RESEARCH ARTICLE OPEN ACCESS

A Reliable Transport Protocol for Code Distribution in Large Sensor Networks

Rajanikanta Sahu, Sambit Kumar Sethi,
Gandhi Institute of Excellent Technocrats, Bhubaneswar,India
Subas Institute of Technology, Bhubaneswar, Odisha, India

ABSTRACT—Re-taskingandremoteprogrammingofsensornetworks is an essential functionality to make these networkspractical and effective. As the availability of more capable sensornodesincreasesandnewfunctionalimplementationscontinuetobeproposed,theselargecollectionsofwirelessnodeswillneed the ability to update and upgrade the software packagesthey are running. Standard flooding mechanisms are too energy-costlyandcomputationallyexpensiveandtheymayinterferewiththe network's current tasks. A reliable method for distributingnewcodeorbinaryfilestoeverynodeinawirelesssensornetwork is needed. This paper proposes a more effective method,called PALER (Push Aggressively with Lazy Error Recovery),which builds upon the previously proposed PSFQ protocol [1], areliabletransportprotocolwhichslowlypacesthepropagationoffilesegments,butusesanaggressivelocalrecoverymethodtoavoidpacketimplosionduetolosspropagation.PALERusesamoreaggressivepushingmechanismandreducesthe recovery mechanism to a single inclusive NACK. Furthermore,PALER uses local neighbor information to reduce redundanttransmissions.Thispaperstudiesthisnewprotocol'senergyefficiency and shows that it scales well to higher densities andfieldsizes.

I. INTRODUCTION

Wirelesssensornetworksareacollectionofsmallsensordevices,typicallybatterypowered,thatmaybestaticormobile,andmayconfigureinanadhocfashion.Theyhaveinspiredagreatdealofresearchintothedesignandimplemen-tationofsuchnetworks,suchasroutingmechanisms,eventtriggering,andclusteringmethods,aswellasmethodsofimprovingenergyefficiencywithradiopowercontrolmecha-nismsandadaptivefunctionality.Re-taskingsensornetworkswhichhavealreadybeendeployedisstillanopenareaofresearch. It requires the reliable distribution of a binary file orcode toallof thenodeswhichrequire thenewprogram, andacoordinatedmethodofloadingthenewprogramamongallofthe nodes. This can be an expensive task, requiring significantbatterypowertofulfillallnecessaryradiotransmissions,andconsuming the computational resources of the wireless nodes.Inthispaper,wewillfocusonthetaskoffiledistributionin support of remote programming. There have been a numberofproposalsforsuchafunctionality.Manyofthesehaveworkedundertheassumptionofasmallbinaryorcodefile,requiring a limited number of segments to be distributed. Thisconditionwastypicalofearlywirelesssensormotes,whichhavelimitedmemorycapacity,butasthecapabilityofsensormotesincreases,sensornetworkprograms willinevitablyincrease with them. In this paper, we will explore a moreefficient method of binary or code distribution for relativelylargedatafiles.Themostbasicmethodoffiledistributionis by flooding the segments into the sensor network. Thismethod is very costly, though, as every node will receive andbroadcasteachsegmentofafile.Manyofthesebroadcastsareunnecessary since sensor networks may be dense, and havehighly overlapping broadcast zones. The redundant transmis-sions are an unnecessary expenditure of power and could leadto increased packet loss due to congestion. Another issue withthenaivefloodingmethodisthatitprovidesnoreliability.Itisimportanttoensurethatallnodesinthenetworkreceivethe entire file in a timely manner, so reliability mechanismswillbeneeded.

Wan, et al. proposed Pump Slowly, Fetch Quickly (PSFQ)[1], a reliable transport protocol which reduces transmissions,making it energy efficient and scalable. The basic premisebehindtheprotocolistopropagatethesegmentsatarelativelyslow pace ("pump slowly"), and use an aggressive NACKmechanism to fetch missed segments ("fetch quickly"). Thisprotocolattemptstominimizethecostoflostpacketsbyreducing NACKsandrebroadcastsinresponsetoNACKstoa single hop. They also reduce the amount of transmissionsthrough forwarding by using a counter mechanism to limittransmissions. These methods are effective in reducing thetotal number of transmissions necessary to flood all segmentsof a file throughout a sensor network, however, the

aggressivetiming of the fetch operation can lead to congestion in a densenetworkwhennodesattempttorespondtoNACKs.Also,the retransmission delay that is added to allow for handlingNACKs places a lower bound on the latency. We proposePushAggressivelywithLazyErrorRecovery(PALER),whichis based on PSFQ with the enhancement that NACKs arereserveduntiltheendofthetransmissionofallsegments.ThisreducestheamountofNACKpackets,particularlyinhigh-lossenvironments. A slower response period reduces collisions inNACK responses, and further reduces the number of totaltransmissions for complete dissemination. This modificationalsoeliminatestheneedforaretransmissiondelay,allowingamoreaggressiveforwardingmethod,leadingtoreducedlatency.
PSFQandPALERhavebeenimplementedandevaluated through simulation to exhibit the effects of node density andhop count on total transmissions and latency. These resultsshowthatamorerelaxedrecoverymechanismavoidsmuchof the contention and collision found in a more aggressivemechanism.Italsoprovidesanopportunityforimprovedcollaboration among nodes to avoid redundant responses. Inaddition,withholdingnegativeacknowledgementsuntilallsegments of a dataset have been broadcast allows a moreaggressive propagation of segments. This allows PALER toimprove energy efficiency by reducing transmissions whilereducing latency. The results show that PALER is capable offunctioning efficiently in very dense networks, and latencyscales very well across an increasing field size. We comparetwo variations of PALER to evaluate the advantages of differ-ingpruningtechniquesinthefloodingphase.VersionIutilizesa counter method to reduce redundant transmissions, whileversion II utilizes local neighbor data to prune unnecessarytransmissions.

