Survivability strategies in multihop wireless networks

Osameh M. Al-Kofahi Ahmed E. Kamal
Department of Electrical and Computer Engineering, Iowa State University
e-mails:{osameh, kamal}@iastate.edu

***Abstract*—Survivability is an important network characteristic that provides a certain level of data delivery guarantees. The degree of survivability is usually governed by the data transfer mechanism or protocol that delivers data from source to destination. In this paper, we survey and discuss a variety of survivability issues, challenges and mechanisms in multihop wireless networks. Unlike some of the previous surveys, we do not focus only on multipath routing techniques. We try to cover a broader spectrum of survivability techniques in the literature. Moreover, we discuss new directions in survivability that uses the network coding technique in order to achieve a better degree of scalability, which is usually an issue in most survivability techniques, especially in wireless networks.**

I. Introduction

Multihop wireless networks, such as ad-hoc, sensor and mesh networks, have drawn a lot of attention in the last decade, and will continue to be an important research topic in the future also. Applications for these types of networks are numerous and diverse ranging from military to public safety, health and environmental applications. The most important merit of multihop wireless networks, which makes them very attractive is their ease of deployment compared to wired networks that need a pre-installed infrastructure to operate. However, this flexibility compromises the robustness of these networks. For example, the nodes in a sensor or ad-hoc network usually have limited power supply, which causes the nodes to die out and interrupt the network information flow or reduce network connectivity. Moreover, the wireless communication medium is prone to various types of interference and impairments causing a wireless link status to dynamically change according to the channel conditions, and thus causing the wireless links to be intermittently unavailable. Besides interference and impairments, the harsh surrounding environments and severe weather conditions may damage either nodes or links (e.g, a damaged antenna) if the network is deployed outdoors as in the case of sensor and mesh networks. These problems emphasize the need for mechanisms to enhance the network survivability.

Survivability is usually defined as *the capability of a network to deliver data successfully in a timely manner, even in the presence of failures*. Network survivability is important to sustain continuous uninterrupted service for the network users, and is crucial to maintain the quality of this provided service. Although survivability is defined as a network property, its realization is coupled with a data transfer session. In every session there is a sending side and a receiving side, each of which may consist of one or more network nodes. In general, network survivability methods can be divided into the following categories:

- *Protection mechanisms:* [1]-[2]. Protection is usually achieved by using redundant network resources to carry redundant data units. Usually, a data unit is duplicated and forwarded on multiple paths from the source to the destination. In this case, a data delivery failure occurs and will be detected only if all paths fail. Otherwise, there is no need to detect the failure or retransmit the information. This is called *proactive protection* and is usually referred to as 1+1 protection. An alternative way to provide protection is to divide the paths into two sets, primary and backup, where only the primary path is used to forward data to the destination. A backup path will only be used if the primary path fails. This is called *reactive protection* and is usually referred to as 1:1. Reactive protection can be extended to M:N, where M backup paths are reserved to protect N primary paths. The M backup paths are shared by the N sources, and can be used by any source if a failure occurs on its primary path, which makes this type of protection more efficient in utilizing the network resources. However, reactive protection is slower than proactive protection since a source must detect a failure first, and then switch the data flow to one of the available backup paths.
  Although reactive protection is known and viable in wired networks, it is not technically accurate to talk about path reservation in wireless networks, since there are no actual physical links that can be reserved. However, a node in a wireless network might learn multiple paths to the destination during the route discovery process and can use them in a fashion similar to that of reactive protection. That is, all paths are known a priori and will be used as needed, but without being reserved in advance.
- *Restoration mechanisms:* [3]. In restoration mechanisms only a single path is used from a source to a destination, and no backup paths are found in advance. Therefor, restoration mechanisms consume fewer resources than protection mechanisms. However, restoration does not provide recovery at the speed of protection. This is because failures need to be detected first (unlike proactive protection), after that a resource discovery procedure is invoked (unlike protection techniques in general), and finally rerouting is done to find a different route for the data units. Note that the rerouting mechanism here is

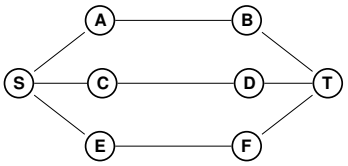

Fig. 1. Graph G: 3 node-disjoint paths between S and T

different from that in reactive protection. In restoration, no information about the available network resource is known to the node that detected the failure, that is why it needs to discover the network resources first to be able to do the rerouting afterwards. However, under protection, multiple paths are computed a priori, and thus, the rerouting mechanism in reactive protection is very simple and is confined to just switching to an available path from the backup set. Finally, it should be noted that restoration is implicitly implemented in all routing protocols in the form of route maintenance mechanisms.

