An Experimental Evaluation of Peer-To-Peer Reliable Multicast Protocols

Giacomo Benincasa, Andrea Rossi, Niranjan Suri
Florida Institute for Human and Machine Cognition
Pensacola, FL USA
{gbenincasa, arossi, nsuri}@ihmc.us

Mauro Tortonesi, Cesare Stefanelli
University of Ferrara
Ferrara, Italy
{mauro.tortonesi, cesare.stefanelli}@unife.it

***Abstract*—Tactical operations often involve the cooperation of multiple actors that need to communicate in a reliable and timely fashion. Numerous critical activities that are performed in this context, such as the dissemination of situational awareness data, or the dissemination of command and control information, present a point-to-multipoint pattern. Therefore, multicast protocols are a suitable approach to perform efficient data dissemination in this context. More specifically, some tactical information requires reliable point-to-multipoint delivery of information. In this paper, we experimentally evaluate four protocols (and corresponding implementations) that have been developed to support reliable multicast communications: NORM, JGroups, OpenPGM, and DisService. We report on two sets of experiments. The first set of experiments measure bandwidth utilization and average delivery time under different emulated network conditions. The second set of experiments performs a more in-depth comparison of the forward error correction approach implemented in NORM with DisService, which adopts an opportunistic approach for information dissemination.**

Keywords: reliable multicast; information dissemination; publish-subscribe middleware; tactical networks;

I. INTRODUCTION

Tactical edge networks operate in dynamic and heterogeneous environments. Different patterns of node mobility, constrained and potentially asymmetric links, frequent disconnections due to occlusion and high bit error rate, frequent congestions, and possible network partitioning are common problems that have to be taken into account when developing communications protocols and implementations. Performing reliable group communication is of significant importance since it would decrease the number of copies of the same message that need to be sent over an unreliable network and therefore greatly reduce the amount of traffic on the network; on the other hand activities such as group membership management, transmission of repair traffic, and acknowledgment handling in group communication pose problems that lead to poor scalability if not properly handled.

These challenging environments fostered research and development of peer-to-peer (P2P) information dissemination systems that continue to function even in presence of network partitioning, and that are able to synchronize when connectivity between partitioned areas is re-established. The robustness of P2P systems is the result of their independence from designated server nodes. In P2P systems, any node can act as a service provider and therefore there is no need to route communication to a centralized location. An analysis of the suitability of P2P systems for tactical network environments can be found in [18].

Tactical applications often have a one-to-many or many-to-many dissemination pattern. Some applications, such as Blue Force Tracking (BFT), do not require reliability as multiple updates are generated over time. Other applications that support team coordination (such as instant messaging, tasking, sharing of intelligence information or sensor data, and collaborative map editing) would benefit from the use of a group communication paradigm. In particular, these applications call for specific solutions supporting reliable multicast communications that are capable of disseminating the information in a timely and efficient manner.

Researchers have devised several techniques to improve the reliability and efficiency of multicast communications in MANETs and tactical edge networks. Cooperative networking lets the peers of the network cooperate in order to achieve greater overall network performance. Techniques such as cooperative caching and cooperative recovery allow for greater availability and survivability of the information over the network, and can balance the load on the nodes of the network. The broadcast nature of the wireless medium can be exploited to communicate efficiently with groups of nodes at the same cost of a unicast message. Packet erasure coding is an effective technique that can potentially minimize transmission errors, at the cost of transmitting redundant data.

Many exhaustive surveys, such as [1] and [2], that qualitatively evaluate the advantages and disadvantages of the mentioned approaches exist. This paper, however, aims at quantifying those advantages and measuring the possible overhead in terms of bandwidth and latency that these approaches may introduce. To this end, we analyzed two reliable multicast communications middlewares that were specifically designed for mobile networks, and that implement the above mentioned approaches: NORM [12] and DisService [6]. NORM is a multicast communication protocol that exploits an efficient reliability mechanism based on a hybrid approach that combines ARQ and packet erasure coding. DisService is an information dissemination middleware that we developed, that opportunistically exploits broadcast communications to perform cooperative networking.

We present two series of tests. In the first one, the performance of NORM and DisServiceis compared with OpenPGM [16] and JGroups [17]. OpenPGM is an

implementation of the Pragmatic General Multicast (PGM) protocol [14], and JGroups is a communications middleware included in the JBoss application server as part of a clustering framework to perform reliable multicast. We considered OpenPGM and JGroups in our tests because even though they were designed for wired networks, they represent state-of-the-art COTS applications. They are therefore interesting reference points for the comparison. The second series of tests introduces constraints such as asymmetric links, with the purpose of comparing NORM and DisService in a more realistic and challenging environment. These tests do not consider OpenPGM and JGroups because they were not designed for such network conditions and hence do not behave acceptably.

While we acknowledge the importance that the routing and membership management protocols have on the performance of the system, and while we take it into account in the following sections when qualitatively describing the different approaches, the performance evaluation has the sole purpose to quantify the performance gain or loss introduced by packet erasure coding, cooperative networking, and broadcast communication. Therefore, the tests were designed in a way to minimize the impact of routing and membership management protocols.