II. RELATED WORK

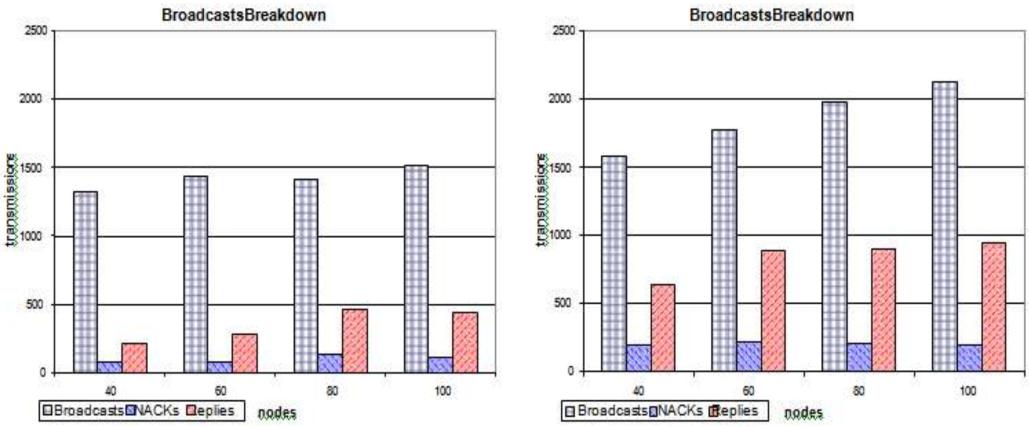
Code distribution has many similarities with reliable mul-ticast, reliable broadcasting and energy-efficent broadcastingresearch. Many traditional techniques for making multicastreliable in wired networks, such as [2], are too computation-ally expensive for the limited resources of a wireless node.Multicast protocols which have been developed for wirelessnetworks, such as [3] and [4], tend to favor robustness toprovidereliability.Thisbuilt-inredundancycomesattheexpense of energy efficiency. Wireless broadcasting protocolsfocused on reliability typically require significant overhead incontrolpackets[5,6].

Energy-efficient wireless broadcast protocols typically fo-cusintwoareas:minimizingforwardingnodes[7,8,9,10], or minimizing transmission power [11, 12, 13]. [7] offersseveral methods of reducing forwarding nodes, such as prob-abilistic, counter, and cluster methods, each with differinglevelsofreliability.[10]developedanalgorithmtoidentifya dominating set in a network, which would make up theintermediatenodesin allbroadcasts.[9] and[8] builtuponthismethodbyreducingthesizeofthedominatingset.Selection of a dominating set may reduce total transmissionsinabroadcast,however,itdoesnotbalancetheload,asthenodesinthedominatingsetwillincurallofthecost.Aninterestingextensionof[9]isaneighboreliminationmethodwhichisverysimilartothemethodusedinPALERto reduce forwarding nodes. [14] also used an approximationof minimum dominating set to achieve reliability and reducethe cost of recovery. Their scheme is capable of providingreliable broadcast of a single packet, and limits the cost ofdropped packets with local recovery, but the focus of thisworkis onreliability ratherthantheenergyefficiencyofthe downstream propagation. Broadcast protocols aimed atminimizing transmission power are typically based upon a setcover[12]orminimumspanning[11,13]problem.Thesetendtoprovideveryefficientdistributiontreesand balanceloads evenly; however, they require a knowledge of the completenetwork topology, and are best used in a static environmentwhereoptimalroutescanbepredetermined.

Others methods of achieving reliability or efficiency in-cludeFECandnetworkcodingtechniques.FECisusedin conjunction with probabilistic forwarding by [15] to addreliability to an efficiency scheme. The FEC technique offersflexibilityinthepropagationandrecoverymethods.Withalargedatadistribution,thenumberofoverheadpacketsgeneratedbytheFECisexpectedtobeminimal relativetothe total packets sent, but this assumes a lossless environmentand ideal MAC layer. A realistic scenario could require anundesirablenumberofoverheadpacketsproducedbytheencoding method to provide reliability. The forwarding prob-ability used for their simulations is optimized for the networktopology, which provided efficient propagation results. Thiswould make implementations sensitive to alterations in thenetwork size and density, however, which could lead to a lossof efficiency or reliability. Network coding is used in [16] toimprove efficiency by reducing the number of transmissions.They provide an alternative algorithm in their approach formulticastinwirelessnetworkstoexploitthewireless multicastadvantage.Whilethenumberoftransmission smaybereducedin this method, the potential overhead cost of the control datawhich must be appended to each packet is not evaluated. Forlarge data transmissions, as may be used for re-tasking, thiscost could be significant. Extensions would also need to bemadetothisschemetoprovidereliabilityandlossrec overy.

Trickle [17] is a popular recent method of code propagationand maintenance. It uses a gossiping protocol with periodicmetadata broadcasts to identify nodes which require an updatetoanewversionofcode.Itusesacountermethodt olimitthe number of gossip messages broadcast during an interval,which makes Trickle a very energy efficient method of main-taininga sensornetwork.Trickleisnotgreatlyconcerne dwithlatency,andisbaseduponanexpectationofavery smallcodesegmentorbinaryfile,onewhichwillfitinas mall number of packets. Our protocol will be more focusedonanefficientmethodforasinglereprogrammi ngeventof a relatively large binary file. Another gossip based codepropagation protocol is GCP [18]. It uses periodic beacons todetect outdated code versions, similar to Trickle. However, italso includes a forwarding control mechanism to balance theload of distribution. Each node has a limited number of tokensthatitmayusefordistributingeachnewversiono fcode.