- *Hybrid mechanisms:* [4]-[5]. In this case a mix of protection and restoration mechanisms can be used together.
- *Coding-based mechanisms:* [6], [7]. These approaches aim to reduce the proactive protection overhead without compromising the recovery speed, by utilizing erasure codes or network coding. If there are only 2 available paths between the source and destination, then coding-based approaches cannot do better than duplication. Consider the example in Figure 1, where there are 3 paths between S and T. In this case, using duplication we cannot protect more than 1 data unit, since to protect 2 data units we need 4 disjoint paths (two for each data unit). However, if network coding is used, two data units can be protected together using only 3 paths (i.e., 25% less resources). To do this, the source sends 2 data units, one to node C and the other to node E, and because of the wireless multicast advantage, node A overhears both transmissions and sends their sum (bitwise XOR) to the destination on the third path. This way, the destination receives 3 equations in two unknowns, where any 2 equations are solvable and are enough to recover the original data units.

The rest of the paper is organized as follows. Survivability issues and challenges are discussed in Section II. A survey of some of the survivability mechanisms is presented in Section III. In Section IV, we compare traditional protection techniques to coding-based techniques. Finally, the paper is concluded in Section V.

II. Survivability issues and challenges

In this section we consider the challenges and issues related to the different survivability mechanisms.

A. Scalability

The scalability challenge rises mainly in proactive protection mechanisms. This is because survivability is provided through using redundant network resources to forward redundant data units. There are two problems in such schemes. The first one is the problem of wasted resources. For example, to provide survivability against k-1 failures, at least $\frac{k-1}{k}\%$ of the network resources used in the communication session will be wasted to provide the required redundancy. The second one is the problem of the produced overhead. The high overhead produced from duplication may affect the network performance and lead eventually to congestion, which becomes more notable as the number of protected sessions increases. In other words, traditional proactive protection approaches do not scale well as the number of communication sessions increases.

To mitigate the effects of duplication, erasure codes or network coding can be used. The main advantage of these techniques is that duplication is eliminated, and thus, the useful throughput is increased. In erasure codes a data unit is encoded into $n + e$ smaller sub-packets that are forwarded on $n + e$ node-disjoint paths to the destination. It is enough for the destination to receive n out of these sub-packets to recover the original data unit, i.e., e failures can be tolerated. In network coding, the coding process is not done at the source, rather the intermediate network nodes are responsible for creating the required combinations. In addition, the created combinations are created from whole data units or packets, and not sub-packets as in erasure codes. If the S-T max-flow equals $n+e$, then a proper network code can be designed to deliver $n+e$ combinations to the destination. These $n + e$ combinations are created from n data units, such that any n of the $n + e$ combinations are solvable. Since different data units are sent in the network coding case, the useful throughput of network coding is even better compared to erasure codes. This will be discussed later in Section IV.

B. Network connectivity

Network connectivity is defined as the minimum max-flow between any two nodes in the network, which is equivalent to the minimum link cut between any two nodes in the network. The definition can be extended to cover the minimum node cut also. That is, network connectivity is defined as the minimum number of nodes (or links) that when removed (e.g., due to a failure) the network will be divided into two components A and A', such that no node in A is connected to a node in A' and vice versa. Alternatively, a network is said to be connected if there exists a path between any pair of nodes in the network. Furthermore, the definition can be extended to k-connectivity, where a network is said to be k-node (link) connected if there exists k node (link) disjoint paths between any pair of nodes in the network.