The rest of the paper is as follows. Section II provides a survey of reliable multicast protocols for MANETs. Section III briefly presents the features of the protocols considered in our evaluation. Section IV describes the experiments that we devised to compare the reliable multicast communications middlewares and presents the results that we obtained, as well as the adopted configuration options for each middleware. Finally, Section V provides concluding remarks and discusses future work.

II. Reliable Multicast Protocols for MANETs

The "reliable multicast" term identifies a very broad set of communication protocols that might significantly differ for aspects such as application domain, scalability, and even level of reliability they provide for message delivery (e.g., totally reliable or semi-reliable [10]). A generic survey of reliable multicast can be found in [1], while [2] provides a more specific overview of reliable multicast designed for MANETs.

This paper considers only totally reliable protocols. Reliable multicast protocols can be classified into one of four categories depending on the mechanisms they adopt to ensure reliable information delivery. These four categories are: Automatic Repeat Request (ARQ) based, gossip-based, Forward Error Correction (FEC) based, and hybrid.

ARQ-based protocols use acknowledgments sent by the receiver and possibly timeouts, to ensure reliable data transmission. They can be classified further as sender-initiated or receiver initiated, depending on which party is in charge of detecting missing packets. In the former case, the sender detects the missing packets based on the reception of acknowledgment messages (ACKs) from the receivers. In receiver-initiated protocols, the receivers explicitly send the list of missing packets (NACKs) to the sender upon the reception of a new, out-of-sequence packet. Hybrid ACK/NACK approaches are also possible. If no action is taken to limit the number of ACKs or NACKs, ARQ-based systems may not scale when the size of the network increases. Furthermore, unless local-repair strategies are used, the sender is exclusively in charge of the retransmission of repair traffic.

In Gossip-based approaches, data are sent in a peer-to-peer fashion, where each node receiving data may choose to forward them to a subset of its neighbors. This decision can be probabilistic or it may be heuristic-based, e.g., taking network topological details into account. The main disadvantage of this approach consists in the non-deterministic delivery.

In FEC-based approaches, forward error correction is used to send redundant data (called *parity*) in such a way that even in case of some packet loss, the receivers may still be able to reassemble the data because of the redundancy. Erasure coding is a particular form of FEC that combines a set of k symbols to obtain a larger set of n transformed symbols for which there exists a subset of symbols of size h that can be used to recover the original set. If any subset of size h of the transformed set is sufficient to recover the original set, the code is called optimal. FEC cannot ensure reliability by itself, but it can be coupled with an ARQ-based approach to reduce the packet loss, which results in fewer retransmissions. NORM is a good example of a hybrid approach that integrates ARQ and FEC. These approaches allow the ARQ mechanism greater efficiency when providing repair traffic; for instance, when optimal erasure codes are used, a single repair packet can repair different packets at different receivers.

Erasure coding encoding techniques are end-to-end, meaning that they are encoded at the source and the decoded at the receivers. Different coding schemes, known as network coding, allow the intermediate nodes that receive a message to re-encode it. Ananalysis of the performance gain introduced by erasure coding and network coding when adopted to perform reliable multicast in ad-hoc networks can be found in [19].

Another essential aspect of reliable multicast communication protocols is their reliance on underlying routing protocols. Multicast routing protocols build either routing trees or meshes, which are costly to build and maintain in highly mobile networks. Many reliable multicast protocols, with the notable exception of Route Driven Gossip (RDG)described in [13], need group-communication primitives to work. In RDG all the traffic, including membership management, route building, data and NACKs is transmitted by gossip. A comprehensive survey of multicast routing protocols for MANET can be found in [3] and in [4].

Finally, support for congestion control represents a critical aspect of reliable multicast protocols. The critical impact of congestion control in MANET when performing multicast and possible solutions by means of rate control have been extensively studied in [7], [8] and [9]. In this paper we focus on approaches designed to perform well in tactical network environments. In particular we consider 2 approaches based on forward-error-correction and cooperative caching in order to improve the efficiency of hop-by-hop communication.

III. Background

This section provides more in depth background on the systems examined / evaluated in this paper.

A. NORM

NORM (NACK-oriented reliable multicast protocol) implements a hybrid receiver-initiated ARQ-based (even though NACKs can be solicited by the sender) multicast protocol that uses packet erasure coding. The packet coding is integrated with the ARQ mechanism. While the packet erasure coding can be provided proactively, it is also possible to configure it as reactive. In this latter case, FEC packets are sent only upon a NACK reception. In this way, the overhead introduced by the FEC mechanism is limited, at the price of slightly higher latency. A tradeoff between bandwidth utilization and latency can be reached by using a hybrid approach that uses a low level of proactive FEC and reactive FEC when necessary. NORM computes FEC over blocks of data; every block of data is then divided in a certain number of segments where a certain percentage of these segments contains FEC data. The number of the segments containing FEC data is configurable. NORM is based on previous work on the Multicast Dissemination Protocol (MDP) Toolkit [8] that was designed to support reliable group transmission in dynamic environments. It implements NACK suppression and adaptive timers to ensure scalability. NORM has three modes of operations: NORM_OBJECT_DATA, NORM_OBJECT_FILE, and NORM_OBJECT_STREAM. The former two modes provide reliable transport of "finite units of data", whereas the latter provides "non-finite streams of continuous data" [21]. All the modes allow for late-joining receivers, but in the last case the application must mark message boundaries so that late receivers can join the stream from a meaningful state. Furthermore, NORM presents forms of congestion control and flow control. Congestion control is enforced either through a "TCP Friendly" algorithm or through an Explicit Congestion Notification algorithm. In order to perform congestion control, NORM uses specific control packets that are part of the NORM "command" set; this set can be extended by the application to tailor the end-to-end flow control as necessary.

