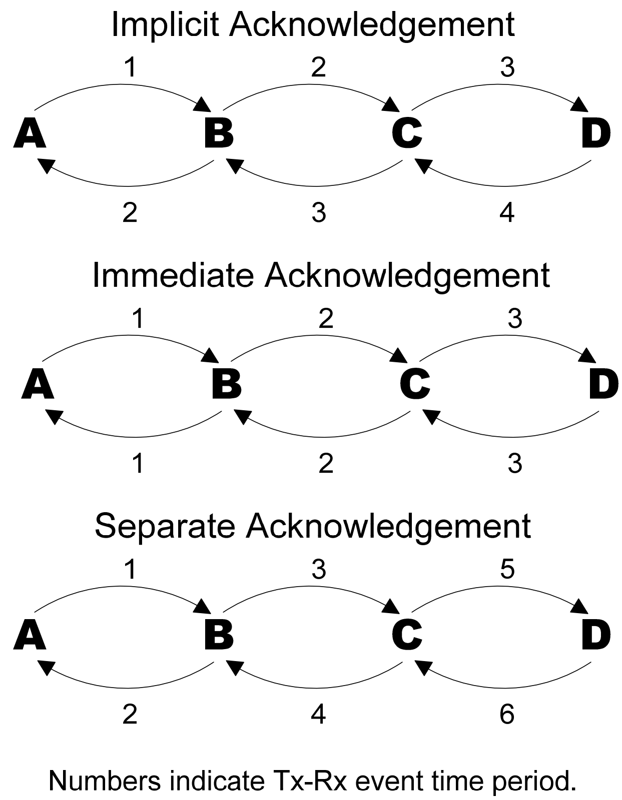


Figure 1. Order of Tx-Rx events across hops A, B, C, D for basic acknowledgment protocols.

$$R = p_{AB} = CDFNB(b; k, p) = \sum_{i=0}^{b} NB(i; k, p)$$

The probability p_{AB} of success on the route from source mote A to destination mote B is also the reliability R of that route. We perform an inverse CDF by computing the sum until the $CDF >= R$ to compute retries b to be accommodated.

$$ForwardTxRxActions = b + k.$$

While this represents the Tx-Rx actions to forward the data packet through the network, acknowledgement activity also consumes resources so this number does not tell the whole story. The data packet is transmitted some number of additional times because the acknowledgement packet is not successfully received.

To ensure that the data packet is delivered and that all acknowledgement activity is complete within the time bound, the probability of delivery and the computation of the retries b to be tolerated to achieve hops k with reliability R must be that of the successful delivery of the data packet and the successful delivery of the acknowledgement packet. To determine the Tx-Rx actions, since both a data and an acknowledgement Tx-Rx action can occur, we must compute the number of both. We assume that duplicate data packets received due to unsuccessful acknowledgement packet reception are dropped or do not interfere with timing. The approximation of the total Tx-Rx actions varies with the three link-layer protocols below, but is similar in form to:

$$RCTxRxActions = (b_{data+ACK} + k) + (b_{ACK} + k).$$

The term $(b_{data+ACK} + k)$ accounts for the number of data packet Tx-Rx actions and the term $(b_{ACK} + k)$ accounts for the number of acknowledgement packet Tx-Rx actions.

The mean Tx-Rx actions for hops k are generally computed as

$$MeanTxRxActions = \frac{k}{p_{data} \cdot p_{ACK}} + \frac{k}{p_{ACK}}.$$

The first term accounts for the data packet Tx-Rx actions. The second term accounts for the acknowledgement packet Tx-Rx actions and considers that some acknowledgment packets are not successful.

Since failures are distributed throughout the route, the average activity for all motes is spread evenly across the route. The source and the destination, however, do not forward the packet, so they generally incur half the activity.

$$MeanRouteCongestion = [a_0 \ldots a_k] \cdot \frac{1}{\sum_{i=0}^{k} a_i} \cdot MeanTxRxActions$$

The vector represents the pattern of Tx-Rx actions across the route while the scalar summation normalizes the distribution. Multiplying by the mean allocates the Tx-Rx actions.

The reliability-case and mean Tx-Rx events are computed similarly to the Tx-Rx actions, but the components may be grouped differently to represent temporally separated events.

D. Implicit Acknowledgment

The link-layer implicit-acknowledgement protocol consists of transmitting a data packet one hop from mote A to mote B, and mote A implying an acknowledgement of successful delivery of the packet by overhearing mote B forwarding the packet to the next mote C. This saves resources required for transmitting an explicit acknowledgement packet, except for the last hop since the packet is not forwarded. However, mote A must store the data to recognize the forwarded packet, and will generally need to listen to a data-packet length transmission rather than an acknowledgement-packet length transmission. This is also part of the type of acknowledgement structure performed by systems that piggyback acknowledgments on returning data packets. While they do not overhear the packet being forwarded, they also do not transmit an explicit acknowledgement. Instead, they wait for a return data packet to carry an acknowledgment.

Since the forwarded packet used for acknowledgement is a data packet and is independent in scheduling and delivery probability, we assume $p_{ACK} = p_{data}$. Thus, the probability that the data packet transmitted from mote A is received at mote B and the packet forwarded by mote B to mote C is received by mote A is

$$p_{data+ACK} = p_{data} \cdot p_{ACK} = {p_{data}}^2.$$

1) Reliability-Case Tx-Rx Actions

Each acknowledgement en route adds only one-half Tx-Rx action for the overheard forwarded packet.

$$RCTxRxActions \cong (b_{data+ACK} + k) + \frac{b_{ACK} + k}{2}$$

2) Mean Tx-Rx Actions

The mean Tx-Rx actions count the same types of activity as the reliability-case values, except using mean values, and with the addition a value for the explicit acknowledgment transmitted at the last hop.

$$MeanTxRxActions \cong \frac{k}{{p_{data}}^2} + \frac{1}{2} \cdot \frac{k+1}{p_{data}}$$

3) Route Congestion

Since the implicit acknowledgment does not transmit an explicit acknowledgement when a packet is forwarded but does receive a packet for acknowledgement, middle motes incur one-half a Tx-Rx action.

$$MeanRouteCongestion = [1\ 1.5 \ldots 1.5\ 1] \cdot \frac{1}{\sum_{i=0}^{k} a_i} \cdot MeanTxRxActions$$

4) Reliability-Case Tx-Rx Events

Since the acknowledgment occurs at the same time as the data packet is forwarded, it does not add latency and counts as the same event except on the last hop, when an explicit acknowledgment must be transmitted.

$$RCTxRxEvents \cong (b_{data+ACK} + k) + 1$$

Networks that piggyback acknowledgments on return data packets will need an additional factor to account for the delay until a return data packet occurs. Many also will transmit a separate acknowledgment packet should a data packet not occur soon enough, making these networks hybrids.

5) Mean Tx-Rx Events

The first data packet Tx-Rx event and last explicit acknowledgement Tx-Rx event are separate, while all the Tx-Rx actions in the middle are combined data-packet forwarding/acknowledgment events.

$$MeanTxRxEvents \cong \frac{k-1}{{p_{data}}^2} + \frac{2}{p_{data}}$$

As above, networks that piggyback acknowledgments on return data packets will need an additional factor to account for the delay until a return data packet occurs. Many also will transmit a separate acknowledgment packet should a data packet not occur soon enough, making the networks hybrids.

E. Immediate Acknowledgment

The link-layer immediate-acknowledgement protocol consists of transmitting a data packet one hop from mote A to mote B, and then mote B immediately transmitting an acknowledgement packet to mote A. This is the acknowledgment type performed by hardware acknowledgments as well as the acknowledgment type performed in some TDMA systems. This saves resources by reducing latency to schedule an acknowledgement packet and greatly increases the likelihood of successful acknowledgement packet delivery due to channel coherence.

As a bounding upper-limit of acknowledgement reliability for link-layer protocols, we assume $p_{ACK} = 1.0$, i.e. that an acknowledgement packet is always successful. Thus, the probability that the data packet transmitted from mote A is successfully received at mote B and the acknowledgment packet transmitted from mote B is also received at mote A is:

$$p_{data+ACK} = p_{data} \cdot p_{ACK} = p_{data}.$$

1) Reliability-Case Tx-Rx Actions

Since the acknowledgment packet is always successful there are no retries and only one acknowledgment packet is transmitted per hop.

$$RCTxRxActions = (b_{data+ACK} + k) + (0 + k)$$

While this is an idealistic representation, it is not far from correct with reasonable packet delivery rates, and is nonetheless positions correctly as a bounding value with respect to the other protocols. See Bounding Values, below.

2) Mean Tx-Rx Actions

The mean Tx-Rx actions count the same types of activity as the reliability-case values, except using mean values.

$$MeanTxRxActions = \frac{k}{p_{data}} + k.$$

3) Route Congestion

$$MeanRouteCongestion = [1\ 2 \ldots 2\ 1] \cdot \frac{1}{\sum_{i=0}^{k} a_i} \cdot MeanTxRxActions.$$

4) Reliability-Case Tx-Rx Events

Since the acknowledgment occurs immediately after the data packet is successfully received it does not add latency and counts as the same event.

$$RCTxRxEvents = (b_{data+ACK} + k) + 0$$

5) Mean Tx-Rx Events

As above, since the acknowledgment occurs immediately after the data packet is successfully received it does not add latency and counts as the same event.

$$MeanTxRxEvents = \frac{k}{p_{data}}.$$

F. Separate Acknowledgment

The link-layer separate-acknowledgement protocol consists of transmitting a data packet one hop from mote A to mote B, and then mote B scheduling and transmitting a separate acknowledgement packet to mote A before forwarding the data packet to the next mote C. This acknowledgment type also occurs when acknowledgments are performed at a higher level in software.

As a bounding lower-limit of acknowledgement reliability for link-layer protocols we assume $p_{ACK} = p_{data}$. Thus, the probability that the data packet transmitted from mote A is received at mote B and the acknowledgment packet transmitted from mote B is also received at mote A is:

$$p_{data+ACK} = p_{data} \cdot p_{ACK} = {p_{data}}^2.$$

1) Reliability-Case Tx-Rx Actions

$$RCTxRxActions = (b_{data+ACK} + k) + (b_{ACK} + k).$$

While this represents a pessimistic representation, certainly in applications where data packets and acknowledgment packets are of similar length this would be nearly typical. See Bounding Values, below.

2) Mean Tx-Rx Actions

The mean Tx-Rx actions count the same types of activity as the reliability-case values, except using mean values.

$$MeanTxRxActions = \frac{k}{{p_{data}}^2} + \frac{k}{p_{data}}$$

3) Route Congestion

$$MeanRouteCongestion = [1\ 2 \ldots 2\ 1] \cdot \frac{1}{\sum_{i=0}^{k} a_i} \cdot MeanTxRxActions$$

4) Reliability-Case Tx-Rx Events

Since all the Tx-Rx actions are separate, they are all also separate events.

$$RCTxRxEvents = RCTxRxActions.$$

5) Mean Tx-Rx Events

Since all the Tx-Rx actions in the mean are separate, they are all also separate events.

$$MeanTxRxEvents = MeanTxRxActions.$$

IV. Comparison of Basic Acknowledgements Protocols

We first look at five common acknowledgment protocols to reliably deliver packets: none, only end-to-end, implicit, immediate, and separate. The basic acknowledgment protocols reviewed vary in work load imposed on the network with the number of hops and the available packet delivery rate. While the number of hops is in part a hardware architectural decision, it does affect the packet delivery rate as the physical distance between motes and their location affects the signal strength and signal multipath [1]. The resulting packet delivery rate, however, varies greatly over time. Figure 3 shows the packet delivery rate data for one link collected over 26 days from an industrial site in Berkeley, California [6]. The network is time-slotted channel-hopping (TSCH), and the graph shows the link packet delivery rate (stability) over all 16 IEEE 802.15.4 channels (0-15 on the graph mapped to 11-26 in the standard). The baseline is 100% packet delivery rate, and the detail below the line shows how much the packet delivery rate has dropped. Note that channels 0, 3, 10, and 14 had high packet delivery rates for the first 13 days with drops in packet delivery rates for the days after, channels 5 and 12 experienced drops in packet delivery rates in the first 13 days and high packet delivery rates for the days after, and 2, 7, 8, 9, 11, 13, 15 had constantly varying packet delivery rates. These packet delivery rate fluctuations are persistent

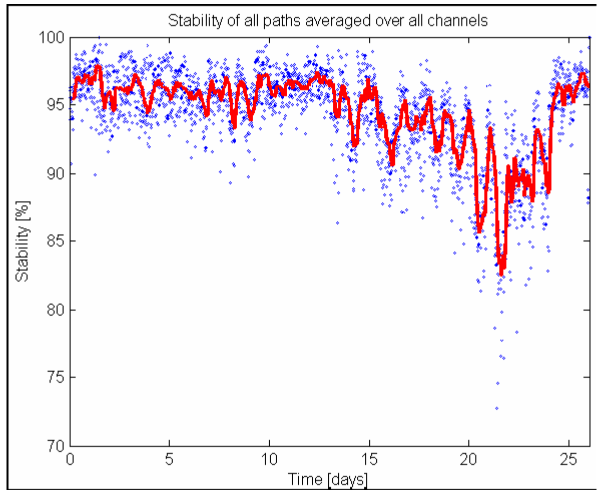


Figure 2. Overall network PDR (stability) as a function of time throughout the sample period with each point representing one 15-minute interval. The solid line is a 5-hour moving average. [6]

on the order of days. Since the network is a type of TDMA, there are no collisions from several nodes trying to share the same channel. All of the packet delivery rate degradation is from the environment. Clearly one could not select one "good" channel and expect a stable packet delivery rate, or even the best packet delivery rate. All channels show packet delivery rate variations lasting for days. Even the normally expected quiet channel 24 (13 on the graph) varies constantly throughout the test period. The variance of the packet delivery rate for this link might be improved by hopping through only the best channels rather than all the channels.