Network connectivity is an important network property that directly affects the network survivability. This is because network connectivity is what limits the number of alternative paths that can be found between a pair of nodes. A certain level of network connectivity can be achieved using node deployment algorithms or satisfied through topology control strategies. Wireless sensor networks motivated the development of numerous such algorithms and strategies. This is because, in many scenarios, WSNs are assumed to be deployed in response to certain large-scale events, such as catastrophes, and thus the deployed network must have a certain level of

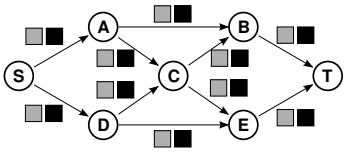

Fig. 2. 4 interleaving paths, and at most 2 are disjoint, i.e., $k = 4$ and $p = 2$

connectivity to guarantee successful data delivery under these conditions.

C. Disjoint Vs. Interleaving paths

Multipath routing is the mainstream approach to proactive protection mechanisms. In multipath routing k paths are found between a source node (S) and a destination node (T), these paths can be either node or edge disjoint, or they can be interleaving, i.e., some edges are shared. When a data unit is to be sent from the source to the destination, the source sends k copies of the data unit to the destination on the k paths. If the paths are disjoint, each of which forwards a single copy to T. This guarantees successful data delivery if failures take place on at most k-1 paths out of the disjoint k paths. However, all the copies will be lost if failures occur on all k paths.

If the paths are interleaving, a shared link does not forward all the copies from all the paths, it only carries one of them and the head node of the shared link duplicates the data unit on all of the outgoing paths. In the interleaving paths case, successful data delivery is guaranteed if failures take place on at most p-1 paths, where p is the maximum number of disjoint paths from the k S-T paths, and $p < k$ (if p=k the paths are disjoint). Unlike the disjoint case, interleaving may enhance the chances of the information to reach the destination even if failures occur on all the k paths. This is illustrated in Figure 2. In the graph four paths are found from the source to destination, namely, $P_1 : S \rightarrow A \rightarrow B \rightarrow T$, $P_2 : S \rightarrow A \rightarrow C \rightarrow E \rightarrow T$, $P_3 : S \rightarrow D \rightarrow C \rightarrow B \rightarrow T$, and $P_4 : S \rightarrow D \rightarrow E \rightarrow T$. Note that the choice of the paths is not unique, and other interleaving paths can be chosen. In the 4 paths above, P_1 shares link (S,A) with P_2, and link (B,T) with P_3. P_4 shares link (S,D) with P_3, and link (E,T) with P_2. To see the advantage of interleaving, assume that links (A,B), (A,C), (D,E) and (C,B) have failed. In this case node C will still receive a copy from D, and will send it to node E, which in turn will relay it to the destination.

D. End-to-End Vs Local Recovery

The recovery process is initiated once a failure is detected. Recovery can be done on an end-to-end basis or it can be done locally. In end-to-end recovery a node that detects a failure notifies the source by sending a specific message. Upon receiving this notification, the source is responsible for finding an alternative path to the destination. On the contrary, local recovery is initiated directly at the intermediate node that first detects the failure. In both cases, the alternative path might be stored in the source (or intermediate node) buffer already, or yet, needs to be discovered. This depends on the memory allocation for routing information at each node.

The main advantage of end-to-end recovery is that it provides the best (i.e., least cost) alternative S-T path, since the scope of the search for a new path is from the source to the destination. However, in end-to-end recovery, the recovery time is longer (compared to local recovery) and the wasted bandwidth is more, since the notification message must be forwarded by all the intermediate nodes on the path all the way back to the source. In contrast, local recovery may provide sub-optimal alternative routes (optimal from the detecting node to the destination), but is faster and more efficient. In some cases both techniques are used. Local recovery can be used as a first aid to help packets in transit to reach the destination instead of dropping them, until a new end-to-end path is found and used by the source.

III. Survivability Mechanisms

In this section we discuss some of the most well known survivability mechanisms proposed in the literature. Since we are limited in space, this discussion is by no means exhaustive, but the discussed mechanisms are sampled in a way that covers the whole spectrum of the survivability approaches. We discuss each class of survivability mechanisms (as identified in the introduction) in a separate subsection. In addition, we add an extra subsection to cover some of the algorithms used for constructing reliable communication backbones.