B. DisService

DisService is a middleware that implements a peer-to-peer message-oriented publish-subscribe dissemination service for tactical networks.

Given the assumption that, in wireless communications, the cost of a (local) broadcast is equal to the cost of unicast, DisService relies solely on broadcast for both data and control messages. The use of broadcast lets neighboring peers receive messages even when they are not the target of the transmission. Peers can store received packets in a cache, and re-transmit them to other peers if requested, thus actively participating inthe repair of missing messages, fragments, or broken communication paths. We call this feature *Opportunistic Listening*. To facilitate peers' decisions on whether to save received messages in their cache, every published message has attached metadata that makes the message self-describing and self-contained, therefore interpretable by any peer. DisService supports any combination of reliable/unreliable and ordered/unordered communication: these requirements are group-based and are specified by the client application when subscribing to the group. To ensure reliability, DisService uses a receiver-initiated ARQ-based mechanism based on Probabilistic NACKs. By "probabilistic" we mean that a single NACK (Missing Fragment Request or MFR in DisService terminology) message specifies a random subset of the missing messages/fragments in order to minimize requests for the same message/fragment in subsequent requests. Every message/fragment is assigned a unique identifier that includes the group that the message belongs to, the ID of the node that published the message, and a sequence ID. Every node keeps track of the messages that have been completely received; therefore it is straightforward for a receiver to identify the missing messages/fragments. By default, messages with sequence IDs lower than the sequence ID of the first message received will not be requested, however the header of every message contains the recommended number of prior messages the peer may want to retrieve to make sense of the message. The decision whether to retrieve "history" messages is up to the subscribing application(s).

NACKs are sent by broadcast, and they are sent only when the incoming traffic is a configurable fraction of the capacity of the interface. Missing messages/fragments are requested until they are received, but a backoff mechanism is used to increment the interval in which a missing message/fragment is requested. Upon arrival of a new peer, or new data, these timers are reset. Since the MFRs are broadcast, the opportunistic listening feature allows potentially many peers to serve the request. In order to avoid multiple peers replying to the same request, DisService keeps track of all the messages that have been transmitted on the network (this is possible because of the broadcast communication) in the last time window, and stores their IDs in a data structure called Network Traffic Memory (NTM). If a MFR is received and the requested message or fragment is in the NTM, the request is not served. A random interval of time is allowed to elapse before a MFR is processed. In this way it is unlikely that multiple peers serve the MFR at the same time, thus voiding the effect of the NTM. DisService does not rely on any multicast protocol; instead it uses broadcast and probabilistic flooding or epidemic protocols to spread the information. Alternative routing protocols may be implemented and plugged into the system. Analogously, DisService peers do not require exchange of membership knowledge. However, alternative solutions that share and use membership knowledge could be incorporated. DisService offers the ability to register custom modules (called controllers) to control the forwarding, replication and the deletion of the cached messages. Finally DisService offers a rudimentary form of congestion control by monitoring the incoming traffic and limiting the number of NACKs.

C. OpenPGM

OpenPGM is an implementation of PGM, which is a receiver-initiated, ARQ-based reliable multicast protocol supporting ordered and unordered, duplication-free multicast data delivery. The reliability is provided only within a receiver-managed transmit window and the repair traffic is served either by the sender of the message or by designed local repairing nodes (DLR). When a receiver detects missing packets those are requested by sending a NACK to the source by unicast. NACKs are propagated reliably to the source following the distribution tree of the source. This usually has the effect that only a single copy of the NACK is received by the source.

Furthermore when a NACK is received, the receiving node will toggle to repair state; in this way, when the requested repair data is received, the node will keep propagating it along the distribution tree. Conversely, if the repair state is not set, the node will not propagate the data. This ensures that repair traffic is sent only along the branches that reach the requesting node.In addition to the ARQ mechanism, PGM senders can also use forward error correction.

OpenPGM also performs rate regulation and tries to guarantee flow fairness using a TCP Friendly congestion control algorithm.

D. JGroups

JGroups [17] is a toolkit for reliable multicast communication and it is part of the JBoss suite of Java middleware. JGroups provides a configurable protocol stack that lets the user choose between the different protocols of the channel. A channel, in JGroups jargon, is a group communication end-point. The channel is comprised of several blocks that implement network protocols. JGroups offers different alternative protocols for each block. The stack's blocks are, in top-down order as follows: flow control, group membership, reliable delivery, failure detection, discovery, transport.