Melete [19] builds upon Trickle to support dynamic group-ing and concurrent applications in sensor networks. It uses aperiodic metadata broadcast to maintain the network. It alsosupports group-based code propagation. Nodes may dynami-cally enter and exit a group, and must broadcast a request forthe new group code. Melete avoids broadcast implosion bypacing requests through a probabilistic and progressive flood-ing mechanism. Because of the dynamic grouping nature oftheirsystem,codepropagationwillbeaccomplishedt hrough



(a) Loss:10% (b)Loss:50%

Fig. 1: Breakdown of the types of transmissions during a simulation of PSFQ. Simulation had a fixed field size with increasingnodecount.(a)showstheresultsofalow-lossenvironment,10%.(b)showstheresultsofahigh-lossenvironment,50%.multi-hopunicastinmanysituations.Inourprotocol,wewillinsteadfocusonaone-to-alldistributionofalargedatafile.

III. PROTOCOL DESIGN

A. PSFQDesign

PSFQ [1] distributes data from a single source to a networkof sinks by slowly pacing the propagation of packets. Nodesthat detect a lost packet due to out-of-order packet receptionwillaggressivelyfetchthemissingpacketbys endingaNACKto its immediate neighbors. Nodes will refrain from forward-ing packets received out-of-order until all packets leading upthat one have been recovered.In other words, regardless ofthe order of reception, nodes will only rebroadcast packetsin-order.Thispreventsthepropagationofalosseventtothe key to PSFQ is the relation between $T_r$and T_{max}. The ratiobetween $T_r$and T_{max}determines how many opportunities anodehastofetchamissingsegmentbeforethenextse gmentisexpectedtoarrive.ThehighertheratioT_{max}/T_rthegreaterthe probability that the segment will be successfully receivedover multiple hops. In their implementation, they selected$T_{max} = 5 \ast T_r$, and $T_{min} = \frac{1}{2} T_{max}$. $T_r$must also be chosento provide an adequate window for recovery. To provide aminimumnecessarywindow,$T_r$shouldbeatleastfo urtimesthe latency for a single packet, but they proposed that areasonablevaluewouldbe$T_r = 6 \ast T_p$. WhenanodereceivesaNACK,itcheckstherequestagai nstits own cache to determine if it has any of the missingsegments.Ifitdoes,itwillschedulea replywiththemissingsegmentatarandomtimewithintheinterval$^1T_r$to^1T_r.To downstreamnodes.Requiringin-orderbroadcastsalsoensures 4 2

that lost packets can be retrieved from at least one immediateneighbor, since the neighbor that broadcast the higher thanexpected sequence number must also contain the expectedsequence number. By localizing recovery, it reduces lost re-covery cost by suppressing the propagation of loss events andnegativeacknowledgements,andreducingrecoverytoasinglehop transmission. The PSFQ protocol is built upon a tightlycoupled timing between the pushing mechanisms and fetchingmechanisms. The pumping mechanism relies on two timers,$T_{min}$and T_{max}. After reception of each in-order packet, thepacket will be scheduled for re-broadcast following a randomdelaybetween$T_{min}$and T_{max}. A counteris maintainedforthenumberoftimeseachpacketisheard. Ifapacketisreceived four times prior to rebroadcasting, the rebroadcasteventiscanceledtosuppressredundanttransmissions. Ifapacketisreceivedout-of-order,thenodewillschedulea NACK with a request for the missing segments after a shortrandom delay. The node will continue to send a NACK every$T_r$untilallofthemissingsegmentshavebeenreceived.The reduce contention and redundant transmissions, it will cancelthe reply event if it overhears any neighbors responding to thesame missing segment prior to its own reply. To improve thelikelihood that only one neighbor will respond to a NACKrequest, each node maintains a table with the average signalstrengthsofitsparentnodes.WhenitsendsaNACK,itincludes its preferred neighbor (node with the highest averagesignal strength) in the header. The neighboring nodes willchecktodetermineifitisthepreferredneighbor;ifnot,itwill double it's delay interval before sending the reply, givingitmoretimetooverheararepl yfromthepreferred neighbor.

The slow pumping mechanism of PSFQ along with thecountermethod for pruningof forwarding nodesis effectiveinreducingcontentionandcollision,andefficientlyprop-agatingafilewithminimalredundanttransmissions.Thefetching mechanism avoids the NACK implosion problem bylocalizing recovery. However, the aggressive nature of thefetch mechanism and the relatively short time frame of therecoveryintervalsleadstoagreatdealofcontention. While the neighboring nodes listen for duplicate replies from otherneighbors,theresponsetimeframedoesnotallow alargewindow to randomly disperse the responses and to overhearresponses. Each node is generating a random delay periodbetween$^1T_r$to^1T_r.Thisintervalequatestojust^3T_p,which iscanceled.BecauseNACKsarewithhelduntilthefinal segment, intermediate nodes do not have to have a minimumdelayperiodpriortorebroadcasting,sotheintervalcanbeanyrandomperiodwithin100ms.Sincethecounterlimitstherebroadcastwithinaregiontothefirstthreeattempts,the provideslittleopportunityforoverhearing.Theinclusionofa preferred neighbor based on received signal strength helpsincreasethewindowofopportunityforoverhearingotherresponses, but nodes that computed a small random delay willstill be transmitting close to the same time as the preferrednode.Inadensenetwork,thismayresultinahighnumberof collisions, possibly resulting in more redundant responses.Figure 1 shows a breakdown of the types of transmissionsobservedinasimulationofPSFQ.Inthissimulation,thefieldissetto1km 1km,andthedensityrangesfrom40to100nodesperkm^2 .Thefirstgraphshowstheresultsforanaverage10%packetloss, thesecondgraphdepictsthe same results for a 50% loss environment. These figuresshow that the number of response messages are a significantpercentage of the total transmissions. Many of these replytransmissions are redundant, and could be avoided with a lessaggressive method. Additionally, in a high loss environment,thein-orderrequirementcanresultinahighnumberofNACKtransmissions. Another impact of the timing constraints is thatitplacesalowerboundonthelatencythatisequivalentttoT_{min}hops, where hopsis the maximum number of hopsneededtoreachallnodesfromthesource.