A survivability mechanism usually targets a certain type of failures. In general, failures can be either node failures, link failures or service node failures (such as access points, gateways, base stations, or cluster heads). We differentiate between regular network nodes and service nodes, because the failure of a service node has a larger impact on the network, since it affects all the nodes associated with it. It should be noted that the focus of this paper is on the survivability mechanisms that enhance the survivability of communication sessions, and not on the mechanisms that enhance the survivability of individual network components. In other words, we focus on mechanisms that mitigate the effects not the causes of failures.

A. Proactive Protection Mechanisms:

The most agile class of survivability methods is proactive protection, since redundancy (in information and used resources) ensures that the destination will receive the information even if a failure occurs. Because of this fact, most of the previous work in the survivability of multi-hop wireless networks belongs to this category. Approaches to solve the problem of finding multiple disjoint paths can be theoretical (based on graph theory or network flows concepts) or practical in the form of protocols. The authors in [1] take an algorithmic approach, and introduce centralized optimal polynomial-time algorithms for finding either minimum energy link-disjoint or minimum energy node-disjoint paths in wireless Ad Hoc Networks. The proposed algorithms use known minimum-weight k-disjoint paths algorithms (e.g, Bhandari's or Suurballes algorithms) on a transformed graph. The transformation takes any graph G, and transforms it to a fully connected graph G' (i.e., with $\binom{n}{2}$ links), where each link is assigned a cost that represents the needed power to transmit on it by any end node. The algorithms minimize the total energy on all

the used paths by exploiting the wireless multicast advantage (WMA), which also makes them more suitable for wireless networks. Specifically, for the node-disjoint case the problem reduces to optimizing the transmission energy at the source, so that its transmission can reach a suitable set of neighbors that allows establishing k node-disjoint S-T paths. For the link-disjoint case, the problem reduces to finding node-disjoint paths between common nodes on the link disjoint paths using the previous algorithm. The proposed algorithms optimally solves the 2 link-disjoint paths problem in $O(kN^5)$, and the k node-disjoint paths problem in $O(kN^3)$.

On demand routing protocols that are able to find multiple node or link disjoint paths, between a source and destination pair, were developed for ad-hoc networks. Some of these routing protocols are extensions to well known routing protocols such as Ad-hoc On-demand Distance Vector routing (AODV), and Dynamic Source Routing (DSR). There are two main differences between single-path and multi-path routing protocols. First, intermediate nodes are allowed to forward duplicate RREQ (Route Request) messages to give the destination more options to choose from. Duplicate RREQ messages result from broadcasting the RREQ by the original source and all the nodes that hear it afterwards. Second, intermediate nodes are not allowed to reply if they have a valid route to the destination in their routing table, in order to ensure disjointedness between paths.

AODVM (AODV-Multipath), an extension to AODV, was proposed in [8] to find node-disjoint paths. In this approach an intermediate node forwards all the RREQs it receives, and keeps a table (called the "RREQ Table") in which it records all the neighbors from which it received the RREQs. Intermediate nodes are not allowed to send back RREP messages. Therefore, RREPs are sent only from the destination node, where for each RREP message the destination includes a new field that contains the ID of the last-hop to the destination (to distinguish node-disjoint paths). Upon hearing an RREP from a neighbor, an intermediate node deletes the neighbor's entry in its RREQ table (if there is any) and inserts a new route in its routing table. In addition, if a node overhears an RREP message from a neighbor it also deletes the neighbor's entry in its RREQ table to prevent a node from participating in multiple paths (to guarantee node disjointedness). The authors propose using reliable nodes (or R-nodes for short) to increase the number of reliable paths, where it is assumed that R-nodes do not fail at any time. A reliable path is composed of a set of connected reliable segments, where a reliable segment, in turn, is defined either as a set of k disjoint paths between two reliable nodes (k is a design parameter) or a path composed only from reliable nodes. This is clarified in Figure 3. The authors show that randomly placing the R-nodes in the network does not increase the number of reliable path a lot, and thus they propose a placement strategy that relies on the randomized min-cut algorithm. They assume that a node knows the local network topology up to a certain number of hops. From this knowledge each node calculates the min-cut in this local sub-graph using the randomized min-cut algorithm. This information is then spread using HELLO messages that are also used to discover the topology. R-nodes are placed or make their movement decisions according to the received min-cut information, where an R-node moves to the proximity of a node with the least local min-cut. If two or more nodes have the same local min-cut an R-node moves to the proximity of the node with the largest min-cut set (the partition resulting from the cut, that has the largest number of nodes).