Figure 2 shows the average packet delivery rate of the entire network of 44 nodes and 89 links over the same period [6]. Note that for the first two weeks of the period the 5-hour average of the packet delivery rate was around 95% most of the time, but for the remainder of the period dropped significantly to as low as 83% and then recovered on the last couple of days. This data is a network average, but individual links fared much worse. In [15], the author analyzes this same site data. Figure 6 shows the probability distribution function of the packet delivery rates collected in 10% intervals. Note that many links have a significant probability of lower packet delivery rates, one even a notable amount in the 0-10% range. Thus while the network-wide packet delivery rates averages in the mid-80% or above even in the worst times, the individual link data tells a different story.[6][18]

We evaluate a design and choose

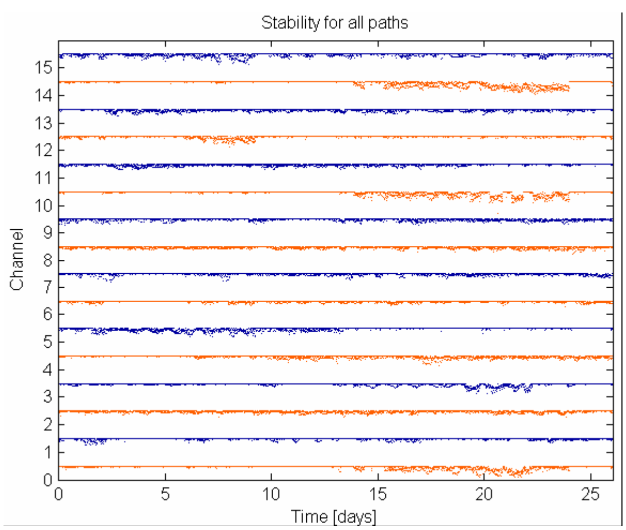


Figure 3. The PDR (stability) variations on the link form node 24 to node 17 shows daily periodicity corresponding to operational days in the factory. [6]

Protocol Sensitivity to Changes in PDR	
Order	**Protocol**
Least	Immediate
	Implicit
	Separate
Most	Only End-to-End
As reliability R increases, sensitivity increases.	

Table 1. Protocol Sensitivity

a time-bounded reliability of just 90% and, based on the early network-wide average packet delivery rate (stability) of above 90%, assume a fixed link packet delivery rate of 90%. To achieve this design, for all links the probability must be 90% or more that a packet delivery rate of at least 90% occurs. See the bar graph for link one in Figure 6. Link one appears to be a reasonable link relative to the mesh diagram, but note that the 90%-100% packet delivery rate interval has a probability of less than 90%, so packet delivery rates in the 80% to 90% interval will need to be tolerated on this link to achieve 90% reliability. At higher target reliabilities of 99%, 99.9% or more, some occurrences of packet delivery rates in the 50% to 60% interval or even the 40% to 50% interval will need to be tolerated. This means that in determining the time bound for a route, these lower intervals must be the design values. If not, then these lower packet delivery rates will occur, requiring more Tx-Rx events than we have allocated time for in the time bound, and since they occur within the probability of our chosen reliability, cause the time bound to be exceeded and the time-bounded delivery to fail more often than the desired reliability R [15].

Not all links on a route may fare so poorly, but because the number of Tx-Rx events is generally an exponential function of the packet delivery rate, using the average packet delivery rate of a network is not sufficient for design. Lower packet delivery rates will regularly occur on some links so the protocol must behave well at lower packet delivery rates for any reliability to be maintained. A review of the graphs in this report shows that the protocols have an overall sensitivity to changes in packet delivery rate as listed in Table 3, an important characteristic since packet delivery rates are time-varying. In [15] the author discusses a heuristic to determine the time bound for a route from the time-varying packet delivery rate behavior on each link. We use constant packet delivery rate for all links to simplify evaluation and believe that reliability-case values from packet delivery rates in the 60% to 90% range should represent likely overall behavior of most links. Mean values from time-weighted mean packet delivery rates are appropriate for other evaluations.

A. *Bounding Values*

The assignments of packet delivery rates to protocols were chosen to give the best range of evaluation for the available graphs and to simplify calculations. We use two bounding values for p_{ACK}, 1.0 and p_{data}. Large differences in the length of data and acknowledgment packets will show up due to Gaussian noise effects in the bit error rate (BER) and subsequent packet delivery rate, however, multipath generally dominates BER in real environments (particularly indoors), making differences much less than would be expected due to any packet length difference [1].

For the only end-to-end acknowledgment protocol and all end-to-end with link-layer acknowledgments, we use $p_{ACKPacket} = p_{data}$ for the returning end-to-end acknowledgment packet. In cases where the length of the end-to-end acknowledgment packet is much less than the length of the original data packet, then the plotted lines would be the upper boundary. On the graphs for the basic acknowledgments, the range of the only end-to-end acknowledgement is at the line and just below. On the graphs of the end-to-end

R=0.999	**PDR:**	**0.3**	**0.6**	**0.9**	**1.0**
Reliability-Case Tx-Rx Actions for 1 hop					
End-to-End		95.2	24.8	7.9	2
None		19.4	7.5	3	1
Separate		94	24	8	2
Implicit		84	20	6.5	1.5
Immediate		21	9	4	2
Reliability-Case Tx-Rx Actions for 4 hops					
End-to-End		1.5e5	1000	69.9	8
None		1200	108	22.3	4
Separate		180	47	18	8
Implicit		161	39	14	6
Immediate		43	20	12	8
Reliability-Case Tx-Rx Actions for 10 hops					
End-to-End		2.8e11	4.7e5	468	20
None		1.7e6	2.8e3	105	10
Separate		314	86	36	20
Implicit		280	71	28	28
Immediate		79	40	26	20

	PDR:	**0.3**	**0.6**	**0.9**	**1.0**
Mean Tx-Rx Actions for 1 hop					
End-to-End		14.4	4.4	2.4	2
None (R=0.999)		19.4	7.5	3	1
Separate		14.4	4.4	2.4	2
Implicit		14.4	4.4	2.4	2
Immediate		4.3	2.7	2.1	2
Mean Tx-Rx Actions for 4 hops					
End-to-End		2.2e4	146	13.2	8
None (R=0.999)		1.2e3	108	22.3	4
Separate		57.8	17.8	9.4	8
Implicit		52.8	15.3	7.7	6.5
Immediate		17.3	10.7	8.4	8
Mean Tx-Rx Actions for 10 hops					
End-to-End		4.1e10	6.8e4	72.3	20
None (R=0.999)		1.7e6	2.8e3	105	10
Separate		141	44.4	23.5	20
Implicit		129	36.9	18.5	18.5
Immediate		43.3	26.7	21.1	20

Table 2. Reliability-case and mean Tx-Rx actions for basic acknowledgments by PDR for 1, 4, and 10 hops.

acknowledgments, all plots would be affected similarly, so while the lines are also upper bounds, all would range similarly lower together.

For the link-layer protocols, there are two separate bounding cases. For Tx-Rx actions, the immediate acknowledgment protocol is a lower bound, with $p_{ACK} = 1.0$. This is not far off, as error rates for the immediate acknowledgment packet are reported to be about 1/5 the error rate for data packets. For a data packet delivery rate of 90%, 60% and 30% the acknowledgment packet delivery rate would be about 98%, 92% and 86%, respectively, though at lower data packet delivery rates this may not be accurate [12]. The upper bound for the immediate acknowledgment protocol is the line for the separate acknowledgement protocol, which is plotted with $p_{ACK} = p_{data}$. The immediate acknowledgment range is thus the area between immediate and separate acknowledgments lines, heavily weighted toward the immediate acknowledgment protocol line. The separate acknowledgment protocol does not have the advantage of a known-good channel since the acknowledgment packet is separately scheduled, thus the packet delivery rate is less. For Tx-Rx events the reasoning is the same but the bounds are different due to difference in the computation of the events. See Table 3.

Graph Bounding Regions		
ACK Protocol	**Bounds**	**Weight**
Tx-Rx Actions		
End-to-End and Link-Layer+EE	Self (as below) and lower	self
Immediate	Immediate to Separate	Heavily Immediate
Separate	Separate to Immediate	Separate
Tx-Rx Events		
End-to-End and Link-Layer+EE	Self (as below) and lower	Self
Immediate	Immediate to Implicit	Heavily Immediate
Separate	Separate and lower	Separate
Notes: No-acknowledgment and Implicit protocols have no bounding value		

Table 3. Graph Bounding Regions

B. Bandwidth

To prevent bandwidth congestion, Tx-Rx actions need to be spread evenly across motes on a route (Figure 7 and Figure 8). The relative metric we use is Tx-Rx actions as a measure the bandwidth consumed by one packet transfer. Note that in interpreting these graphs for bandwidth comparison, the Tx-Rx action values for the implicit acknowledgment protocol are slightly high because the forward/acknowledgment single transmission is one bandwidth usage but is counted as 1.5 Tx-Rx actions since there are two receives and one transmitter active.

At $PDR = 1.0$, all acknowledgments are redundant and unneeded, so the no-acknowledgment protocol has the least bandwidth usage. As the packet delivery rate drops to 0.9 the saved acknowledgment transmission of the implicit acknowledgement reveals its efficiency. However, as the packet delivery rate drops further, the requirement of the implicit acknowledgment that both the data packet and the acknowledgment packet be successful, $p = {p_{data}}^2$, causes it to lose ground to the immediate acknowledgment protocol.

While we consider the probability of delivery at each hop as independent, Figure 2 shows that effects on packet delivery rate can be network-wide. As packet delivery rates drop, more links will require greater retries, toward their reliability case values (Figure 10). The system designer must be concerned with having sufficient bandwidth to accommodate the reliability-case Tx-Rx actions on many links when necessary.

Since the failure of the transmission of packets in the no-acknowledgment and only end-to-end acknowledgment protocol follow a geometric progression, the bandwidth usage is similarly exponentially loaded toward the source, increasingly so as the packet delivery rate drops. As can be seen from Figure 9 with the Tx-Rx actions representing the relative total bandwidth usage along a route of k hops for a single packet transfer, these protocols consume an impractical amount of bandwidth as the packet delivery rate drops or the hops increase.

C. Node and Network Energy

Transmitting and receiving packets are generally the most energy consuming activates of a mote. The relative metric we use is mean Tx-Rx actions. Each unit of the metric represents one transmitter and receiver in operation for one packet. Since the implicit acknowledgment protocol performs the forward/acknowledgment in a single transmission with two receivers, each such activity is counted as 1.5 Tx-Rx actions. Many current radios specify about the same current for both transmission and reception so the mean Tx-Rx actions on the graphs can be directly translated into energy usage by multiplying by an appropriate constant.

To prevent nodes depleting their energy sources unevenly, Tx-Rx actions need to be spread evenly across motes on a route (Figure 7 and Figure 8). Nevertheless, application source and destination requirements will place burdens on specific motes. At $PDR = 1.0$, all acknowledgments are redundant and unneeded, so the no-acknowledgment protocol has the least energy usage per mote and for the route. As the packet delivery rate drops to 0.9, the saved acknowledgment transmission of the implicit acknowledgement protocol again reveals its efficiency. However, as the packet delivery rate drops further, the requirement for the implicit acknowledgment protocol that both the data packet and the acknowledgment packet be successful, $p = {p_{data}}^2$, causes it again to lose ground to the immediate acknowledgment protocol.

Figure 9 shows mean Tx-Rx actions for an entire route for k hops and they behave here as above. Since the failures of transmissions of packets in the no-acknowledgment and only end-to-end acknowledgment protocol follow a geometric progression, the energy usage is similarly exponentially loaded toward the source, increasingly so as the packet delivery rate drops. In Figure 9 with the Tx-Rx actions representing relative total energy for a single packet transfer, these protocols consume an impractical amount of energy as the packet delivery rate drops or the hops increase.

D. Resource Usage

Resource usage refers to the duration of usage measure in Tx-Rx events for two types: resources that are used during the short duration of a Tx-Rx event and those that are used for the long duration of a successful or failed delivery, possibly spanning many Tx-Rx events. The first type include items such as the availability of CPU, radio, temporary storage, program state and other items that are used while a packet is being processed. The second type is primarily packet buffers of which both mean and reliability-case values are of interest. The metric for both are units of time measured in multiples of Tx-Rx events.

Figure 11 shows the total mean Tx-Rx events for a route of k hops. For the link-layer protocols, the values are k times the mean individual usage, and represent how many times, for this packet, the CPU will likely wake up and go to sleep, the radio will be activated, etc. For the only end-to-end acknowledgment protocol, the value is usage of resources on the source mote of long duration since the source mote must wait for the acknowledgment packet from the destination before it can release the buffer with the packet. The values for the other motes on this route are of short duration distributed geometrically from the source, similar to the distribution in Figure 7. Further, while the source mote is waiting for the end-to-end acknowledgment on one packet, many other packets may also be sent. This creates packet buffer resource usage accumulating over a long duration. The longer the route in hops and the lower the packet delivery rate the more buffers can need to be in service concurrently.