B. PALERDesign

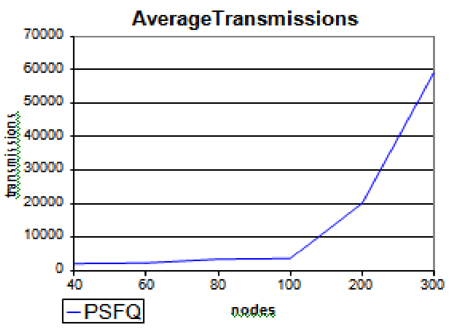
To avoid the contention and collisions resulting from theaggressive recovery mechanism of PSFQ, PALER eliminatesthein-orderreceptionrequirementandmaintainsalistofmissing segments. After all packets have been broadcast, eachnode will broadcast a NACK to its neighbors. An importantaspect of PALER is that, even though it does not require in-order reception, it still maintains a local recovery mechanism.Thereasonthisispossibleisbecauseifaneighborreceivesa NACK, it can first check its own cache for the missingsegments. Any segments that are present in the local cachemaybesenttotherequesterinasinglehoptransmission.As for segments that are not present in the local cache, thesesegments must also be among the list of missing segments onthe local node, which means that they will be included in thelocal nodes NACK. Therefore, it does not need to propagatethe NACK from its neighbor, because the missing segmentswill be redundant. This maintains local recovery with singlehopNACKtransmissions,whilestillensuringth atrequestfor missing segments will be propagated until the missingsegmentsarefound.

Whenanodereceivesasegment,itchecksifth esegmentisalreadycontainedinitscache;ifitis,itincre mentsacounter for that segment. If it is the first reception of thissegment, it schedules a forwarding event at a random delaybetween 0 and 100 ms.If the counter for the segment reachesthreebeforethesegmentisrebroadcast,theforw ardingeventaverageintervalpriortoarebroadcasttends totrendcloserto0 than 100 ms, particularly in dense networks. This eliminatesthelowerboundonlatency,andprovidesam uchimprovedlatencythatscaleswellacrosslargemulti-hopenvironments.Whenanodereceivesthelastsegme ntofadistribution,itschedulesabroadcastofaNACKto itsone-hopneighborswhichincludesalistofsegmentsitismiss ing.TheNACKisscheduledfollowingarandomdelayp eriod,whichisusedtoreducecollisions,andtoallowthe nodeanopportunitytooverhearrebroadcastedsegment sinresponsetootherNACKs.Ifitoverhearsasegmentth atitwasmissingpriortobroadcastingitsNACK,itwillre movethatsegmentnumberfromitslistofmissingsegme nts.WhenaneighboringnodereceivesaNACK,itcheck sthelistofmissingsegmentsagainstitsowncache.Ifito wnsanyofthemissingsegments,itwillschedulea reply withthemissingsegmentfollowingarandomdelaybet ween0to50ms.Ifitownsmorethanonesegment,itwills cheduleeachadditionalsegmentforareplyat10msinter vals.Thedelaypriortorepliesreducescontentionandall owsthenodestooverhearotherreplies.Ifanodeoverhea rsareplyforasegmentthatithasscheduled areplyfor,itwillcancelitsreplyevent.

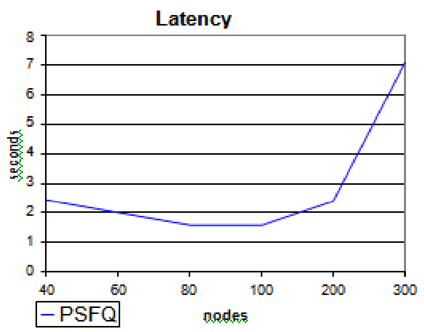
Since this recovery mechanism is dependent upon receptionofthelastsegment,atimeoutperiodisusedin caseoflossofthe last segment. The timeout period is continuously updatedto be $T_{begin}+ \alpha (T_{avg}[Seg_{tot}Seg_{rec}])$. Here, T_{begin}isthe time the first segment was received, $T_{avg}$is the averageinterval between reception of each segment, $Seg_{tot}$is thetotal number of segments, $Seg_{rec}$is the number of segmentsreceived, and α1 is a small multiplier that determines howaggressive the timeout value should be, typically this wouldbe less than 1.5. Following the transmission of each NACK, anew timeout value is set to specify a maximum time expectedbeforeallmissingpacketsarerecovered.Ifan ysegmentsarestillmissingattheendofthistimeoutper iod,anotherNACKwillbetransmitted,andanewtime outvaluewillbeset.Thefirst timeout following a NACK will be calculated using thesame formula as above. If additional NACK's are necessary,each additional timeout period will be doubled to avoid aNACK implosion resulting from downstream nodes waitingforamulti-hoprecoverytopropagate.

C. PALERFloodingMechanism

With the lazy error recovery of PALER, NACK and re-covery transmissions are greatly reduced, leading to a veryefficient broadcast with minimal wasted energy. However, thepushing operation still requires a relatively large number oftransmissions, even in fairly dense networks, and many ofthese transmissions are redundant. PSFQ increased efficiencyoverstandardfloodingbyusingacounterwit hacutoffof4to



(a) Transmissions



(b) Latency

Fig.2:ResultsofPSFQsimulationwithT_r=6ms.Simulationhadafixedfieldsizewithincreasingnodecount.(a)showsthet otalnumberoftransmissionsduringthesimulation.(b)showstheaveragelatency.

locally limit the number of rebroadcasts within a region.