IV. Quantitative Evaluation

Two different sets of experiments were conducted to measure the performance of reliable multicast protocols and implementations. All of the experiments were conducted using the NOMADS Tactical Network Emulation Testbed, which utilizes a modified version of the Mobile Ad-hoc Network Emulator (MANE) [15]. The testbed allows the evaluation of the performance of the different systems in a reproducible and controlled laboratory environment. The Testbed allows independent control of the capacity, reliability, and latency of each network link.

One aspect of the behavior of MANE worth noting is the implementation of reliability. In MANE, reliability is independently enforced for each link between a pair of nodes. Furthermore, reliability is enforced randomly on a packet-by-packet basis. The implication is that there is no correlation between links, or between adjacent packets. For example, if the reliability between a sender and a set of receivers is specified as 80%, then on average, 80% of the packets are successfully delivered to each receiver, but there is no correlation between the behavior with each receiver (as might happen in a real-world scenario, when occlusion or interference might prevent a sender's packet from not being received by all the receivers simultaneously, or a contiguous set of packets being lost, as opposed to an independent random sample).

A. Baseline Test

The first test was a simple baseline test, which we designed in order to compare the performance of the middleware in a simple, symmetric configuration. We used a flat topology with a variable number of fully connected nodes. One of these nodes was designated as the sender and transmitted a fixed amount of data (800 KB) to the rest of the nodes in the network. There were either one or three receivers, depending on the configuration.

In terms of the network, we used different settings for the capacity and the reliability of the links. We collected data with the capacity of the links set to 56, 128, 256, 512, 1024 and 10240 Kbps. Those values of link capacity are of relevance because 56 Kbps is the typical capacity of a portable SATCOM link, whereas 256 Kbps models a UAV downlink and 512 Kbps models the typical bandwidth for high-capacity SATCOM link. Furthermore, we used combinations with the reliability set to 100%, 90%, 80% and 70%.

One final configuration action worth noting is that in each case, the protocol implementation was configured with the capacity of the link setup in MANE. While this assumption may not hold in a real deployment scenario, performing this configuration allowed the results collected to be more comparable across all the four systems.

1) NORM Configuration

For this test, we used NORM version 1.4b3. We configured NORM to use only reactive FEC and we set the rate limit to the link capacity. NORM computes FEC over blocks of data. The size of these blocks is configurable and it usually set to multiple of the network MTU (also called segment, in NORM terminology). We set the MTU to 1400 Bytes and the size of the blocks to 64 times the MTU.

2) DisService Configuration

In this test we configured DisService to send MFRs with a frequency which is a function of the nominal capacity of the link. The nodes send and reply to MFRs only if the network traffic is below 80% of the maximum link capacity and keep a history of recent messages. Since the network topology is fully connected, the nodes perform no forwarding. We configured the rate limit of the nodes to give priority to the sender, and split the remaining capacity evenly between the receivers. We set the payload size to be 1400 bytes for each message.

3) OpenPGM Configuration

We used OpenPGM version 5.1.115 for this test. We set up OpenPGM so that in case of missing messages the receivers send NACKs to the sender, which transmits without adding FEC overhead. We configured the total rate limit as high as the link capacity and the split between original data and repair data, to be 70% and 30% respectively. We set the payload size as 1400 bytes for each message.

4) JGroupsConfiguration

We used JGroups version 2.12.0.Final for this test. Since JGroups provides a configurable stack, we configured JGroups to use the *pbcast.nakack* algorithm as the reliable delivery protocol, which provides reliable delivery and FIFO ordering. This also guarantees that the messages are received in the order that they were sent (FIFO delivery is part of all the reliable options in JGroups). The receivers ask the sender for the missing messages. We configured JGroups to use UDP as the transport protocol. We set the rate limit to 75% of the link capacity, using the RATE LIMITER configuration parameter, to preserve part of the bandwidth for repair traffic. We also

Table 1 - Baseline Test 1-to-1

Reliability %	Link Cap. (Kbps)	NORM		DisService		OpenPGM		JGroups	
		Time (s)	Bytes	Time (s)	Bytes	Time (s)	Bytes	Time (s)	Bytes
100	10240	2.5	942964.8	0.7	907356.0	9.8	850125.0	0.8	889002.0
100	1024	6.6	864440.0	6.9	908163.0	13.5	866651.0	7.1	889442.7
100	512	13.3	864588.0	13.7	908970.0	21.8	850125.0	15.5	888426.5
100	256	26.4	864884.0	27.6	910853.0	42.7	850125.0	31.7	895479.7
100	128	52.6	865117.6	55.3	914709.0	78.5	850125.0	63.4	901369.3
100	56	120.2	865476.0	126.2	924483.0	183.7	850125.0	155.7	910909.5
90	10240	7.2	1032832.4	3.1	1033143.0	2.6	880949.4	3.5	1000539.5
90	1024	10.6	955303.6	10.7	1002889.0	12.5	985418.2	9.6	997163.0
90	512	17.4	961341.2	18.9	1004696.0	24.7	949455.4	15.8	1016195.5
90	256	32.2	972123.2	38.9	1024240.0	41.0	957794.2	33.1	999644.0
90	128	59.2	963196.4	84.6	1008027.0	77.1	961534.6	64.7	1032559.0
90	56	135.2	969608.4	164.3	1031373.0	171.1	960520.0	155.7	1052703.0
80	10240	10.6	1138095.6	6.5	1122470.0	5.1	922778.8	13.0	1619951.5
80	1024	13.9	1072339.2	17.6	1105450.0	13.2	1103939.8	14.7	1116129.0
80	512	26.2	1100016.8	31.5	1130841.0	25.6	1099152.2	28.8	1514743.7
80	256	34.9	1073580.5	55.3	1141901.0	40.9	1104456.5	61.5	1201655.7
80	128	67.5	1093099.6	131.2	1136655.0	78.3	1093053.0	65.0	1150092.7
80	56	153.0	1091555.2	197.2	1134085.0	169.8	1081134.2	156.3	1197473.5
70	10240	16.8	1271891.0	9.5	1329893.0	8.4	975493.6	10.2	1299267.0
70	1024	20.3	1243872.0	34.5	1293379.0	15.5	1230398.8	18.4	1343783.0
70	512	28.8	1325833.0	64.6	1335993.0	22.6	1262850.8	24.1	1341610.0
70	256	45.5	1219135.0	108.4	1292584.0	43.4	1261431.0	38.9	1858443.5
70	128	81.4	1262697.2	131.0	1329647.0	76.1	1243386.0	66.8	1350325.0
70	56	172.3	1208168.5	268.0	1327894.0	212.3	1192103.0	159.4	1425371.0