As discussed above, the packet delivery rate can degrade network-wide. As packet delivery rates drop, more links will require greater retries, toward their reliability case values (Figure 12). The system designer must be concerned with having sufficient resources to accommodate the duration of reliability-case Tx-Rx events on many links when necessary.

Note that the immediate acknowledgment protocol has the lowest Mean Tx-Rx events. This implies that resources are tied up the least amount and for the shortest duration. It also implies that, other things being equal, it can achieve the highest packet transfer rate

E. Reliability

Reliability, we have defined, is the probability of delivering a packet within a given time bound. Given a proper estimate of the packet delivery rate and an adequate time bound, all of the basic acknowledgment protocols can deliver the same reliability R, within a factor for the inherent reliability of the network $\alpha, \alpha \leq 1,$ which results from extraordinary events (discussed below). That is, given a $protocol_i$ that for reliability R requires up to e_i Tx-Rx events to achieve that reliability, and we allow sufficient time to accommodate e_i events, every $protocol_i$ will succeed or fail (late or no delivery) with reliability R or αR.

The no-acknowledgment and link-layer protocols cannot overcome the inherent reliability of the network α and will deliver the packet with reliability αR. The only end-to-end acknowledgment protocol (and other end-to-end protocols) can overcome the inherent reliability of the network α and will have delivered the packet with reliability R, by designing for reliability $\frac{R}{\alpha}$ (if $\alpha \geq R$) such that the final reliability $R = \alpha \cdot \frac{R}{\alpha}$. In general, we assume $\alpha \approx 1$, as is often the case, so α can be neglected. The case of $\alpha < R$ is discussed in Extraordinary Events, below. Each of these protocols is a mechanism with differing work load requirements to perform the basic function of ensuring that the packet reaches its destination with time-bounded reliability R. If we misestimate the packet delivery rate, the effect on the available reliability varies greatly with the protocol.

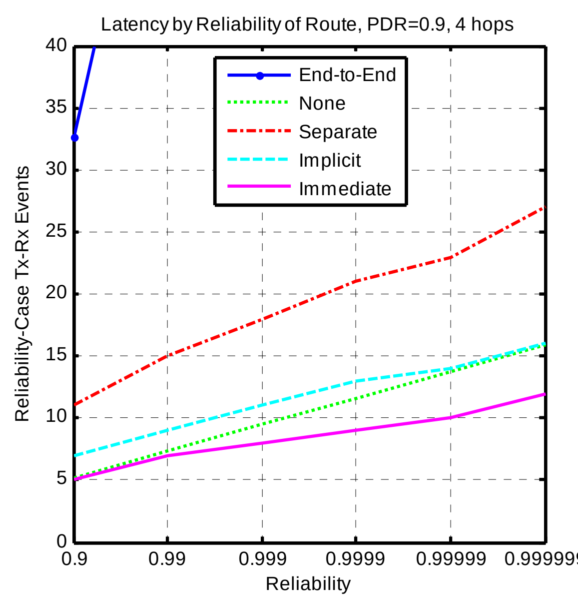


Figure 4. Tx-Rx events by basic acknowledgments for reliabilities from 0.9 to 0.999999.

Given the low packet delivery rates of wireless networks, reliability requires repeatedly transmitting a packet until it is successfully delivered. The ability of the network to do the least amount of work to deliver a packet means that network resources are available for retrying the delivery of other packets. Thus, the more Tx-Rx events that a protocol requires for a given set of conditions the less reliable the network becomes overall. Refer to Figure 4. Note that the implicit acknowledgment protocol requires 12 reliability-case Tx-Rx events for a reliability R of 0.999 at $PDR = 0.9$. For those same 12 Tx-Rx events, using the immediate acknowledgment protocol, we can achieve over two orders of magnitude greater reliability. Refer to Figure 9 and compare the mean Tx-Rx actions for implicit and immediate acknowledgements between the $PDR = 0.9$ and the $PDR = 0.6$ graphs, the area where the network will likely operate. The two protocols change places such that they are close in value within our operating range. Thus, by choosing the immediate acknowledgment protocol over the implicit acknowledgment protocol we can improve the reliability of the network at the same average energy, bandwidth, etc. Alternatively, by keeping the time bound for reliability R at 0.999, we can accommodate a higher network load and still have the resources to meet the reliability requirements.

F. Latency

To simplify the translation of Tx-Rx events to latency in seconds, the packet delivery rate used for determining Tx-Rx events (mean or reliability-case) should include the effect of CCA failures and scheduling delays. The effects should be considered delivery failures for this purpose, thus lowering the effective packet delivery rate. We would expect then that single-channel and CSMA systems could have lower effective packet delivery rates than TDMA or TSCH systems [8][18]. For similar reasons we would expect less variance in the latency in TDMA or TSCH systems.

For most CSMA architectures, and simplistically for TDMA architectures, the total latency is

$$Latency = TxRxEvents \cdot T_{hop}.$$

T_{hop} is the time interval between transmission attempts on a link. In a TDMA system, if at least one transmission can be attempted within the same frame at each hop, then latency is

$$Latency \leq k \cdot T_{frame} + (TxRxEvents - k) \cdot T_{hop}.$$

T_{frame} is the maximum time between two slots in a frame. Since $T_{frame} \leq T_{hop}$, the latency can be less in TDMA compared to CSMA systems given equal T_{hop}. Further, for critical routes, multiple TDMA slots can be allocated to a link, further decreasing the latency.

Many systems are more complex, such as those that piggyback the acknowledgement if a return data packet is to be transmitted soon; otherwise the acknowledgement is transmitted as a separate packet. Appropriate metrics to combine these models and account for the additional delays need to be devised for the behavior of each system.

Given equal values of T_{hop}, the delivery latency of systems of similar network architectures can be compared directly in units of Tx-Rx events. While at high packet delivery rates several protocols are similar in mean and reliability-case Tx-Rx events (Figure 11 and Figure 12), as packet delivery rates drop the immediate acknowledgment protocol has the clear advantage. By performing the acknowledgment immediately there are simply fewer events to schedule.

R=0.999	PDR:	0.3	0.6	0.9	1.0
Reliability-Case Tx-Rx Actions for 1 hop					
End-to-End		95.2	24.8	7.9	1
Separate+EE		126	33	12	2
Implicit+EE		113	27.5	9.5	4
Immediate+EE		29	13	7	3
4					
Reliability-Case Tx-Rx Actions for 4 hops					
End-to-End		1.5e5	1000	69.9	8
Separate+EE		271	74	30	16
Implicit+EE		242	61	23.5	12
Immediate+EE		67	34	21	16
Reliability-Case Tx-Rx Actions for 10 hops					
End-to-End		2.8e11	4.7e5	468	20
Separate+EE		512	144	63	40
Implicit+EE		456	119	48.5	48.5
Immediate+EE		133	71	49	40

	PDR:	0.3	0.6	0.9	1.0
Mean Tx-Rx Actions for 1 hop					
End-to-End		14.4	4.4	2.4	2
Separate+EE		28.9	8.9	4.7	4
Implicit+EE		28.9	8.9	4.7	4
Immediate+EE		8.7	5.3	4.2	4
4					
Mean Tx-Rx Actions for 4 hops					
End-to-End		2.18e4	146	13.2	8
Separate+EE		116	35.6	18.8	16
Implicit+EE		106	30.6	15.4	13
Immediate+EE		34.7	21.3	16.9	16
Mean Tx-Rx Actions for 10 hops					
End-to-End		4.1e10	6.8e4	72.3	20
Separate+EE		289	88.9	46.9	40
Implicit+EE		259	73.9	36.9	36.9
Immediate+EE		86.7	53.3	42.2	40

Table 4. Reliability-case and mean Tx-Rx actions for end-to-end acknowledgments by PDR for 1, 4, and 10 hops.

V. Comparison of End-to-End Acknowledgements

End-to-end acknowledgments are often used as an absolute verification of delivery and are well suited to that task. However, the reliability improvement $(1-\alpha)$ that is available is typically small, except in systems that drop packets when delivery is difficult—some systems have no retries or limit retries to as little as three [18]. The end-to-end acknowledgment can then supply the mechanism to inform of the need to retry delivery. Referring to Table 2, we find that at packet delivery rates of 0.6 and 0.3 for one hop, for the immediate acknowledgment protocol the mean number of transmissions is 2.7 and 4.3 respectively. However, to have a 99.9% probability of success, we need 9 and 21 attempts, respectively. Taken as a whole, the retries due to a missing end-to-end acknowledgment can supply the additional attempts to raise reliability, but at the great expense of an end-to-end protocol rather than a more local approach.

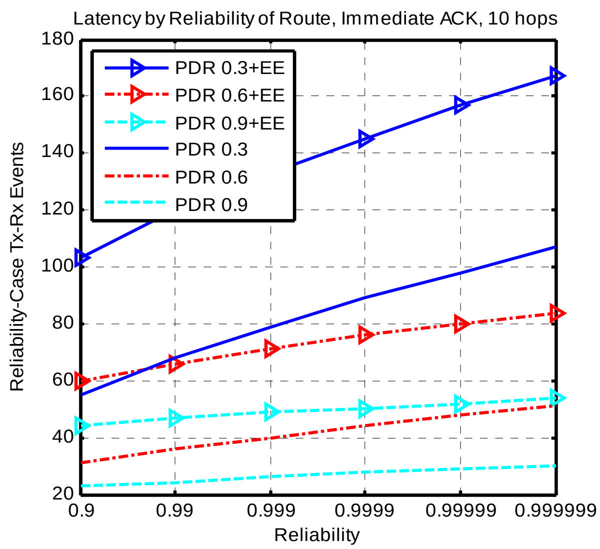
A. *Extraordinary Events*

Other than intentionally dropped packets, end-to-end acknowledgments also add delivery security against events where a packet might be successfully delivered and acknowledged at an intermediate mote on the route to the destination and then might not be forwarded. This can occur due to hardware or software bugs, node failure, link failures, unreported buffer overflows, node isolation, or other transient or permanent failures. We capture the limited reliability due to these extraordinary events in an inherent-reliability-of-the-network factor α. Generally, these events should occur very infrequently, often at much less than the typical reliability-design of the system, and far, far less frequently than delays due to normal packet delivery rates that can readily cause a packet to be delivered late. (Or, conversely, if they occur more frequently a new sensor mote vendor should be selected.) For example, in [6], over a 26 day period at an industrial site, only 17 packets out of 3.6 million generated were lost, for an inherent reliability $\alpha = 0.999995$. Time bounds were not mentioned, but the Dust Networks products used at the site essentially never stop retrying to deliver a packet so the effective time bound may have been large. From this, it can be said that inherent reliability can be high and extraordinary events can be rare.

Compared to link-layer acknowledgment protocols, end-to-end acknowledgment protocols can only add the benefit of mitigating the inherent reliability α, generally a very small additional factor of reliability. They cannot create 100% time-bounded reliability because they cannot guarantee that the acknowledgment packet will arrive within the time bound. However, in cases where the environmental or other conditions cause some extraordinary events to be ordinary, such that the events become a material failure rate relative to the desired reliability R, per-packet or selective end-to-end acknowledgments can be necessary.

B. *When to Use End-to-End Acknowledgments*

Per-packet end-to-end acknowledgments can be very expensive. An only end-to-end acknowledgment requires more mean and reliability-case Tx-Rx events than any other basic protocol reviewed under almost all conditions of PDR<1.0. Adding an end-to-end acknowledgement to a link-layer protocol is generally more efficient, but typically doubles the mean Tx-Rx actions at all packet delivery rates because twice as many hops are traversed. The effect is not as direct on reliability-case Tx-Rx actions because twice as many hops result in less than twice as many trials.



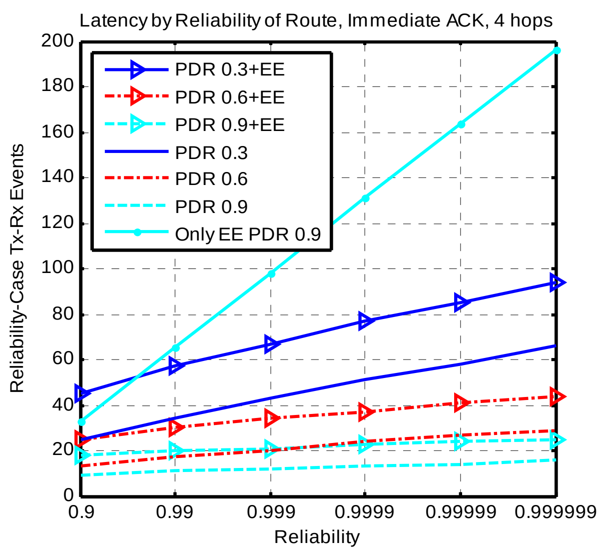


Figure 5. Tx-Rx events by end-to-end acknowledgments for reliabilities from 0.9 to 0.999999.

If the inherent reliability α is high such that $\alpha \gg R_{design}$, and reliability $R_{final} = \alpha R_{design}$ is suitable, then end-to-end acknowledgments should not be used. They will lower the overall network reliability in that resources are unnecessarily consumed and unavailable. Refer to Figure 5. At PDR=0.9+EE, four hops, and a design reliability of 0.9, it appears we can obtain five orders of magnitude more reliability for the same number of Tx-Rx events by not using end-to-end acknowledgments. At this point our reliability would likely be limited by the inherent reliability α. At ten hops the reliability improvement appears unlimited. If we start at a more reasonable value at R=0.999 then we have at least three orders of magnitude before leaving the plot.