[7]showedthatifanodeoverhearsamessage4times,the averageadditional coverage that can be achieved by performing itsown rebroadcast of the message is just 5%. The additionalcoveragethatcanbeexpectedafteroverhearingamessage3 times was shownto beabout9%. In PALER,a counterwithacutoffof3isusedtolimitredundanttransmissions.Theselectionof3asthecutoffprovedtoreduceforward-ing transmissions without significantly affecting NACKs orrecoverytransmissions.

Tofurtherimprovetheefficiencyofthepropagationmethod, a new mechanism was devised that used two-hopneighbor information that is dynamically acquired. To gener-ate a representation of each node's two-hop neighborhood,eachnodewillincludealistofitsimmediateneighborswith the first n segments of a broadcast. The value n is animplementation set value that will determine the strength oftheneighborhoodrepresentation;inourimplementationnis 10. When a node receives a segment with this list in theheader,itwilladdpairingsofthesendingnodewitheachofitsneighbors to a table that list all of the immediate neighbors ofnodes within its two-hop neighborhood. To reduce the amountof contention and collision during the propagation of the firstnsegments, particularly in dense networks, a probabilisticmethod can be used to limit the number of rebroadcasts. Aprobabilistic method is used instead of a counter, because acounter could result in the same nodes rebroadcasting eachtime, resulting in an incomplete neighborhood representation.Theprobabilityselectedshouldbehighenoughtoensurethateachnodebroadcastitsneighborhoodinformationatleast once,which is implementationdependentonn. It canbe determined by using a cumulative probability distributionfunctiontodeterminewhatvalueofpisneededforthedesiredconfidencelevel.

Following the initial n segments, when a node receives asegment for the first time, it generates a list of its neighborsandschedulesarebroadcasteventatarandomtimeasbefore

Thelistofneighborsrepresentsalloftheimmediateneighborsthat may need this current segment. For each transmission ofthatsegmentitreceivespriortorebroadcasting,itremovesthe sending node from the list of neighbors associated withthat packet. It then acquires the list of the sending node'simmediate neighbors from its neighbor table, and removeseachofthesenodes(ifpresent)fromitslistofneighborsfor this segment. These are removed because it is able toassumethateachofthesenodesreceivedthesegmentfromthesendingnode,andthereforedonotneedthissegment.Ifthelistofneighborsforasegmentbecomesemptypriorto forwarding the segment, the rebroadcast event is canceled,sinceitisassumedthatallneighborsreceivedthesegment.

IV. SIMULATION METHOD

To evaluate PALER, it was implemented in the Jist/Swanssimulation environment [20, 21], a scalable wireless ad hocnetwork simulator based in Java. PSFQ was also implementedfor comparison. Swans provides a full representation of thecomplete network layer model, with accurate representationsof a wireless environment, including path loss, environmentalnoise and collision interference. Each node in the simulationwasimplementedwithan802.11radio,with arangeofapproximately250meters.Twosetsofsimulationswereperformed,onewithafixedfieldsizeof1km×1km,withan increasing density, and another with a fixed density andincreasingfieldsize.Foreachsimulation,afileofsize50KB was distributed. The file was segmented into 50 chunks,each1KB.Theprimarymetricsusedtoevaluate theprotocolsare total transmissions and latency, where latency is measuredasthetimeneededforallnodestoreceiveeverysegmentof the data file. The goal of PALER is to improve on theefficiencyofPSFQbyreducingtransmissions,whilestillmeetingorexceedingthelatencyperformanceofPSFQ.Inou simulation results, we show that this was accomplished.Our evaluation of PALER will show that energy efficiencycanbefurtherimprovedusingdynamic,local neighborhood

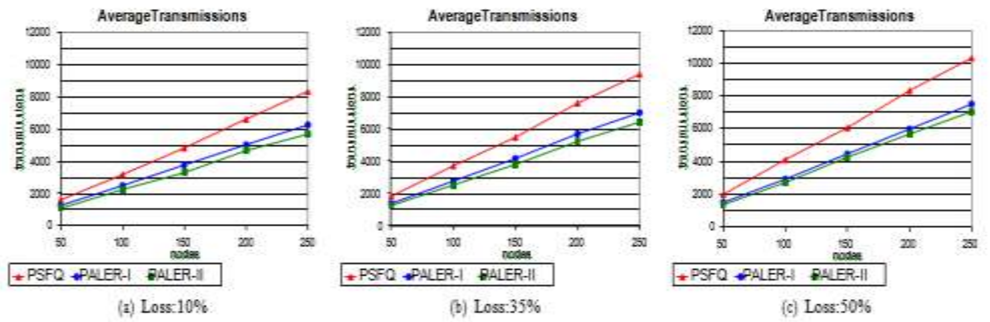


Fig. 3: Comparison of total transmissions of PSFQ, PALER-I and PALER-II, measured across increasing field size. Density isfixedat50nodesperkm^2.A50KBfileistransmittedin1KBsegments.(a)showstheresultsforalow-lossenvironment,10%.(b)showstheresultsfor35%loss,and(c)fora50%loss.PALER-IandPALER-IIexhibitasignificantdecreaseinthe number of total transmissions required to complete a code distribution, with PALER-II demonstrating the best energyefficiency.