enabled the *frag2* configuration parameter in order to instruct JGroups to fragment messages at application level. This is necessary because MANE applies drop probability to each individual packet. So, letting UDP fragment the messages at the IP level decreased performance significantly, as the loss of one fragment would cause the whole message to fail.

Table 1 shows the results for different values of capacity and reliability in the case of 1sender and 1receiver. Analogously, Table 2 shows the results for different values of capacity and reliability in the case of 1sender and 3 receivers. The first two columns on Tables 1 and 2 state the reliability and the capacity of the links; the following columns report the measured values of average delivery time (expressed in seconds) and the total bandwidth utilization (expressed in bytes) for each of the studied systems.

The 1-sender-1-receiver (1-to-1) test shows similar results bandwidth-wise for all the protocols, whereas time-wise, NORM shows a slight advantage over the other protocols. The 1-to-3 tests confirm NORM's advantage in terms of both bandwidth and average delivery time. JGroups and DisService are less bandwidth efficient. For JGroups, the problem is finding a good ratio between original data and repairs. This baseline test does not present any network disconnection or asymmetric network links that DisService was designed to support, thus resulting in sub-optimal performance. In particular, the extra metadata attached to every packet in order to make each packet self-describing and self-sufficient adds some bandwidth utilization. Furthermore, the ability for DisService to allow any peer to respond to missing fragments adds some additional overhead while not providing any benefit in this simplistic scenario. The one exception is when the link capacity is set to 10 Mbps – in which case DisService consistently outperforms NORM time wise. It should also be noted that even in this simplistic scenario, when bandwidth and reliability are more constrained, JGroups and OpenPGM were not able to fully complete the reception of the message in a time comparable with NORM and DisService; therefore measurements of their performances in these cases are not reported.

B. Tactical Network Test

This second test evaluates and compares the performance of NORM and DisService when the sender communicates through an asymmetric and constrained link. We used a similar topology to the baseline test, but we increase the number of peers to 10, while keeping the number of receivers, in other words, the number of peers interested in retrieving the information being published, to 3. Analogously to the previous tests, the 10 nodes are fully-connected.

Table 2 - Baseline Test 1-to-3

Reliability %	Link Cap. (Kbps)	NORM		DisService		OpenPGM		JGroups	
		Time (s)	Bytes	Time (s)	Bytes	Time (s)	Bytes	Time (s)	Bytes
100	10240	2.5	948150.0	0.7	907594.0	9.6	1989130.0	0.9	892293.8
100	1024	6.6	864849.5	6.9	908155.0	15.0	955917.0	7.6	888637.8
100	512	13.3	865192.5	13.7	909196.0	26.4	850125.0	16.9	892303.3
100	256	26.4	865742.0	27.8	919999.0	44.4	850125.0	39.5	897310.4
100	128	52.6	866174.5	55.6	914803.0	79.7	850125.0	64.6	908170.5
100	56	142.7	880559.5	176.3	930912.0	174.9	850125.0	126.2	916631.3
90	10240	9.6	1086475.5	4.9	1209148.0	3.7	948204.3	6.8	1246315.0
90	1024	12.2	1007351.5	18.5	1306889.0	14.3	1253240.0	11.5	1241616.7
90	512	18.3	997283.0	22.1	1371243.0	26.1	1159804.3	56.8	1342400.0
90	256	31.6	992976.5	41.2	1317697.0	64.0	1244235.0	64.6	1253253.0
90	128	61.2	1005547.0	83.8	1329992.0	85.7	1267727.8	#N/A	#N/A
90	56	140.5	1020322.7	289.5	1542430.0	173.5	1144592.3	#N/A	#N/A
80	10240	12.7	1228000.0	3.8	1605987.0	12.0	2057962.3	8.9	1658596.0
80	1024	16.3	1179473.5	19.4	1731909.0	19.0	1520958.8	16.5	1906684.5
80	512	21.7	1139268.0	36.9	1868916.0	31.0	1504776.5	27.2	1634349.0
80	256	38.2	1159159.5	50.7	1756663.0	50.4	1505103.8	66.7	1649083.0
80	128	72.1	1150811.5	163.3	1997177.0	95.0	1493852.8	#N/A	#N/A
80	56	161.0	1144260.5	341.8	2179191.0	260.2	1276224.0	#N/A	#N/A
70	10240	14.7	1476802.0	6.7	2024430.0	11.2	1203126.0	#N/A	#N/A
70	1024	20.0	1437628.7	25.6	2427424.0	19.5	1853620.3	47.0	2202264.0
70	512	29.9	1439708.0	43.0	2076855.0	66.2	2505978.0	159.7	8793292.0
70	256	52.0	1411541.0	92.0	2248010.0	136.2	1714531.0	#N/A	#N/A
70	128	89.3	1435667.0	156.6	2242740.0	#N/A	#N/A	#N/A	#N/A
70	56	204.0	1441158.0	349.6	2729084.0	#N/A	#N/A	#N/A	#N/A