If the inherent reliability α is low such that it encroaches on or is lower than the desired reliability R, then an end-to-end acknowledgment protocol is need. Since we cannot compute reliability greater than 1.0, we perform the computation in two layers. First we compute the reliability-case values for an end-to-end acknowledgment protocol, then we compute how many retries of those reliability-case values must be accommodated to obtain the desired reliability R. We can use equations for one hop of the only end-to-end acknowledgment protocol.

$$p = \alpha \cdot R_{EEprotocol}$$

$$b_{EEprotocol} = \frac{\log(1-R)}{\log(1-p^2)}.$$

The selected end-to-end acknowledgment protocol that had been designed (using the computations below) to deliver reliability $R_{EEprotocol}$will need accommodation for retries $b_{EEprotocol}$ to achieve desired reliability R. Since incremental increases in reliability are exponentially less expensive, the highest reliability $R_{EEprotocol}$ that is practical should be selected. Since link-layer with end-to-end acknowledgments are more efficient than the only end-to-end acknowledgment protocol, we chose one of those for the internal primary link-layer protocol.

C. End-To-End Acknowledgement with Link-Layer Acknowledgements

Adding a per-packet end-to-end acknowledgement to a link-layer protocol simply transfers a packet the same k hops back along the route from the destination to the source. While this creates additional activity, since the packets are still moved forward with a link-layer acknowledgment protocol, the average workload grows linearly rather than exponentially as it does for the only end-to-end acknowledgment protocol. The only end-to-end acknowledgment protocol requires fewer Tx-Rx actions and Tx-Rx events at extremely high packet delivery rates, as has been found on wired connections. However, this advantage drops with packet delivery rate, though there is still an advantage in Tx-Rx actions up to five hops, and Tx-Rx events against some link-layer alternatives up to three or so hops at $PDR = 0.9$. Similarly, the route congestion in Figure 13 and Figure 14 flips for the only end-to-end acknowledgment protocol being the best at four hops to the worst at ten hops.

Further, link-layer acknowledgements are far less sensitive to variations in hop depth or packet delivery rate and are thus more robust. In cases described above when an end-to-end acknowledgment is needed, it should be on top of an efficient link-layer acknowledgment protocol (Figure 17, Figure 18). While the only end-to-end acknowledgment fares better against the link-layer protocols with end-to-end acknowledgment than to the basic acknowledgment protocols, the advantage quickly dies with increased hops or decreased packet delivery rate.

Thus, an end-to-end acknowledgment with link-layer acknowledgement results in better performance [2], so long as the timing for retries is adequate so that the behaviors of the two layers do not interact poorly [11]. However, all end-to-end protocols create significant buffer demands at the source mote that do not exist with only link-layer acknowledgments. The latencies are quite long, as described above, and may entail many buffers simultaneously as several end-to-end transactions may be in process. Thus, unless reliability R exceeding the inherently reliability α is required, end-to-end acknowledgments add significant stresses to the network that may elicit failures and lower the inherent reliability α rather than increase overall reliability.

1) Calculations

End-to-end acknowledgments with link-layer acknowledgments perform identically to their non-end-to-end counterparts, except the return acknowledgment packet doubles the number of hops traversed. For all link-layer acknowledgment calculations with hops k, apply $2k$.

2) Bandwidth

As discussed above, the metric for bandwidth is mean Tx-Rx Actions (Figure 15). The only end-to-end acknowledgment protocol uses less bandwidth up to a few hops, on average, than the end-to-end acknowledgment with link-layer acknowledgment protocols, but this advantage is exponentially eliminated and is unevenly distributed along the route (Figure 13) as the packet delivery rate drops.

3) Node and Network Energy.

The same patters already discussed follow here. There are narrowly bounded conditions where the only end-to-end acknowledgment consumes less energy than using an end-to-end acknowledgment with a link-layer-acknowledgment.

4) Resource Usage

Using an end-to-end acknowledgment with a link-layer acknowledgment now causes all the protocols to have the same long-term packet buffers on the source mote. However, the mean and reliability-case durations are significantly less in most cases for the link-layer acknowledgment alternatives (Figure 17, Figure 18). This implies that by using the link-layer alternatives, potentially fewer extraordinary events, such as unreported packet buffer exhaustion, occur, and thus the number of end-to-end acknowledgment retries should be reduced.

To a degree, the exponential resource usage duration of the only end-to-end protocol can be mitigated by unconditionally retransmitting packets from the source rather than waiting for the failed end-to-end acknowledgment. However, while the mean Tx-Rx actions become comparable to the link-level alternatives at high packet delivery rates, as the packet delivery rates drop exponential growth again occurs and any benefit is lost after a few hops. In the reliability-case, the exponential growth is slower, but still exponential and exceeds the link-level alternatives after a few hops.

5) *Reliability*

As discussed above, there are well defined instances when an end-to-end acknowledgment is required to meet application reliability requirements. If the reliability requirements do not need an end-to-end acknowledgment protocol, then much higher reliability is available from the basic link-layer protocol for the same number of Tx-Rx events (Figure 5). When needed, using an end-to-end acknowledgment with a link-layer acknowledgment protocol consumes far fewer Tx-Rx events in almost all cases than the only end-to-end acknowledgment protocol for reasons previously discussed (Figure 16). In all cases, the end-to-end acknowledgment will allow overcoming the inherent reliability α limitation.

6) *Latency*

Latency follows the same behavior as Resource Usage and Latency previously discussed.

VI. Cumulative or Selective Acknowledgments

Either cumulative or selective acknowledgments can reduce the overhead for the end-to-end acknowledgment packet by reducing the number that is transmitted. While these acknowledgments were not specifically modeled for this report, we believe sufficient data is available to glean their first order behavior.

Cumulative or selective acknowledgments are often used as a replacement for per-packet only end-to-end acknowledgments. As such, the packet goes out toward the destination with no link-layer acknowledgments. If the ratio of data packets to acknowledgments is high, the closest model to the mean Tx-Rx actions of data packet transmissions is the only end-to-end acknowledgment protocol at one-half the hop distance. We use one half because we want just to evaluate the Tx-Rx actions to reach the destination but not to return the end-to-end acknowledgment. We want to determine the activity to get a packet to the destination, but assume an infrequent and yet uncounted acknowledgment packet to indicate which packets will need to be retransmitted.

Refer to Figure 9. For $PDR = 0.9$ at five hops (ten equivalent for our comparison), the only end-to-end acknowledgment protocol is comparable to the link-layer protocols at ten hops. Thus, at very high packet delivery rates as seen in the first half of Figure 2, selective or cumulative acknowledgments may be an advantage. However, as the packet delivery rate drops toward 0.6 in the next plot, the number of Tx-Rx actions for our new protocol is off the plot. At up to two hops, the situation is more comparable, but there is no definite win for the new protocols. Clearly, operating in the expected packet delivery rate range the new protocol is comparable or not efficient compared to a per-packet end-to-end acknowledgment with a link-layer acknowledgment.

However, using cumulative or selective acknowledgments with link-layer acknowledgments would be very efficient, if an end-to-end acknowledgment were necessary. The inherent reliability α of the network could be overcome with fewer resources used compared to per-packet end-to-end acknowledgments, aside from the packet buffers required until successful delivery.

VII. Conclusions

Careful evaluation of each wireless system is necessary to determine the time bounds required to meet the desired delivery reliability. While many acknowledgment protocols are in use, few perform well across a range of network conditions. The best in each area are summarized in Table 5 and Table 6, for R=0.999, with changes for higher and lower reliabilities. The no-acknowledgment protocol is suitable for use when the packet delivery rates are high, or required reliabilities are low, but otherwise is more expensive than alternatives. Its simplicity does not save resources outside of those ranges. The only end-to-end acknowledgment protocol fares well at very high packet delivery rates and few hops, as it would in a wired environment. Otherwise, it, too, quickly becomes too expensive. Of the link-layer protocols, the implicit acknowledgment protocol fares well briefly at high packet delivery rates, but, in general, the immediate acknowledgment protocol performs best overall, even when the bounded immediate range is reviewed. It also is the most robust with the lowest sensitivity to changes in packet delivery rate. As Figure 6 shows, packet delivery rates on many links drop well below the average for enough of the time that the lower packet delivery rates will dominate the packet delivery rate for those links, and thus routes that include those links.

While link-layer protocols typically operate independently from higher-layer protocols, that does not mean that the higher-layer protocol should operate unaware of the link-layer behavior. Ignoring actual link-layer behavior while considering parameters such as delivery timing is not link-layer agnostic, but the attempted ignorance itself is assuming some generic model and can lead to unfortunate interactions between the layers that reduce goodput. Studies that find layer interaction problems are typically looking at the results of using the TCP, which has significant mechanisms that assume the characteristics and reliability of a wired connection at the lower layers. When not taken into account, inappropriate timing interactions between the layers then generates degraded goodput results [7][11]. Inter-layer awareness and cooperation to prevent redundant operations has been found to significantly improve goodput [3][11], but this must be carefully abstracted to allow the architecture to evolve and prevent obsolesce [13] .

Arbitrarily adding end-to-end acknowledgments to packet deliveries without understanding the underlying link-layer reliability can reduce overall network reliability by increasing resource usage and limiting resource availability with little, if any, gain in reliability. Only if the inherent reliability α of the network is low should end-to-end acknowledgments be considered, and then only on top of the most efficient link-layer acknowledgment, which we find to be immediate acknowledgments. Even more efficient are cumulative or selective acknowledgments when resources are available to hold the packets that may need to be retransmitted. These appear efficient on a link-layer acknowledgment protocol, and still should be used only when necessary, i.e. because inherent reliability α is low.

In general, reliability is obtained by accommodating the time required for sufficient retransmissions to achieve the desired reliability. In this report we show how to determine the required reliability, how to determine the required time bound given a sufficient estimate of the packet delivery rate behavior, metrics to select the best type of acknowledgment for the conditions, and reasoning for when end-to-end acknowledgments are required and how to improve reliability up to the inherent limit of the network and above.

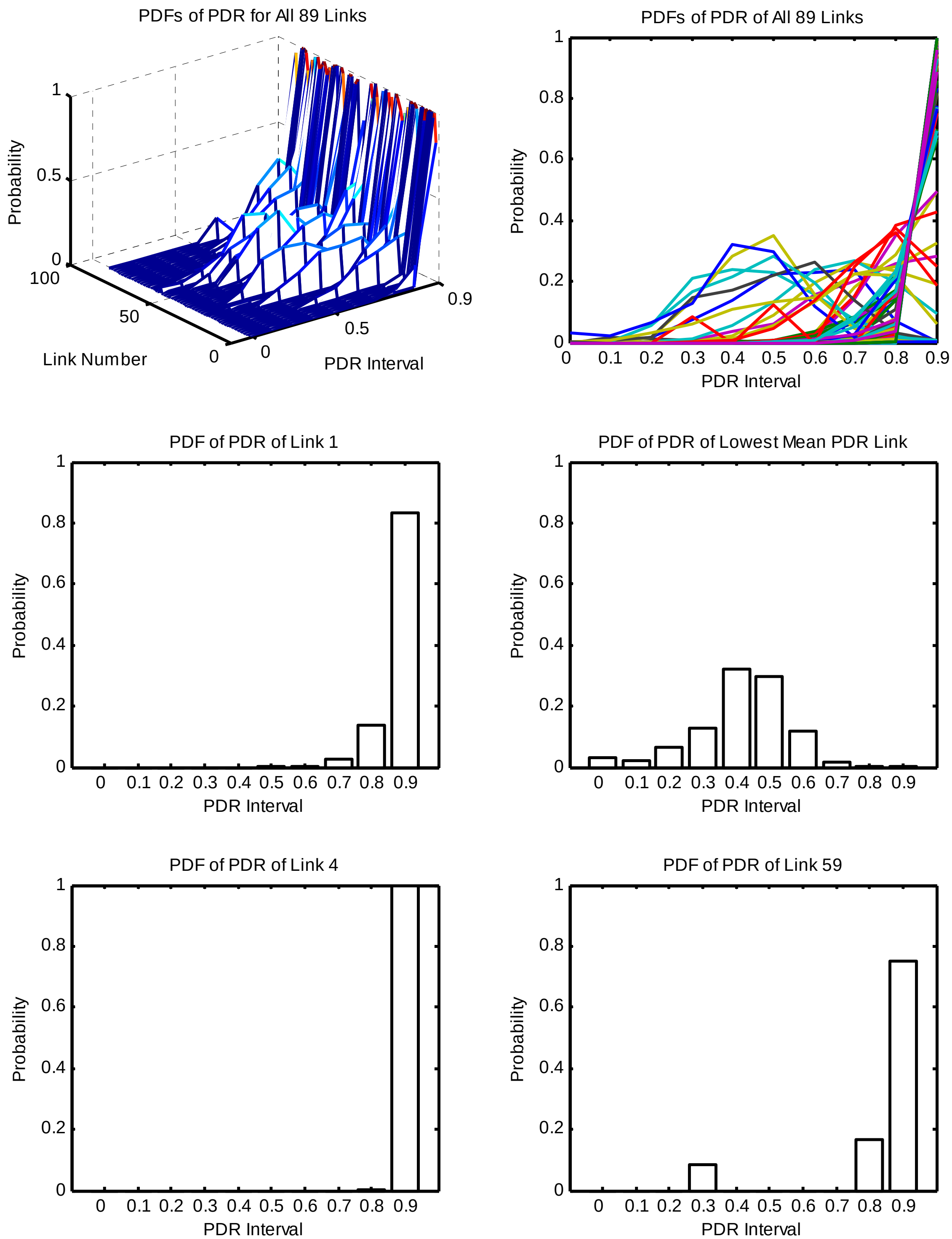


Figure 6. Time-varying packet delivery rate probability distributions. To achieve even reasonable reliability, mid-valued PDRs on many links and the corresponding larger number of retransmission must be accommodated in the time bound.