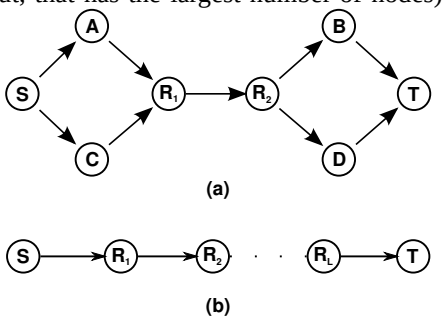

Fig. 3. AODVM:(a) A path composed of three reliable segments, where k=2. (b) A path composed totally from R-nodes

An extension to DSR, referred to as MP-DSR (Multipath-Dynamic Source Routing), was proposed in [9]. MP-DSR is a modification to DSR that enables the computation of multiple node-disjoint paths. In MP-DSR, reliability is treated as a QoS metric, which is used to determine the number of paths to be used. In other words, the reliability of the set of disjoint paths that will be computed by the destination should collectively satisfy a certain reliability requirement. The source starts by determining 1) the lowest acceptable path reliability, $\prod_{low}$, 2) the number of paths, m_0, that the source aims to discover, and 3) the period of time in which the routes will be used, t_w. The values of $\prod_{low}$ and m_0 are carried in the RREQ messages. In addition, the RREQ messages have a field that contains the accumulated reliability of the traversed path. Upon receiving an RREQ an intermediate node updates the accumulated reliability field in the RREQ, and decides to forward the message if the accumulated reliability is larger than $\prod_{low}$. Each intermediate node is allowed to forward m_0 duplicate RREQ messages. Before sending the RREP messages either 1) the destination node waits a certain time period before running a path selection algorithm to choose the disjoint routes (to have enough options), or 2) waits until the received RREQs give enough paths to satisfy the reliability requirement. Route maintenance is needed only when all paths are broken or when t_w expires.

The routing protocols discussed above are proposed to protect unicast connections. However, other connection structures were also considered. For example in [10] to achieve survivable broadcast and multicast, the use of redundant trees was proposed. Basically two broadcast/multicast trees are created, and then, information is forwarded on both trees. The two trees are said to be survivable, if for every destination node each tree has a path from the source to that node, such that the two paths are node-disjoint. Two flavors of this problem were introduced;1) minmax survivable broadcast/multicast trees, in which the maximum used power by any node is minimized and

2) minimum survivable broadcast/multicast trees in which the sum of the transmission power of all the nodes is minimized. An optimal algorithm was presented for the first problem of order $O(n^2 logn)$ and an effective heuristic was given for the second problem of order $O(n^2(m+n))$.

In addition to Ad-hoc networks, many survivable routing protocols have been developed for WSNs. To make use of the dense deployment of WSNs, the authors in [11] presented GRAdient Broadcast (GRAB). Basically, a cost field is first constructed, which assigns each node a cost that represents the needed energy to forward a packet from this node to the sink along the least cost path. Then, when an event occurs, the sensors in the proximity of the event elects a node called the Center of Stimulus (CoS), which has the best reading and will be the only node to send a report to the BS. The CoS assigns a credit ($\alpha + C_{CoS}$) to each report it creates, where C_{CoS} is the cost of the CoS, and α is an additional credit calculated by the CoS. Upon receiving a report, an intermediate node checks the remaining credit in the report, if the ratio of the remaining credit to the original credit is higher than the ratio of the current node cost ($C_{current}$) to the original source cost (C_{source}), i.e., $\frac{\alpha - \alpha_{used}}{\alpha} \geq (\frac{C_{current}}{C_{source}})^2$, the node broadcasts the report with a power high enough to guarantee that the nearest 3 downstream neighbors will receive the report. Otherwise, the node uses its minimum cost path to the sink. This creates a forwarding mesh that is composed of a set of interleaving paths, which will forward the report to the sink, as shown in Figure 4. Obviously, $\alpha + C_{CoS}$ controls the width of the forwarding mesh; a larger α means more robustness. The authors showed through simulation that when $\alpha \geq 6 * C_{CoS}$ the delivery ratio is more than 95%.