However, in this case, the link between the sender and any other node is constrained to 230Kbps. We experimented with two different configurations of the sender-to-receiver links, configuring it as symmetric in the first run, and as asymmetric in second one. When the link is configured as asymmetric the sender is not able to receive and therefore serve NACKs/MFRs. Conversely the receiving nodes are always connected to each other by relatively reliable and capable links. The reliability among the receivers is 90% and they are fully connected by 1 Mbps links. In this second test we did not consider OpenPGM and JGroups. The unsuitableness of these two protocols is obvious in the asymmetric link case, because of the reliance on the source to perform retransmission and because they do not have any other methodology to provide reliability. As for the second case, with symmetric sender-receivers links, it is almost analogous to the baseline test with 3 receivers and 256 Kbps of link capacity. The larger number of peers does not influence the test because the peers interested in retrieving the published information are still limited to three. Moreover, even though the receiver peers are allowed to exchange information, these peers do not take advantage of this possibility. We can therefore conclude that the results of the baseline test with 3 receiving peers for OpenPGM and JGroup would be an upper bound for the tactical network test.

This tests models several possible scenarios, in which an entity (such a UAV) reaches a squad deployed in the battlefield to deliver updated information. This entity may be allowed only asymmetric communication, or, because of the limited amount of time, it is configured not to provide repair traffic.

As with the baseline test, we ran the tactical network test under different reliability settings. While the reliability of the link among the receiving nodes is fixed to 90%, the reliability between the sender and the any of the receivers was set to 90%, 80%, 70% and 50%. Furthermore, in this test, the sender transmitted messages of varying sizes - 1024, 7168, 15360 and 35840 Bytes. However, since the UAV had a fixed loiter and communications time, the total amount of data transmitted is approximately the same (i.e., with larger messages, fewer are sent within the available time window).

1) NORM Configuration

Because of the different constraints on the link between the sender and every other receiver, we experimented with two different configurations for NORM: when the sender link was configured as symmetric, we configured NORM with the same configuration we used in the baseline test. In particular, when configured to use proactive FEC, NORM was configured with a FEC overhead of approximately 33%, but the actual value depends on the message size (for

Table 3 - Tactical Network Test - 3 Subscribers

Message Size (Bytes)	Messages Sent			Messages Received			Success Rate (%)			Sender Overhead (Bytes/messages)			P2P Overhead (Bytes/messages)		
	DS	*NORM*		*DS*	*NORM*		*DS*	*NORM*		*DS*	*NORM*		*DS*	*NORM*	
		pfec	*rfec*		*pfec*	*rfec*		*pfec*	*rfec*		*pfec*	*rfec*		*pfec*	*rfec*
Sender Link Reliability – 90%															
1024	1578.0	1604.5	1274.5	1577.7	1453.2	1272.5	100.0	90.6	99.8	139.0	122.2	417.0	558.2	24.7	32.8
7168	228.0	170.0	189.0	228.0	157.5	188.5	100.0	92.6	99.7	834.0	3564.5	2487.2	4050.9	25.5	175.5
15360	108.0	87.0	93.0	107.0	81.3	92.5	99.1	93.5	99.5	1551.1	5542.2	4191.5	7765.0	33.9	267.9
35840	46.0	36.0	40.5	46.0	35.3	40.0	100.0	98.1	99.8	3753.0	14700.6	9063.6	36798.0	54.0	410.5
Sender Link Reliability – 80%															
1024	1578.0	1599.5	978.0	1577.7	1275.7	975.7	100.0	79.8	99.8	139.0	126.0	853.9	909.8	35.9	60.7
7168	228.0	167.0	157.0	228.0	115.5	155.5	100.0	69.2	99.0	834.0	3768.5	4447.1	6990.5	107.3	388.9
15360	107.0	86.0	81.0	107.0	50.8	80.0	100.0	59.1	98.8	1668.0	5803.2	7093.9	13373.9	142.8	425.7
35840	46.0	36.0	35.0	46.0	22.8	34.5	100.0	63.4	98.6	3753.0	14708.2	16123.1	32581.4	156.4	574.3
Sender Link Reliability – 70%															
1024	1578.0	1608.5	812.5	1576.3	1129.3	808.3	99.9	70.2	99.5	139.0	119.4	1236.4	1286.9	44.1	82.4
7168	228.0	166.0	129.5	228.0	74.3	127.0	100.0	44.8	98.0	834.0	3838.9	6911.5	8553.7	176.3	431.2
15360	108.0	84.0	67.5	107.0	26.7	66.3	99.1	31.7	98.3	1551.1	6308.5	11570.8	18.423.7	284.2	576.5
35840	46.0	34.5	30.0	46.0	9.5	29.5	100.0	27.5	98.3	3753.0	16928.2	24778.2	41625.2	435.2	769.4
Sender Link Reliability – 50%															
1024	1578.0	1615.5	528.0	1574.0	810.2	517.8	99.8	50.1	98.0	139.0	114.1	2454.4	1634.0	64.6	171.0
7168	228.0	166.0	84.0	227.0	21.2	81.2	99.6	12.8	96.6	834.0	3835.0	14512.5	11020.9	347.7	707.8
15360	108.0	82.0	46.0	106.0	5.2	44.0	98.2	6.3	95.7	1551.1	6839.9	24176.7	22086.6	520.2	978.9
35840	46.0	32.3	20.0	46.0	3.7	19.7	100.0	11.4	98.3	3753.0	21030.1	55108.0	53386.5	910.0	1501.4