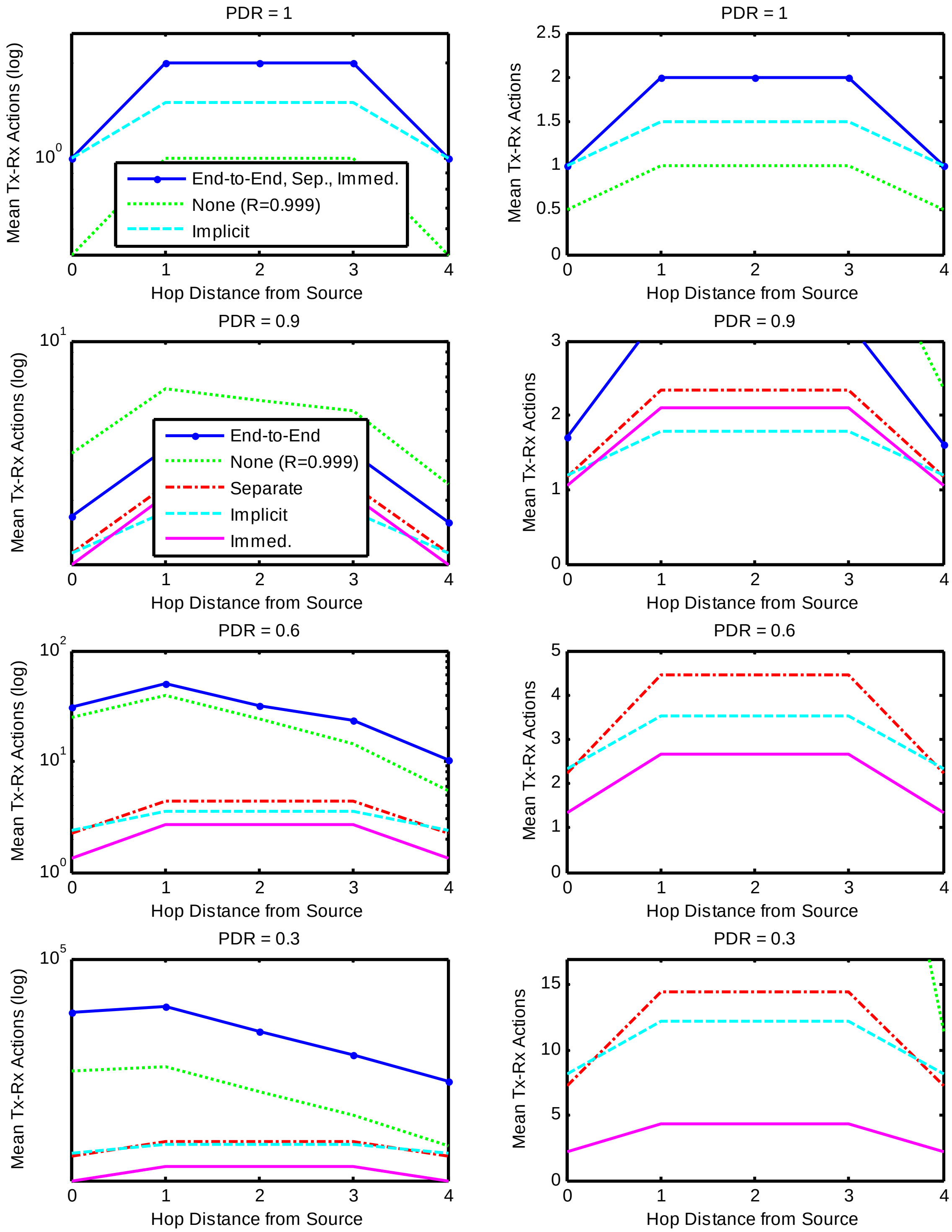


Figure 7. Mean Tx-Rx actions by basic acknowledgment type at each hop on a linear route of 4 hops. Tx-Rx actions are a metric for bandwidth and energy usage.

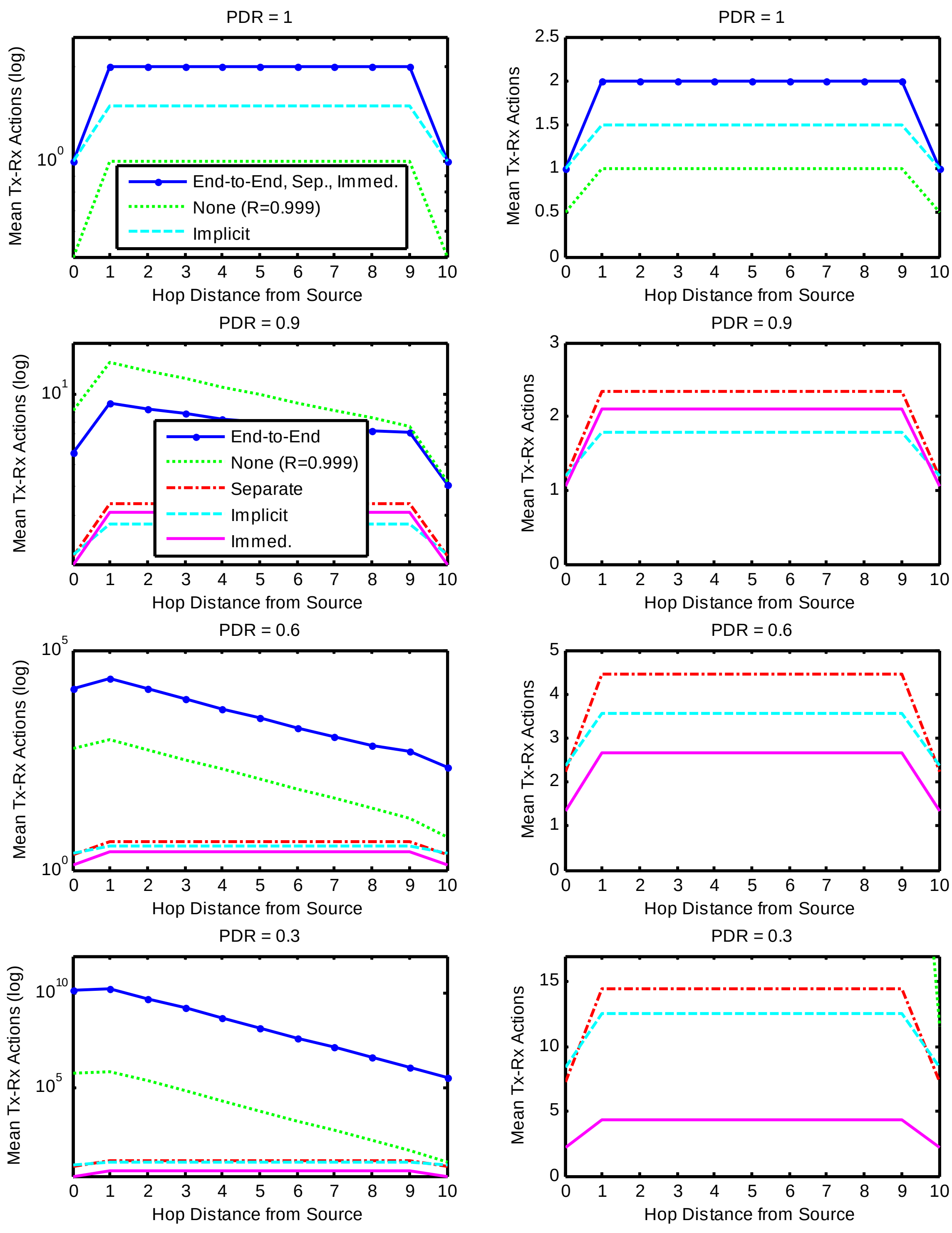


Figure 8. Mean Tx-Rx actions by basic acknowledgment type at each hop on a linear route of 10 hops. Tx-Rx actions are a metric for bandwidth and energy usage.

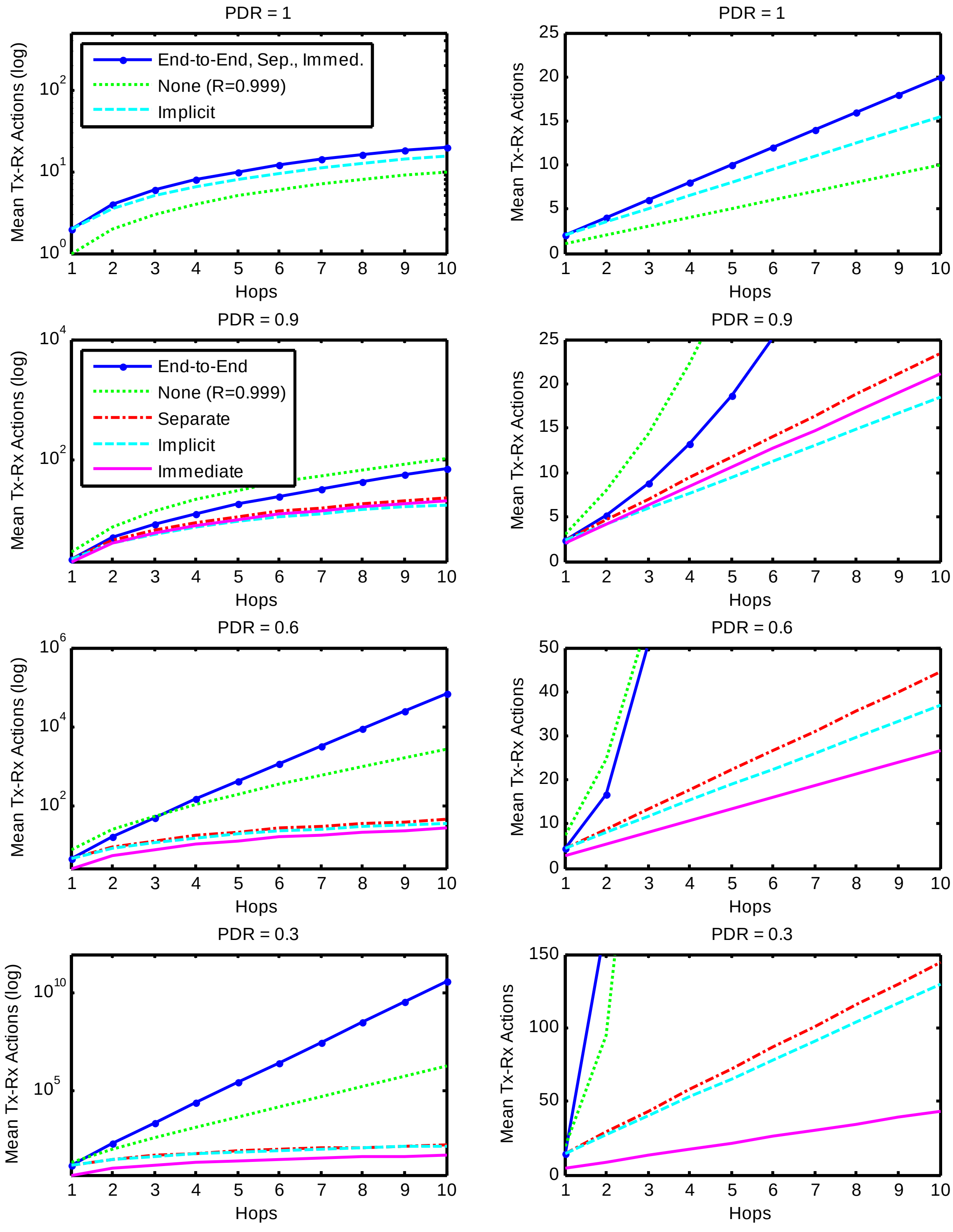


Figure 9. Mean Tx-Rx actions by basic acknowledgment type for 1 to 10 hops. Tx-Rx actions are a metric for bandwidth and energy usage.