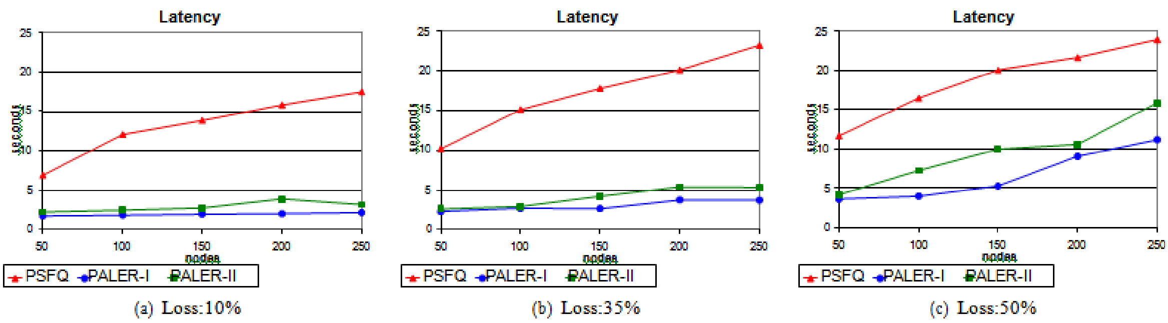


Fig. 4: Comparison of average latency of PSFQ, PALER-I and PALER-II with a fixed density and increasing field size.

Theselatenciescorrespondtothesimulationresultsin Figure3.PALERshowsanabilitytoscaleextremelywell acrossmultiplehops,scalingbelowalinearrateofincrease.

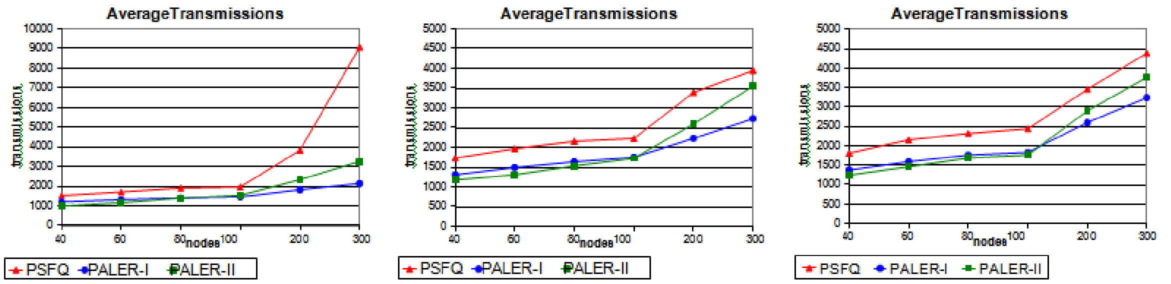
information,thoughthismayslightlyimpactlatency.F orcomparison, we have implemented both flooding mechanismsin PALER. The results for the PALER implementation usingthe counter method are presented as PALER-I, and the resultsfor the implementation utilizing the neighbor pruning methodarepresentedasPALER-II.

To determine the timing parameters used for the PSFQ im-plementation,atestscenariowasimplementedinJist/S wansto determine the average time for a packet transmission withthe range of densities used in our simulations. The≈calibrationresults showed that T_p1ms. Based on the recommendationin [1], $T_r$was set to 6T_p. Therefore, in our initial imple-mentation of PSFQ, the timing parameters were T_r= 6ms,T_{min}= 15ms, and T_{max}= 30ms. The simulation results forthis implementation exhibited a very poor energy efficiency,anddidnotscalewellwithhighnodedensities.Theresultsof these simulations for a low loss environment are shown inFigure 2. The simulations were implemented with a field sizeof 1km1km, with an average packet loss of 10% and 40 to300randomlyplacednodes,eachwitharangeof250m .Even at40nodes,thenumberoftransmissionsisabove2000, whichis what would result from a basic flooding mechanism. It canbeseenthatasthedensityincreasesabove100perkm^2,PSFQdoes not scale well, and the number of transmissions quicklyescalatestomultipletimesthevalueofflooding. Therefore,to provide a stronger basis for comparison, the timing metricswerechangedtothoseusedinWanetal.'simple mentation[1].For their implementation, the timing metrics were T_r= 20ms,T_{min}=50ms,T_{max}=100ms.Thisprovidedagreatlyimproved energy efficiency with transmissions an order ofmagnitude smaller, though the latency did increase as a resultofthelowerboundimposedbyT_{min}.

V. RESULTS

The first simulation performed a

comparison for varyingfieldsizeofanetwork.Thedensityforthesesimulationsis fixed at 50 nodes per km^2. The number of nodes in thesimulationrangesfrom50to250,whichcorrespond stoafield size ranging from 1km 1kmto 2.236km 2.236km(orawidthof4hopstoawidthof9hops).Theresultsin



(a) Loss:10% (b) Loss:35% (c) Loss:50%

Fig.5:ComparisonoftotaltransmissionsofPSFQ,PALER-IandPALER-IImeasuredagainstincreasingdensity.Thefieldisfixedat1km1km,eachnodehasarangeof250m.A50KBfileistransmittedin1KBsegments.(a)showstheresultsfor a low-loss environment, 10%. (b) shows the results for 35% loss, and (c) for a 50% loss. The benefits of PALER-II aregreatestinnetworkswithanaveragenodeconnectivitylessthan20.

Figure 3 show the average number of transmissions, measuredby the network size, for an average loss environment of 10%,35% and 50%. The graphs show that PALER-I exhibits a 20-30%reductionintransmissionsfromPSFQ,whilePALER-IIexhibitsa30-35%reduction.Theincreaseinefficiencyisfairlyconsistentacrossdifferinglossenvironmentsanddifferingfieldsizes.