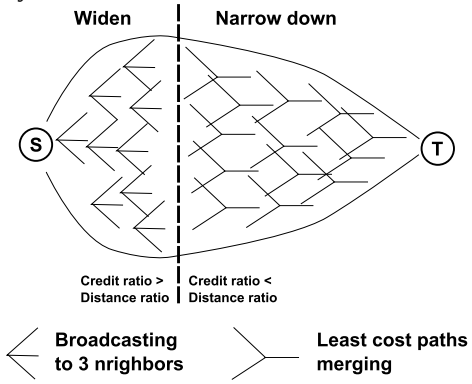

Fig. 4. GRAB: As long as the remaining credit ratio is larger than the remaining distance ratio, the forwarding mesh width increases. When it is less, each node uses its least cost path to the sink, and these paths might start merging as they approach the sink

Angle-based dynamic path construction was introduced in [12] for WSNs. The mechanism is a variation of geographical routing. In geographical routing a node chooses to forward the data packets to the neighbor that makes the best progress towards the sink, which creates a single path to the sink. However, in angle-based routing, a source node (the one that generates a data packet, not the one that forwards) chooses all its neighbors that are located in a certain area to forward its packet. This area is called the angle zone, which in turn depends on an angle θ_{v_i} called the routing angle. Other sensors that forward the packet perform normal geographical routing. The routing angle depends on the distance from a sensor node to the base station, where the longer the distance the narrower the angle. In order to guarantee that power consumption is distributed among the sensor nodes, it is assumed that a sensor node will notify its neighbors when its remaining energy drops below a certain threshold. Upon receiving a notification, an upstream sensor chooses an alternative downstream node.

To tolerate base station (BS) failures in addition to normal sensor node failures, an algorithm to solve the colored tree multiple pair (CTMP) problem was introduced in [13]. Basically, to tolerate BS failures, it was assumed that a WSN may contain more than one base station, and the crux of the algorithm is to find for each node in the network two node-disjoint trees, such that each one of them is rooted at a different BS. Therefore, the WSN can tolerate a single node failure even if it was the BS without loss of information.

B. Reactive Protection Mechanisms

To reduce the amount of traffic produced in a proactively protected communication session, and hence energy consumption, reactive protection can be used. In reactive protection mechanisms, multiple paths are known in advance before the communication session is started. However, they are not used unless a failure was detected on the primary path. Split Multipath Routing (SMR) [14] is an example of such reactive protocols. As in other on-demand source routing protocols, the route discovery is initiated at the source by flooding an RREQ message in the network. When duplicate RREQ messages are received by intermediate nodes only those coming from different links (i.e., neighboring nodes) with the number of hops less than that in the first received RREQ are forwarded, otherwise the message will be discarded (compared to discarding all duplicates as in single path protocols). The reason for this is to give the destination more options to pick maximally disjoint paths. The destination will always choose the path with the least delay (the one included in the first received RREQ) as the primary path. After that, the destination finds the maximally disjoint path(s) with the least delay path (i.e., with the minimum number of shared hops). Then, the path with the least number of hops from those maximally disjoint is selected as an alternative. The authors tested two variations of SMR. In the first, SMR-1, the route discovery process is repeated upon a single failure on any of the paths. In the second, SMR-2, the route discovery process is initialized only when both paths are disconnected. It was shown through simulation that SMR-2 performs better than SMR-1 and outperforms DSR (in terms of packet delivery ratio, delay and overhead).