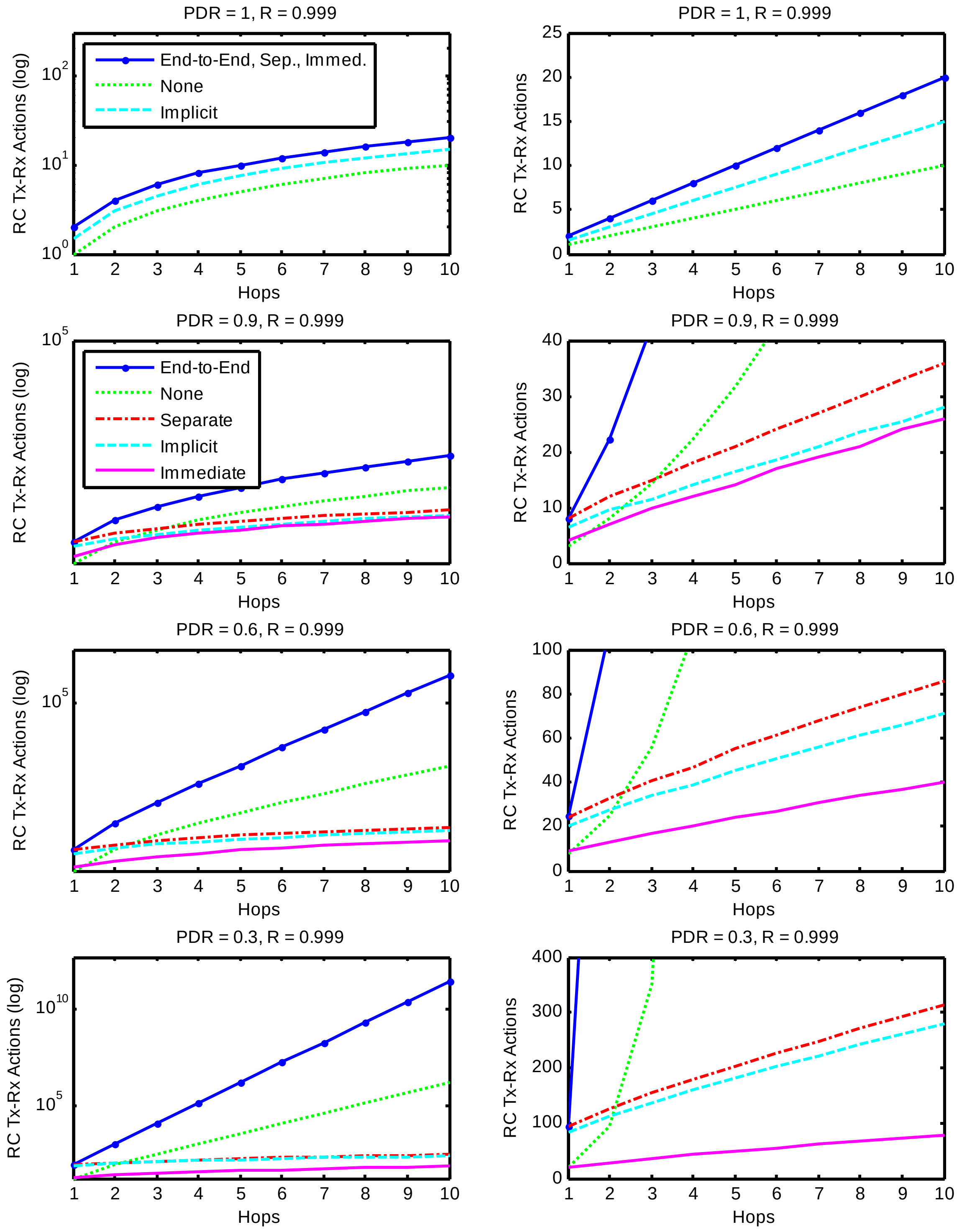


Figure 10. Reliability-case Tx-Rx actions by basic acknowledgment type for 1 to 10 hops and R=0.999. Reliability-case values indicate the activity that must be accommodated to met the desired reliability.

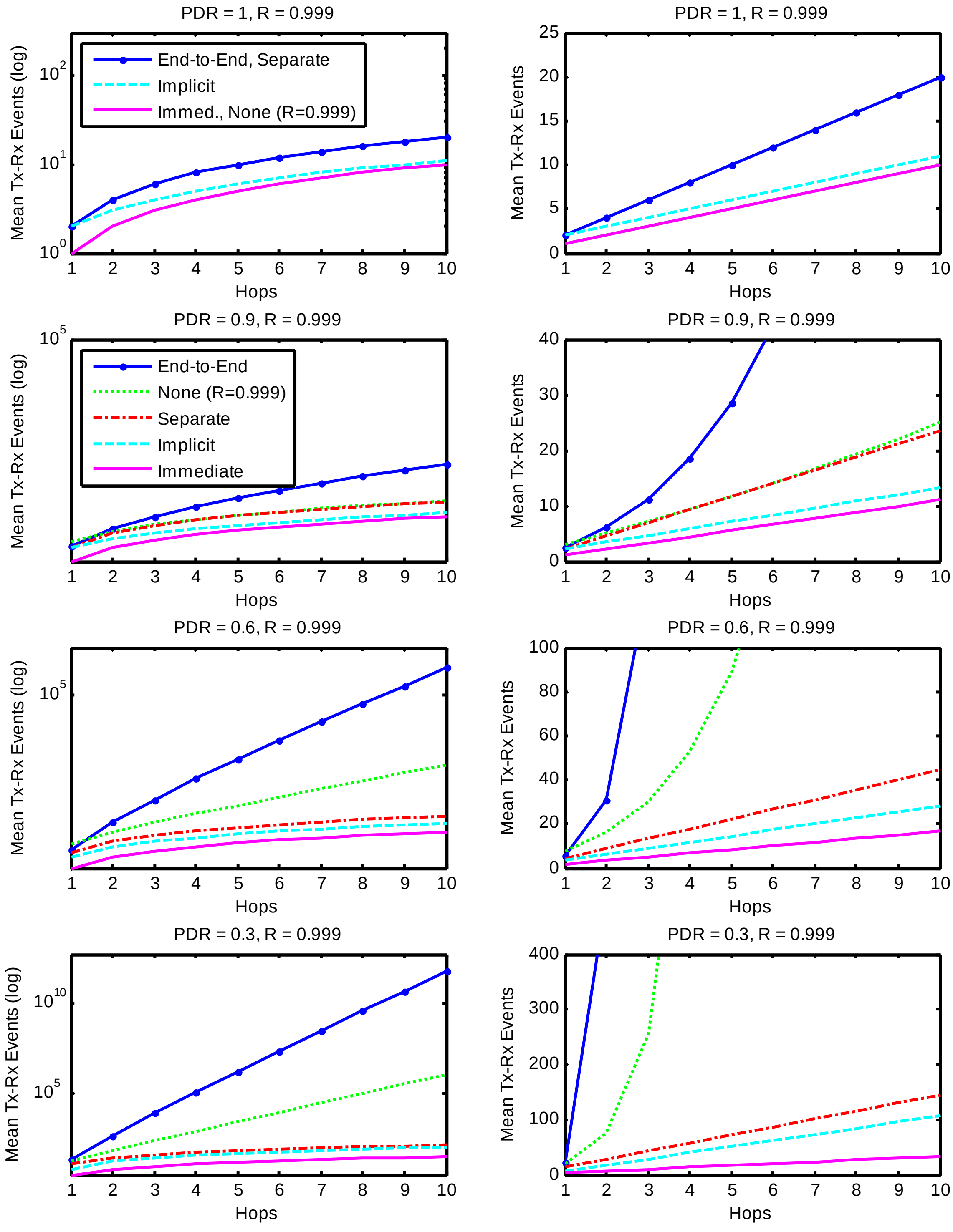


Figure 11. Mean Tx-Rx events by basic acknowledgment type for 1 to 10 hops. Mean Tx-Rx events are a metric for resource usage duration and latency.

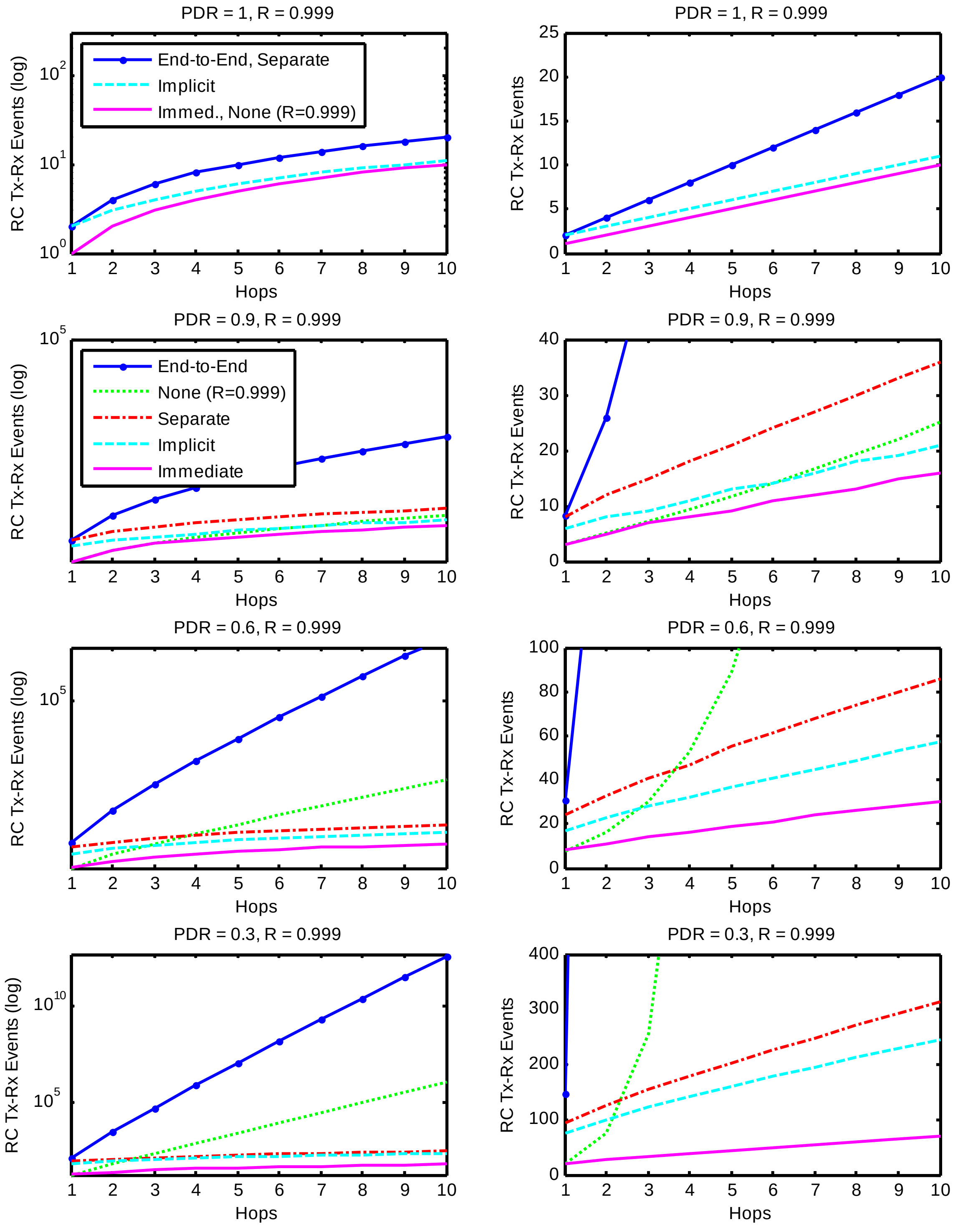


Figure 12. Reliability-case Tx-Rx events by basic acknowledgment type for 1 to 10 hops and R=0.999. RC Tx-Rx events are a metric for resource usage duration, reliability time-bound, and latency.

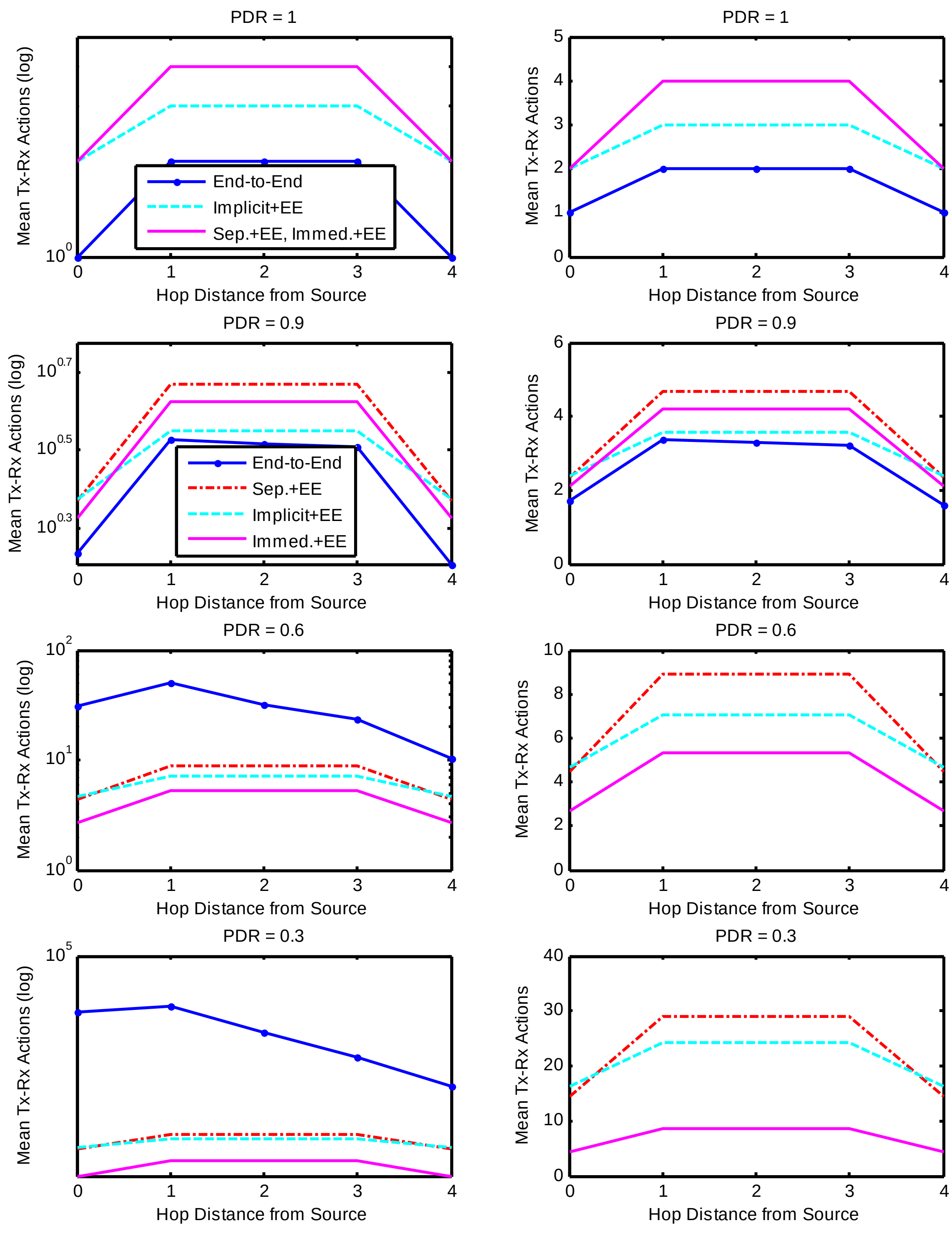


Figure 13. Mean Tx-Rx actions by end-to-end acknowledgment type at each hop on a linear route of 4 hops. Tx-Rx actions are a metric for bandwidth and energy usage.