example, for messages of 7168 bytes, the FEC overhead is 2 segments, which makes the real overhead 39%).

Messages of 1 KB had no FEC because the smallest overhead that would have been added was larger than the message itself. We also took into account the possibility to use NORM in stream mode. However, stream mode only supports ordered delivery, which means that a received message can be delivered only if all the previous messages have been delivered already. Because the test was bound by a time limit, this option would have potentially penalized NORM's performance.

2) DisService Configuration

The configuration is the same as in the first test, except that the sender is configured to avoid replying to MFR. In contrast to NORM, DisService was configured not to serve the MFRs for both the symmetric and asymmetric link scenario, thereby taking advantage of cooperative networking. The nodes share the network leaving the sender all the bandwidth needed and sharing the rest of the bandwidth equally among the receivers.

Because of the reliance on the source node for repairing any messages, and considering the suboptimal performance with lower levels of reliability and a high number of nodes in the baseline test, we observed that OpenPGM and JGroups would have not performed well in the tactical network tests; therefore we did not consider them in this context.

Table 3 shows the results we obtained for the tactical network tests. The DS columns report the results for DisService, whereas the NORM-pfec columns report the results for NORM configured to use proactive FEC and NORM-rfec columns report the results for NORM configured to use only the reactive FEC. Due to space constraints in this table, we do not report delivery time data. Instead we focus solely on the average number of messages that were completely delivered and the total bandwidth utilization.The tactical network test shows that in a case where the sender transmits through an asymmetric link (or for some reason is not able to provide repair traffic), the cooperative approach of DisService has better performance than FEC in terms of both success rate and in number of messages sent, especially in the case of larger messages. The high success rate is due to the cooperation among the receiving peers. The probability that all of them miss a fragment is very low. The reason for the higher transmission rate of the sender is simply that the sender does not use up any bandwidth sending FEC packets. The benefits of the cooperative approach become more evident as the reliability decreases. That is easily explainable and the reason is that even with low reliability, given a sufficient number of receivers, it is very unlikely that all of them lose the same fragment. With FEC instead, the overhead to overcome such a high packet loss would be very large. When the sender transmits though the symmetric link and the reliability of the network is relatively high (80%), NORM shows results similar to DisService, in terms of number of sent and received messages and the total overhead is

comparable or lower to the one introduced by DisService. However, when the reliability decreases, the NORM sender only manages to send a considerably lower number of messages then DisService. The reason is that the NORM sender spends a considerable amount of time serving repair traffic; in DisService instead, the burden of the repair is completely handled by the cooperating receivers, and the sender will be able to send new data instead of repair traffic. In both scenarios, symmetric and asymmetric link, the overhead introduced by repair traffic served by DisService receivers is quite high. However, this communication takes place over the P2P ground network between the receivers, which has higher capacity.

V. Conclusions And Future Work

In this paper we evaluated NORM and DisService, two protocols (with corresponding implementations) for reliable multicast communications in tactical network environments. We also compared their performance to two state-of-the-art COTS applications – OpenPGM and JGroups. The results show that even in relatively stable wireless scenarios, as the reliability of the network decreases, NORM and DisService either perform better or show comparable performance. In particular, in the context of a network with symmetric links, we observed the advantage offered by the integration of the erasure coding and ARQ mechanisms implemented in NORM over the approach that exploits cooperative networking implemented in DisService. The primary reason is that multiple receivers have the possibility to provide repair traffic and the bandwidth sharing among the peers and with the sender is not optimal. We are currently working on further optimizations to DisService to improve performance in the baseline case. Further tests showed that cooperative networking is a viable solution to handle more constrained networks. In case of an asymmetric link from the sender, it is not possible to rely on its repair traffic; therefore it is necessary to configure NORM to use high levels of forward error correction that introduces high overhead both in terms of bandwidth and delivery time. In this case, the redundant data needed to perform the FEC is sent over the more constrained link. On the other hand, DisService is able to retrieve missing messages from the neighboring clients, and given a sufficient number of receiving nodes, it is able to reassemble almost all of the messages. Furthermore, unlike NORM, no additional stress is added on the most constrained link. We are currently developing more sophisticated (and more realistic) scenarios involving network disconnection, a common occurrence in tactical environments. DisService, by its very nature, is disruption tolerant and hence should provide good performance in comparison with the other approaches. We also continue to improve the performance of DisService in the baseline case, even though that is not an ideal scenario for DisService. We are also going to consider larger networks and consider mobility and routing in future experimentation.