The second set of graphs in Figure 4 show the correspond-ing latencies for the same set of simulations. Due to its lowerbound on latency, PSFQ scales fairly linearly with field size(number of hops). However, it does perform rather consis-tently among increasing loss environments. As seen, PALERscales very well across increasing field sizes, particularly inmoderate loss environments. In high loss environments, suchasthatofFigure4c,latencyincreasesinstrongerrelationtofieldsize,butstillperformsstronglyincomparisontoPSFQ.PALER-Iexhibitedthebestlatency,revealingthetradeoffofenergyefficiencyandlatencybetweenthefloodingmechanisms. This slight increase in latency in PALER-II isdue in part to the random timing mechanism of PALER. InPALER-I, the nodes which do not choose the lowest randomdelay periods are the ones that suppress forwarding, whereasin PALER-II, the pruning selection is partially independent ofthedelayperiodchosen.

To examine how PALER performs in differing densities,anothersetofsimulationswasperformedwithdensityvaryingfrom 40 nodes per $km^2$to 300 per km^2. The results are shownin Figure 5, for loss environments of 10%, 35% and 50%. Theefficiency improvement of PALER-II is approximately 30%,whichisconsistentacrossalldensitieswhenaveragedoverthedifferent loss environments. PALER-I increases in efficiencyimprovement as density increases. It results in a reduction oftransmissions of 23% for a density of 40 nodes per km^2, butnearly 45% for densities of 300 nodes per km^2. In extremelydenseenvironments,PSFQactuallyperformedworseinalow-lossenvironment,duetohighcontentionduringrecovery.Asaresult,Figure5aisshownonadifferentscaletoaccommodatetheplotofPSFQ.TheseresultsshowthatPALERscaleswell to extremely dense networks. The overhead of generatingneighborhood information in PALER-II can be seen in verydense networks. Since PALER-II requires nodes to broadcastat a certain probability for an initial period, collisions canresult in very dense networks. For densities below 100 nodesperkm^2,PALER-IIperformsmoreefficiently,butatdensitiesabove100nodesperkm^2PALER-Iperformsmoreefficiently.This point of intersection corresponds to an average nodeconnectivity of approximately 20. In many cases, this may bean acceptable upper bound, but if an extremely dense networkisexpected,PALER-Iwillperformefficiently.

The greatest factor in the reduction of transmissions byPALER is the reduction of contention and collisions duringrecovery. The lazy recovery method allows nodes to carefullypace recovery operations and avoid redundant transmissions.Figure6showsthebreakdownofPALERtransmissionsbetween broadcasts, NACKs, and reply messages. The graphsfrom Figure 1 are included for comparison, it can be seen thatthe number of broadcasts is fairly similar, but the number ofNACKsandrepliesmakeupasignificantlysmallerpercentageoftotaltransmissionsinPALER.

VI. CONCLUSIONS AND FUTURE WORK

The results from the previous section show that PALER isable to consistently improve energy efficiency above PSFQacrossabroadrangeofnetworksizesanddensities.Inaddition, latency proved to scale very well in PALER, par-ticularly in moderate loss environments. The main reason fortheimprovedefficiencyinthenumberoftotalbroadcastisdue to the more relaxed nature of the recovery mechanism,which avoided much of the contention that affected PSFQ.ThecomparisoninFigure6showsthatPALERperformsfairlyconsistentlywithPSFQinthetotalnumberofforwardedbroadcasts.However,thepercentageofNACKandreplymessages is greatly reduced in PALER. PALER implementedwithaneighborpruningmechanismshowedanadditional