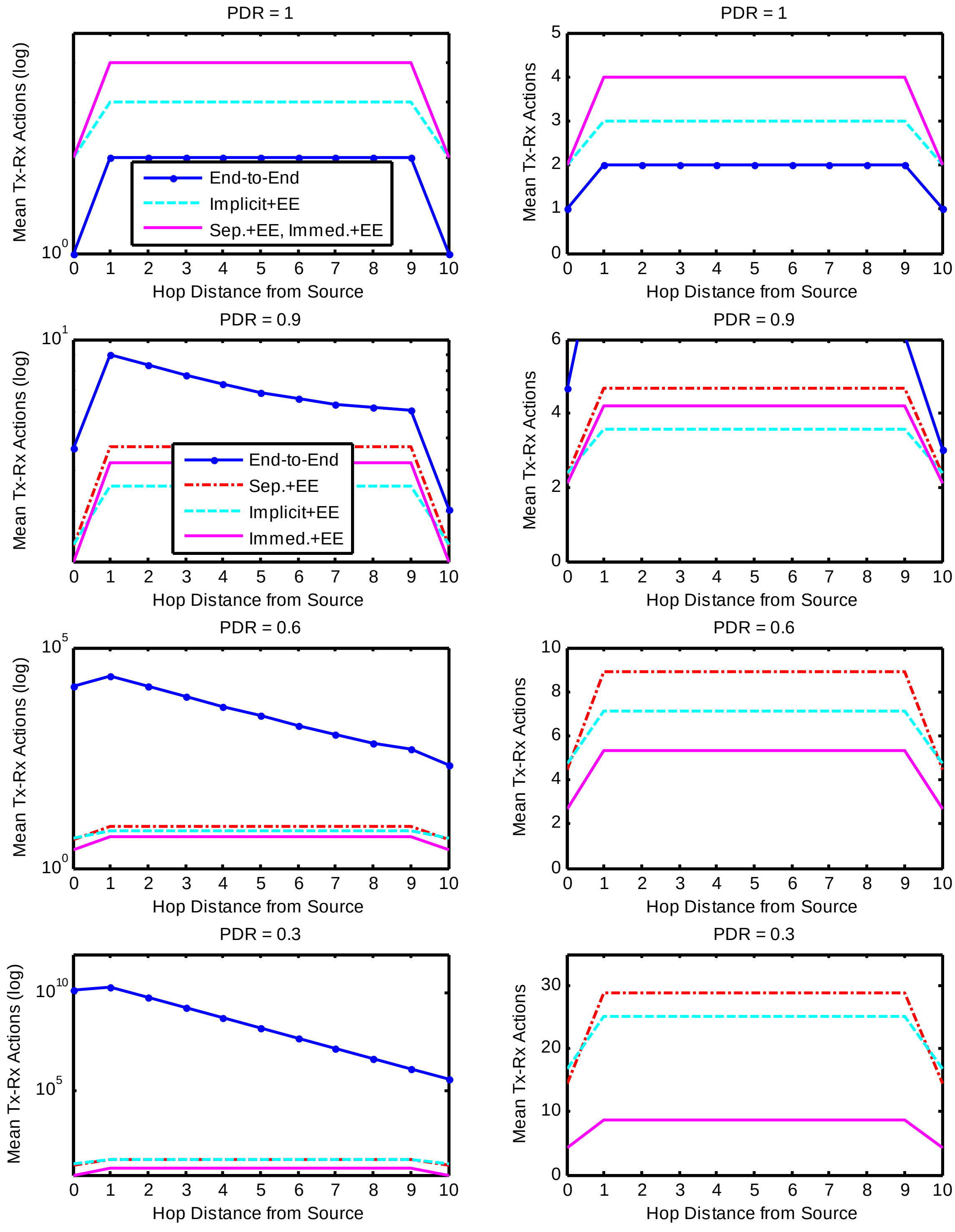


Figure 14. Mean Tx-Rx actions by end-to-end acknowledgment type at each hop on a linear route of 10 hops. Tx-Rx actions are a metric for bandwidth and energy usage.

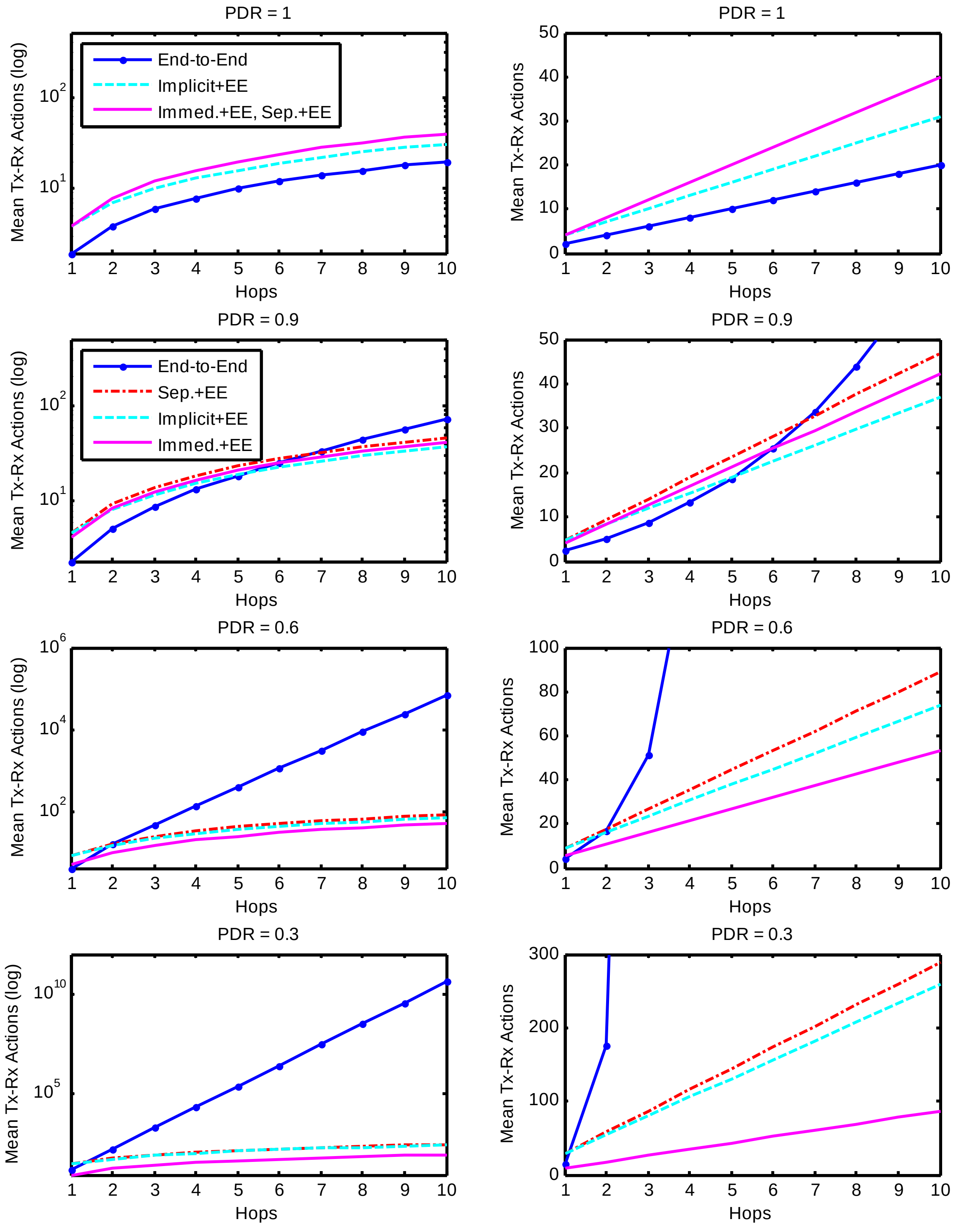


Figure 15. Mean Tx-Rx actions by end-to-end acknowledgment type for 1 to 10 hops. Tx-Rx actions are a metric for bandwidth and energy usage.

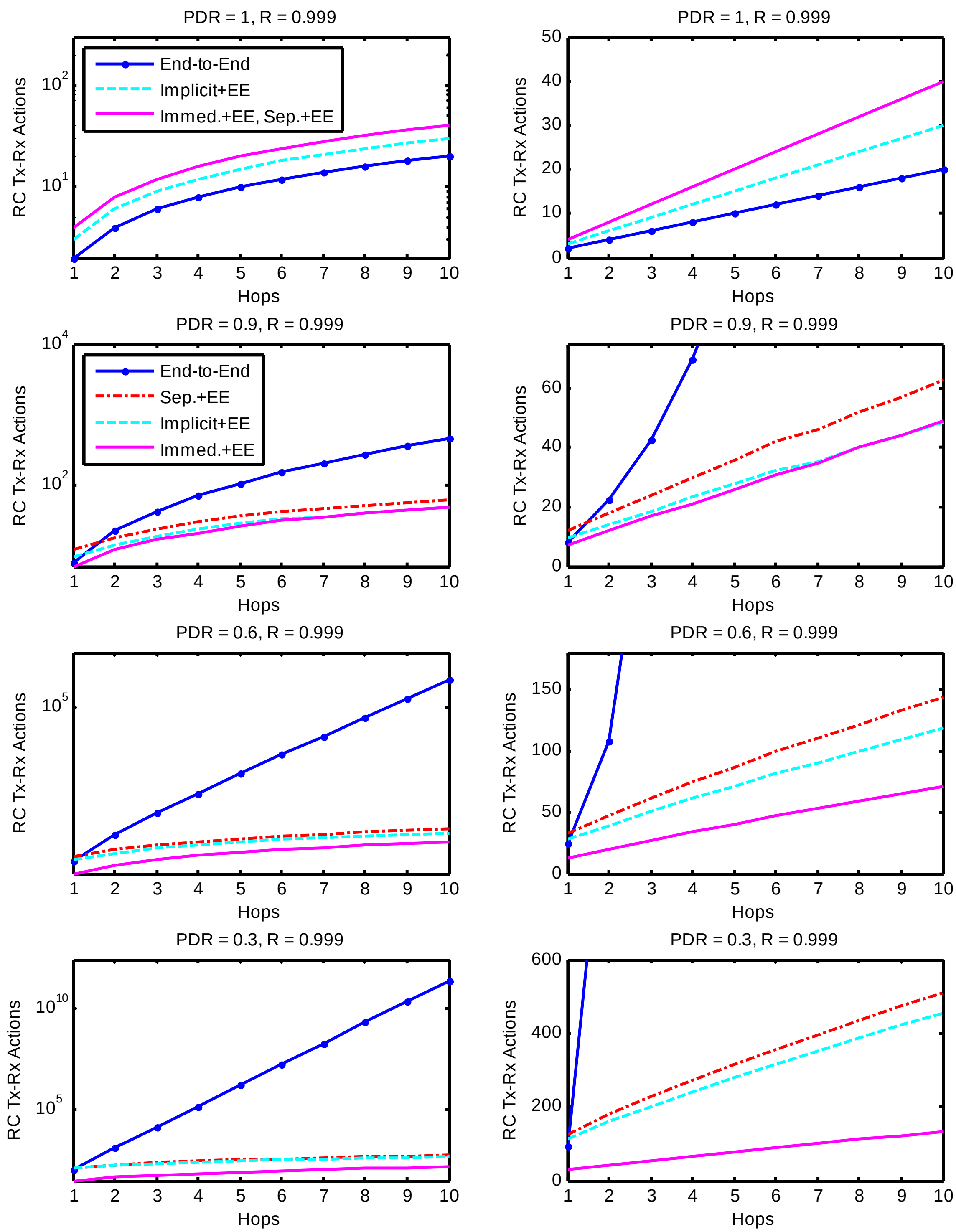


Figure 16. Reliability-case Tx-Rx actions by end-to-end acknowledgment type for 1 to 10 hops and R=0.999. Reliability-case values indicate the activity that must be accommodated to met the desired reliability.