Acknowledgment

This work is supported by the Army Research Lab under grant W911NF-04-2-0013 and the Office of Naval Research under grant N00014-09-1-0012.

References

[1] A. Popescu, D. Constantinescu, D. Erman, and D. Ilie, “A Survey of Reliable Multicast Communication”, Next Generation Internet Networks, pp. 111-118., May 2007.

[2] B. Ouyang, X. Hong, “A Comparison of Reliable Multicast Protocols for Mobile Ad Hoc Networks”, in Proceedings ofIEEE SoutheastCon 2005, pp. 339-344.

[3] L. Junhai, Y. Danxia, X. Liu, F. Mingyu, “A survey of multicast routing protocols for mobile Ad-Hoc networks”, IEEE Communications Surveys & Tutorials, vol. 11, 2009, pp. 78-91.

[4] O.S. Badarneh, M. Kadoch, “Multicast Routing Protocols in Mobile Ad Hoc Networks: A Comparative Survey and Taxonomy,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, 2009, pp. 1-42.

[5] I. SM. Ge, S.V. Krishnamurthy, M. Faloutsos, “Application versus network layer multicasting in ad hoc networks: the ALMA routing protocol”, Ad Hoc Networks, vol. 4, Mar. 2006, pp. 283-300.

[6] N.Suri et al“Disservice: A peer-to-peer disruption tolerant dissemination service”, in Procedings of2009 IEEE Military Communications Conference.

[7] V. Rajendran, K. Obraczka, Y. Yi, S.-ju Lee, K. Tang, M. Gerla, “Combining Source- and Localized Recovery to Achieve Reliable Multicast in Multi-Hop Ad Hoc Networks”, NETWORKING, Springer, 2004, pp. 122-124.

[8] K. Tang, K. Obraczka, S.-J. Lee, M. Gerla, “Reliable adaptive lightweight multicast protocol”, in Proceedings of IEEE International Conference on Communications (ICC’03),, 2003.

[9] K. Obraczka, M. Gerla, “Congestion controlled adaptive lightweight multicast in wireless mobile ad hoc networks”, in Proceedings of 7thInternational Symposium on Computers and Communications (ISCC 2002), 2002.

[10] M. Handley, S, Floyd, B. Whetten, R. Kermode, L. Vicisano, M. Luby, “The Reliable Multicast Design Space for Bulk Data Transfer”, IETF Request for Comments: 2887, August 2000.

[11] J.P. Macker and P.B. Adamson, “The multicast dissemination protocol (MDP) toolkit”, in Procedings of1999 IEEE Military Communications Conference (MILCOM 1999), pp. 626-630.

[12] B. Adamson and J.P. Macker, “Reliable Messaging for Tactical Group Communication”, in Procedings of2010 IEEE Military Communications Conference (MILCOM 2010), pp. 1439-1444.

[13] J. Luo, P.T. Eugster, J.-P. Hubaux, “Route driven gossip: probabilistic reliable multicast in ad hoc networks”, in Proceedings of IEEE INFOCOM 2003, pp. 2229-2239.

[14] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, L. Vicisano, PGM Reliable Transport Protocol Specification, IETF Request For Comments 3208, December 2001.

[15] N. Ivanic, B. Rivera, B. Adamson, “Mobile Ad Hoc Network Emulation Environment”, in Proceedings of 2009 IEEE Military Communications Conference (MILCOM2009).

[16] http://code.google.com/p/openpgm/

[17] B. Ban, “Design and Implementation of a Reliable Group Communication Toolkit for Java”, Architecture, 1998, pp. 1-14.

[18] N. Suri, G. Benincasa, M. Tortonesi, C.Stefanelli, J. Kovach, R. Winkler, R. Kohler, J. Hanna, L. Pochet, S. Watson, “Peer-to-Peer Communications for Tactical Environments: Observations, Requirements, and Experiences”, IEEE Communications Magazine,, Vol. 48, No. 10, October 2010.

[19] A. Fujimura, O.Y. Soon, and M. Gerla, “Network Coding Vs. Erasure Coding: Reliable Multicast in Ad HocNetworks”, in Proceedings of 2008 Military Communications Conference (MILCOM 2008).

[20] X. Shen et al., “Handbook of Peer-to-Peer Networking”, Springer, 2009.

[21] B. Adamson, C. Bormann, M. Handley, J. Macker, “NACK-Oriented Reliable Multicast (NORM) Transport Protocol”, IETF Request for Comments: 5740,2009.