The many-to-one communication paradigm in wireless sensor networks was considered in [15]. An efficient algorithm was proposed to provide each node in the WSN with a set of node-disjoint paths to choose from in the case of a failure on the primary path. The route discovery process is initiated by the BS that broadcasts a beacon message. Upon hearing the beacon message each of the first hop neighbors of the

BS includes its ID in the beacon (in a newly added field that was left empty by the BS) to distinguish the branches of the tree rooted at the BS. The parent of a node that receives the beacon message is set to be the node from which it received the beacon. A node learns an alternative node-disjoint path to the BS if it receives a beacon from the same route update round but with a different first-hop ID. This way all nodes that can hear beacon messages from multiple branches know multiple node-disjoint paths to the BS. To enhance the chances of other nodes that cannot receive beacons from more than one branch, every node that discovers an alternate path broadcasts this information. Upon hearing an alternate route update message, a node checks if the message was received from a node different from its parent (to guarantee node-disjointedness); if so, the new route is added to its routing table, and its next hop on the alternate path is set to be the node from which it received the route update message. After that the route update message is rebroadcast so that other nodes can benefit from it. Figure 5 shows a simple network, in which the sink has 2 neighbors, i.e., 2 branches are constructed. The solid links represent a branch, the dotted links represent another branch, and the dashed links represent the available links between nodes. In the figure, Black nodes are the nodes that are able to hear beacons from 2 branches, and thus learn an alternate path directly. Grey nodes learn alternate paths from alternate route update messages. If a node wants to forward a data packet and its parent has failed, Per-hop Alternate Path Packet Salvaging (PAPPS) is done, the node randomly chooses an alternate path from its routing table, such that it does not have any node in common with the nodes on the route from the source to the current node, and thus, avoids cycles. Assume that link (A, C) has failed in Figure 5, then node A uses its alternate path through node B.

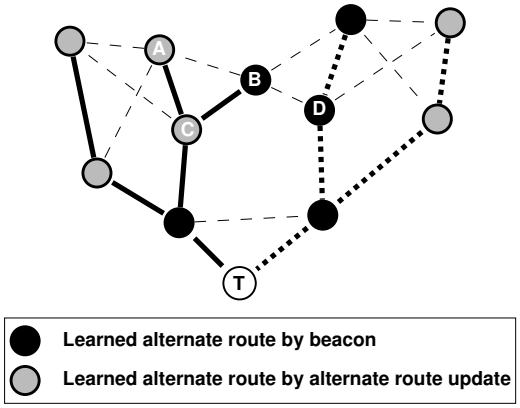

Fig. 5. As long as the remaining credit ratio is larger than the remaining distance ratio, the forwarding mesh width increases. When it is less, each node uses its least cost path to the sink, and these paths might start merging as they approach the sink

Reactive protection can be applied in a different context other than recovering from path failures. Lost association with service nodes, such as APs or cluster heads, can be recovered quickly if a network node knows other service nodes within its range in advance. An example on such a scheme is found in [2]. In this paper a fault-tolerant clustering mechanism for WSN was proposed. In a WSN each sensor node is associated with a *Gateway (Cluster Head)*. To preserve the scarce sensor energy, a sensor associates itself with the cluster head that can be reached using minimum transmission energy. The sensors associated with a gateway are said to be in the final set *FSet* of that gateway. The sensors that can be reached by a certain gateway, say G_i but use less energy to reach another gateway, say G_j, are said to be in the backup set *BSet* of G_i, where the union of Bset and Fset is the range set *RSet*. Upon a gateway node failure or a range failure (the sesnor node cannot reach its initial gateway) the sensor associate itself with the gateway that needs minimum energy to be reached (i.e., the sensor needs to be in at least one BSet). Otherwise, the failure cannot be recovered from.

C. Restoration or Recovery Mechanisms:

Restoration mechanisms are implicitly implemented in all routing algorithms as route maintenance procedures. Ad hoc On-Demand Distance Vector (AODV) Routing uses either local or end-to-end restoration depending on the design parameter "MAX_REPAIR_TTL". This parameter represents the radius (in hops) around the destination in which intermediate nodes are allowed to do local recovery if a failure was detected on the used route. This parameter in turn depends on the network diameter. After local recovery is done and a new path is obtained, the length of the new path is compared to the length of the old path. If the new path has a higher number of hops, a RERR message is sent to the originating source to inform it about this change. Upon receiving the RERR, the source can choose to either keep the new route, or can initiate a new path discovery process. If the new path has similar number of hops as that in the old path, the recovery process will be invisible to the originating source and it will not be notified. The route maintenance procedure in dynamic source routing (DSR) is a combination of restoration and reactive protection and will be discussed in the following section.