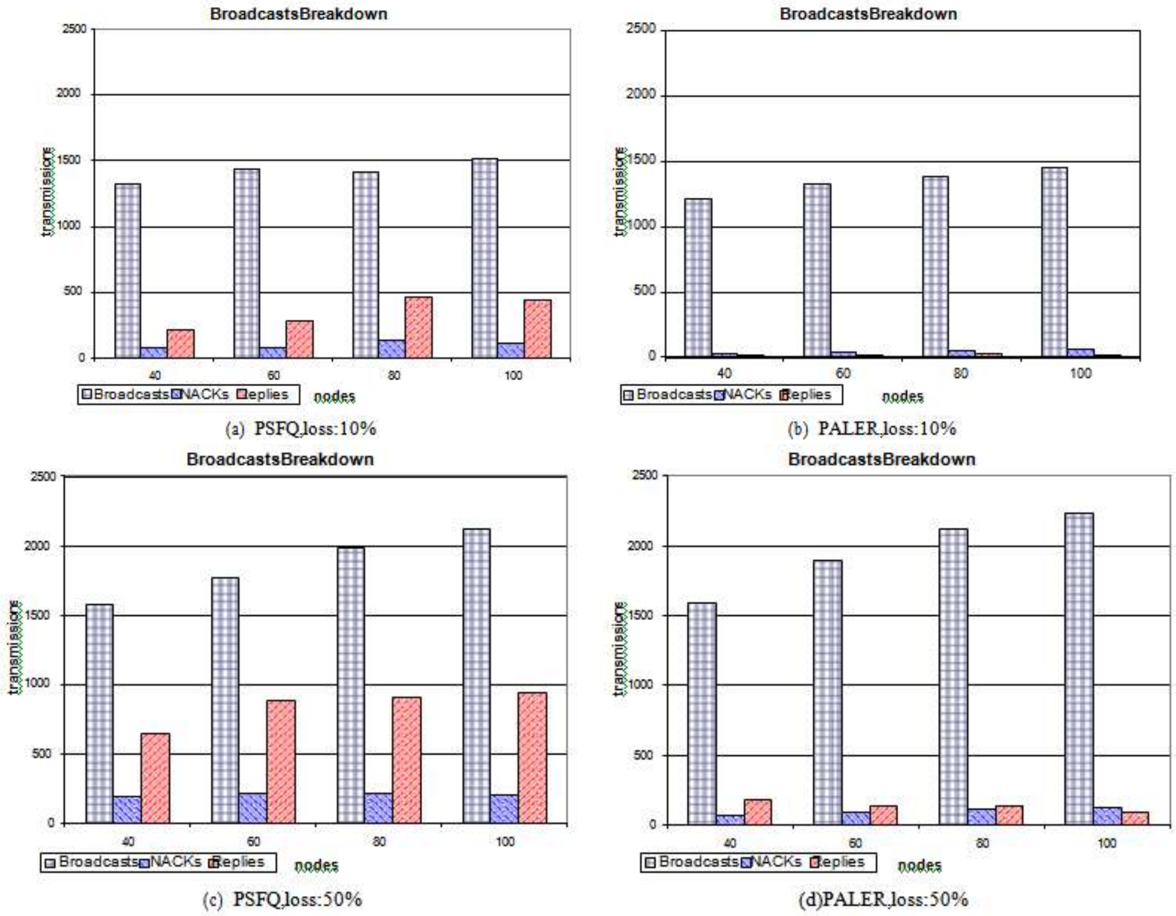


Fig. 6: Breakdown of the types of transmissions during a simulation of PALER. Simulation had a fixed field size withincreasing node count. PSFQ graphs as shown in Figure 1 are included for comparison (a)&(b) shows a comparison for alow-loss environment, 10%. (c)&(d) shows a comparison for a high-loss environment 50%.

PALER shows a much smallerpercentageofNACKandreplymessages,relativetobroadcasts,incomparisontoPSFQ.

improvement in energy efficiency in moderate densities. Thisincreaseinenergyefficiencycameatthecostofslightlyhigherlatenciesthanthecountermethodimplementation.However,theaveragelatenciesofbothimplementationsscaledconsiderably better than PSFQ, and in many sensor networkenvironmentsthesavingsinpowerconsumptionwilloutweighthesmallincreaseinlatency. Weplantofurtherdevelopacomprehensivesensornetworkreprogramming framework. Additional methods to improveenergyefficiencywillbeexplored,suchasdynamicallyadjust-ing radio transmission power. Neighbor locality informationmay be estimated from received signal strength. This infor-mation may be used to determine the minimum transmissionpower needed to reach all neighbors. The neighbor pruningmechanism leads nicely in this direction, since a list of neigh-borsrequiringthesegmentismaintained.Iftheestimated transmission power to reach a neighbor is added to each nodein the list, nodes may dynamically adjust their transmissionpowertothemaximumofthesevaluesintheremaininglistof neighbors. This has the potential to reduce the total powerexpendedfor distributing newcode,andreducecontentionand collisions since the transmission range will be limited tothedesiredcoveragearea. Otherareasforfutureworkincludeaddingversionmetadatatofacilitateautomaticupdatesandmaintenance.Afullyimplemented system should be capable of ensuring that allnodes maintain the most current version of code. This willrequire the ability to detect new or outdated versions andupdateanynecessarynodesinamannerthatallowsthenetworktocontinuetoperformreliablyduringtheupdate.Wewouldalsoliketosupportgroup-centricre-tasking.Sensornetworknodesmaybeorganizedintogroupswhichperform

specialized functions. These group assignments may adjustdynamically to respond to the environment. We will exploremethodstoefficientlysupportsuchfunctionality.

REFERENCES

[1] A. K. L. Wan, C.-Y.; Campbell, “Pump-slowly, fetch-quickly (psfq): A reliable transport protocol for sensornetworks,”SelectedAreasinCommunications,IEEEJournalon,vol.23,no.4,pp.862–872,April2005.

[2] L.RizzoandL.Vicisano, “A reliable multicastdatadistributionprotocolbasedonsoftwareFECtechniques,”inThe Fourth IEEE Workshop on theArchitecture and Implementation of High PerformanceCommunicationSystems(HPCS’97),SaniBeach,Chalkidiki,Greece,Jun.1997.[Online].Available:citeseer.ist.psu.edu/rizzo97reliable.html

[3] Y.YiandS.Lee,“Demandmulticastroutingprotocol,”2002.[Online].Available:citeseer.ist.psu.edu/yi02demand.html

[4] C.Ho,K.Obraczka,G.Tsudik,andK.Viswanath,“Flooding for reliable multicast in multi-hop ad hoc networks,” in Proceedingsofthe3rd International Workshop on Discrete Algorithms andMethods for Mobile Computing and Communications,Seattle,WA,1999,pp.64–71.[Online].Available:citeseer.ist.psu.edu/ho99flooding.html

[5] C.-S. Hsu, Y.-C. Tseng, and J.-P. Sheu, “An efficientreliable broadcasting protocol for wireless mobile ad hocnetworks,” Ad Hoc Netw., vol. 5, no. 3, pp. 299–312,2007.

[6] E.PaganiandG.P.Rossi,“Reliablebroadcastin mobile multihop packet networks,” in MobiCom ’97:Proceedings of the 3rd annual ACM/IEEE internationalconference on Mobile computing and networking.NewYork,NY,USA:ACM,1997, pp.34–42.

[7] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, “Thebroadcast storm problem in a mobile ad hoc network,”Wirel.Netw.,vol.8,no.2/3,pp.153–167,2002.

[8] J. Wu and F. Dai, “Broadcasting in ad hoc networksbased on self-pruning.” [Online]. Available: citeseer.ist.psu.edu/wu03broadcasting.html

[9] I.Stojmenovic,M.Seddigh,andJ.Zunic,“Dominatingsetsandneighborelimination-basedbroadcasting algorithms in wireless networks,”IEEE Transactions on Parallel and Distributed Systems,vol. 13, no. 1, pp. 14–25, 2002. [Online]. Available:citeseer.ist.psu.edu/stojmenovic01dominating.html