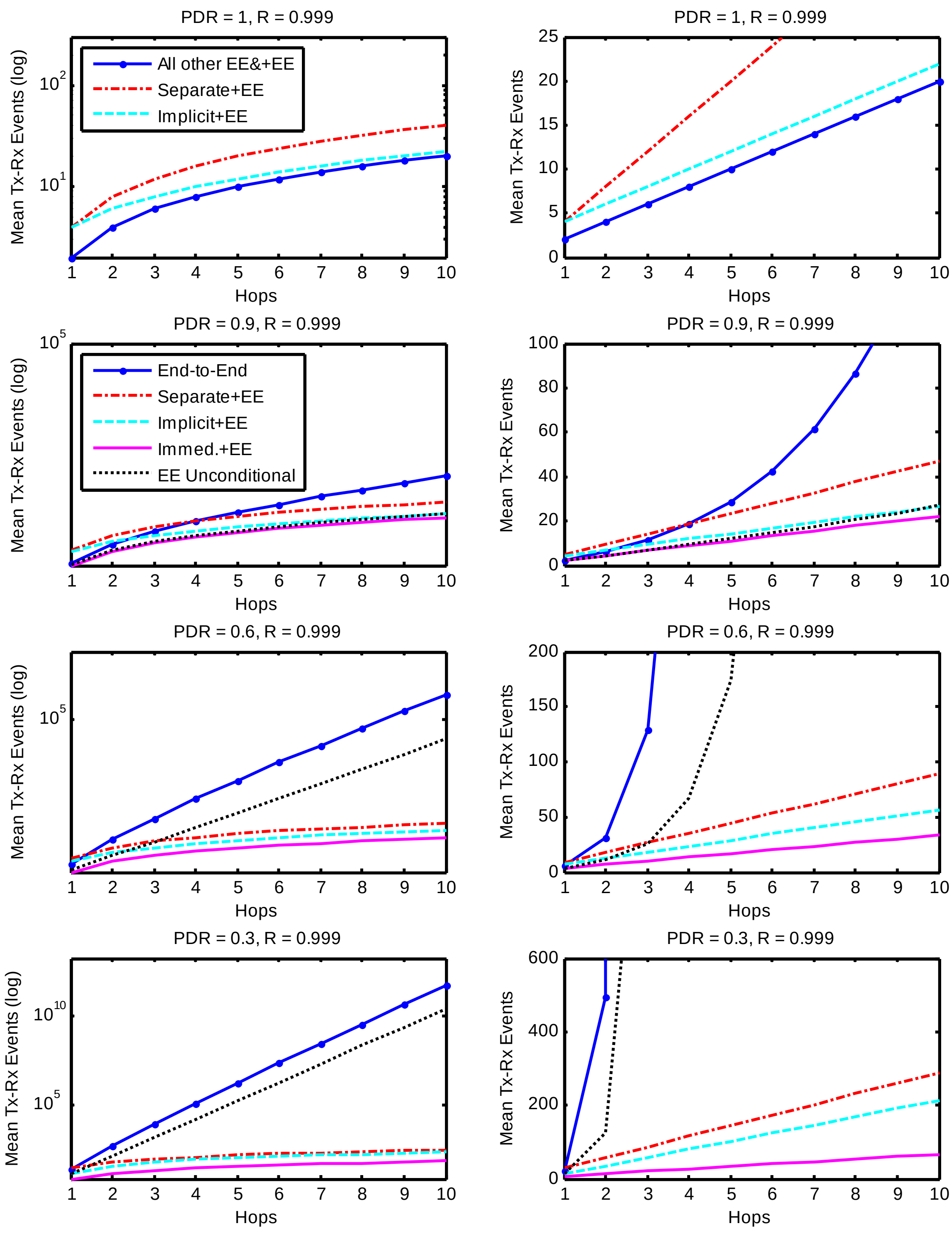


Figure 17. Mean Tx-Rx events by end-to-end acknowledgment type for 1 to 10 hops. Mean Tx-Rx events are a metric for resource usage duration and latency.

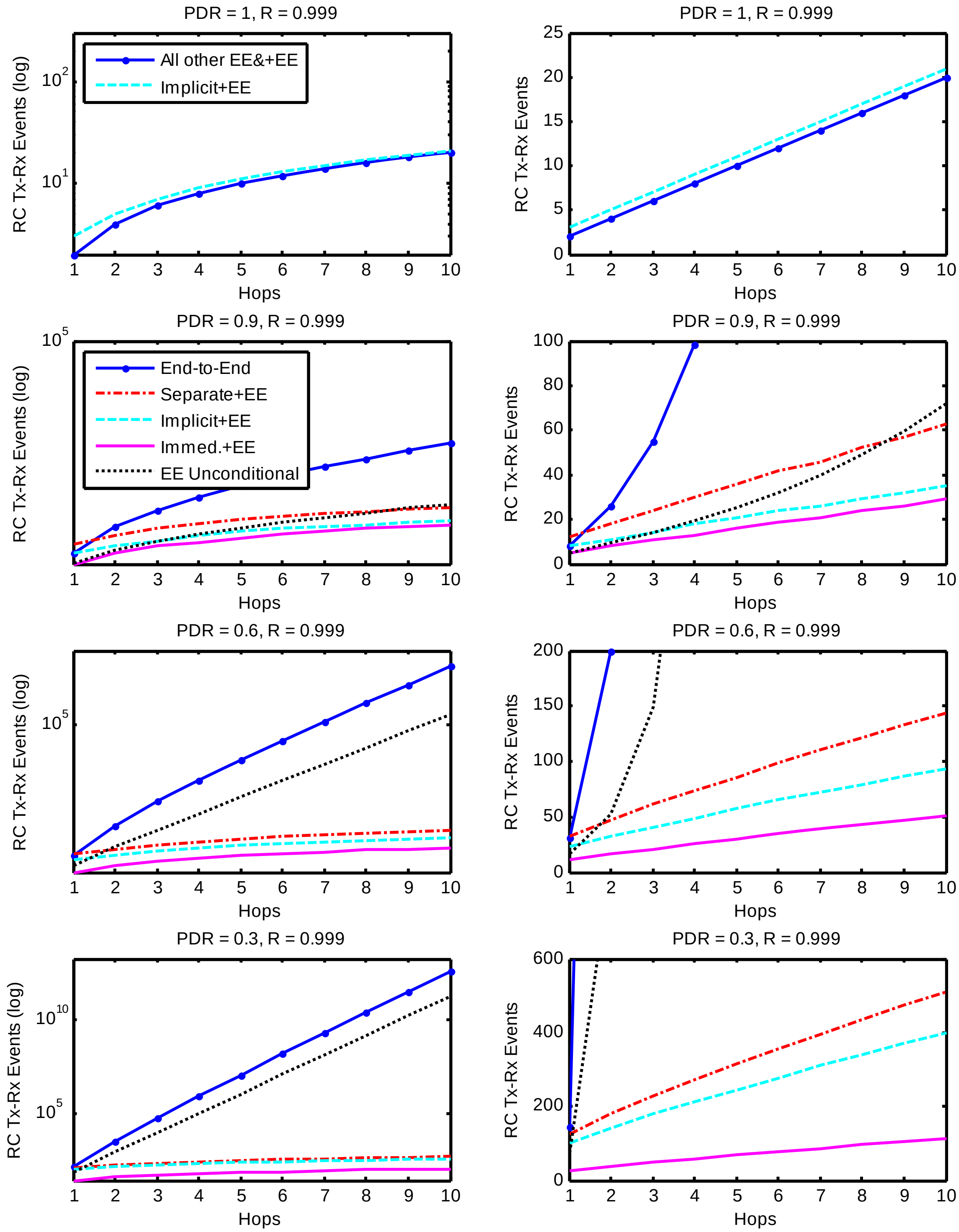


Figure 18. Reliability-case Tx-Rx events by end-to-end acknowledgment type for 1 to 10 hops and R=0.999. RC Tx-Rx events are a metric for resource usage duration, reliability time-bound, and latency.

Basic Acknowledgments Summary									
	Changes at R=0.9			**Compared to R=0.999**			**Changes at R=0.99999**		
	Lowest Mean Tx-Rx Actions								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			No ACK	All		same		
0.9	No ACK	1 to ~4	Implicit	Implicit/all	All				
Lower	same			Immediate	All				
	Lowest Reliability-Case Tx-Rx Actions								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			No ACK	All		same		
0.9	No ACK	1 to ~6	Implicit/ Immediate	No ACK	1 to ~2	Immediate	Immediate	All	
Lower	same			Immediate	All		same		
	Lowest Mean Route Congestion								
PDR	**4 Hops**	**10 Hops**		**4 Hops**	**10 Hops**		**4 Hops**	**10 Hops**	
~1.0	same			No ACK	No ACK		same		
0.9	No ACK/ Implicit/ Immediate	same		Implicit/ Immediate	Implicit/ immediate				
Lower	same			Immediate	immediate				
	Lowest Mean Tx-Rx Events								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			Immediate/ No ACK	All		same		
0.9				Immediate	All				
Lower				Immediate	All				
	Lowest Reliability-Case Tx-Rx Events								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			Immediate	All		same		
0.9				Immediate	All				
Lower				Immediate	All				
Notes: Protocol1/Protocol2… means that the two are very close									

Table 5. Summary of basic acknowledgments and changes that occur at higher and lower reliabilities.

End-to-End Acknowledgments Summary									
	Changes at R=0.9			Compared to R=0.999			Changes at R=0.99999		
	Lowest Mean Tx-Rx Actions								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			End-to-End	All		same		
0.9				End-to-End	1-~5	Implicit/All			
Lower				Immediate	All				
	Lowest Reliability-Case Tx-Rx Actions								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			End-to-End	All		same		
0.9	End-to-End	1 to ~3	Implicit	Immediate/ Implicit	All		Immediate	All	
Lower	same			Immediate	All		same		
	Lowest Mean Route Congestion								
PDR	**4 Hops**	**10 Hops**		**4 Hops**	**10 Hops**		**4 Hops**	**10 Hops**	
~1.0	same			End-to-End	End-to-End		same		
0.9				End-to-End/ Implicit	Implicit/ immediate				
Lower				Immediate	immediate				
	Lowest Mean Tx-Rx Events								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			Any except Separate, Implicit	All		same		
0.9				Immediate /Unc EE	1 to ~6	Immediate			
Lower				Immediate	All				
	Lowest Reliability-Case Tx-Rx Events								
PDR	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**	**Which**	**Hops**	**Then**
~1.0	same			Any except Implicit	All		same		
0.9	UC-EE	1 to ~4	Immediate	Immediate	All				
Lower	Imediate			Immediate	All				

Notes:
1. Protocol1/Protocol2… means that the two are very close
2. “UC-EE” means Unconditional End-to-end
3. Immediate, Separate, Implicit all imply +EE for this table only.

Table 6. Summary of end-to-end acknowledgments and changes that occur at higher and lower reliabilities.

REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, R. Morris, “Link-Level Measurements from an 802.11b Mesh Network,” in *Proc. SIGCOMM ’04*, Portland, Oregon, Aug. 30-Sep. 3, 2004.

[2] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, R. D. Gitlin, “AIRMAIL: A link-layer protocol for wireless networks,” *ACM ACM/Balzer Wireless Networks J*, vol. 1, pp. 47-60, February 1995.

[3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A Comparison of Mechanisms for Improving TCP Performance over Wireless Links,” in *Proc. ACM SIGCOMM ’96*, Stanford, CA, August 1996.

[4] A. Cerpa, N. Busek, and D. Estrin, “Scale: A tool for simple connectivity assessment in lossy environments,” *Technical Report 0021*, Center for Embedded Networked Sensing, Univ. of California, Los Angeles, Sept. 2003.

[5] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing,” in *Proc. MobiCom ’03*, San Diego, CA, September 2003.

[6] L. Doherty, W. Lindsay, J. Simon, “Channel-Specific Wireless Sensor Network Path Data,” in *Proc. ICCN 2007*, Honolulu, Hawaii, August 2007.

[7] A. DeSimone, M. C. Chuah, and O. C. Yue, “Throughput Performance of Transport-Layer Protocols over Wireless LANs,” in *Proc. Globecom ’93*, Houston, TX, December 1993.

[8] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, M. Gerla, “The Impact of Multihop Wireless Channels on TCP Performance,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 209-221, Mar/Apr, 2005

[9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, “Complex behavior at scale: An experimental study of low powered wireless sensor networks,” Technical Report UCLA/CSD-TR 02-0013, 2002.

[10] D. Messina, M. Ortolani, and G. Lo Re, “Achieving Robustness through Caching and Retransmission in IEEE 802.15.4-based WSNs,” in *Proc. ICCCN 2007*, August 2007.

[11] C. Parsa, J. J. Garcia-Luna-Aceves, “TULIP: A Link-Level Protocol for Improving TCP over Wireless Links,” in *Proc. WCNC 1999*, New Orleans, LA, September 1999.

[12] K. Pister, University of California, Berkeley, CA, private communication, October 2008.

[13] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, I. Stoica, “A Unifying Link Abstraction for Wireless Sensor Networks,” in *Proc. SenSys ’05,* San Diego, CA, November 2005.

[14] J.H. Saltzer, D.P. Reed, D.D. Clark, “End-to-End Arguments in System Design,” ACM Transactions in Computer Systems 2, 4, November, 1984, pp. 277-288.

[15] G. W. Shaw, “Determining Hard Time Bounds for Reliability Delivery in Wireless Sensor Networks,” unpublished.

[16] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, “Some Implications of Low Power Wireless to IP Networking,” in Proc*. HotNets-V*, Irvine, California, November 2006.

[17] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, M. Gerla, “The Impact of Multihop Wireless Channels on TCP Performance,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 209-221, Mar/Apr, 2005

[18] A. Woo, T. Tong, D. Culler “Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks,” in *Proc. SenSys ’03*, Los Angeles, CA, November 2003.

[19] J. Zaho, R. Govindan, “Understanding Packet Delivery Performance In Dense Wireless Sensor Networks,” in *Proc. SenSys ’03*, Los Angeles, CA November 2003.