As in protection mechanisms, restoration can be done to recover association with service nodes. In [3], the authors presented a scheme to provide survivability against AP failures. They presented *SAWAN* (Survivable Architecture for Wireless LANs). Basically, upon network deployment and before any failure, the AP should identify two kinds of nodes, 1) Bridge nodes, which are nodes that can hear from more than one AP, and 2) Leader nodes, which will act as control heads after the AP fails, and are responsible for calculating new routes to the remaining network. The authors suggested that, the associated nodes to a certain AP should switch to Ad hoc mode upon detecting the failure of that AP, and try to connect to the remaining network with the aid of the leader and bridge nodes.

D. Hybrid Mechanisms:

A mix of protection and restoration techniques can be used in this case. The main advantage of hybrid mechanisms is the design flexibility they provide, which helps in tailoring survivability mechanisms to fit certain application needs. In dynamic source routing (DSR), when a failure is detected by an intermediate node, this node sends an RERR message

to the source. In addition, if the node that detected the failure has an alternate path to the destination in its memory, it uses this path to salvage the packet that triggered the RERR. However, if no routes are available then the packet is discarded. When the source receives the RERR it initiates a new route discovery process to restore its connectivity to the destination. A modification to this operation was proposed in [16]. Upon detecting a failure a node attempts to repair (salvage) the failed route using information in its cache. If no route was found, bypass routing (restoration) is done without sending an RERR to the originating source. A prototype called SLR (Source routing with Local Recovery) is proposed, which is essentially a variation of DSR. This differs from DSR in being a little bit more optimistic, since no RERR message is sent to the source if salvaging fails. Simulation results show that this algorithm reduces the number of broadcasts done for path maintenance, and thus the number of route requests. In addition it was shown that it has a higher delivery ratio and goodput compared to DSR.

This combination of reactive protection, packet salvaging and restoration was also proposed in the CHAMP (Caching And Multi-Path) routing protocol [5]. Basically, the protocol exploits temporal locality to help in salvaging a packet with a failed route (dropped packet), which is done through caching a number of recently forwarded packets at each node. When a failure occurs on the used route the affected node tries to salvage the packet using routing information in its memory. If it fails, it sends back an RERR message to the originating source, which contains the header information of the affected packet(s) that used that failed route. Upon receiving an RERR message, an upstream node checks to see if it has this packet in its cache. If so, it checks to see if it has an alternative route to the destination in its route cache and sends the packet on this route. Otherwise, the RERR message is sent back to the next upstream node until it reaches the source. During route discovery a node that rebroadcasts a RREQ message keeps track of the minimum forwarding count $\min fc$ and the node(s) that sent an RREQ message with $\min fc$ in a set P. This is done per destination. The set P will be used to distinguish the nodes that should forward the RREP back to the source. The same thing is done when dealing with the RREP messages from destination, i.e., the hop count hc to the destination is monitored, thus creating multiple source-destination paths of the same minimum length. Each node on the route keeps track of its next possible hops in a set S, and it alternates between them in a round robbin manner to forward packets to the same destination. This helps in distributing power consumption and the burden of extra storage at the nodes.

A dynamic policy-based multi-layer self-healing mechanism was proposed in [4]. It was suggest that recovery from a failure can be done in different layers according to the survivability needs for the affected application (or applications). The mechanism is multi-layer because it uses different survivability schemes in layers 1, 3 and 4, where it was recommended that SCTP (Stream Control Transport Protocol) should be used instead of TCP in layer 4. The authors suggested choosing from 1:N protection in L1, dynamic on demand re-routing in L3 or the multi-homing ability of SCTP in layer 4, depending on the nature of the running application. For example if the application is delay sensitive, 1:N should be chosen, while if it is delay tolerant the multi-homing ability of SCTP would be more suitable.

E. Survivable Backbones:

Another problem that was studied in the literature is the problem of constructing a reliable network backbone. For example, in [17] the authors presented centralized and distributed algorithms to compute k-vertex connected spanning subgraphs. Simply, a k-vertex connected subgraph is a generalization of the minimum spanning tree, which is 1-connected. A different approach to solve the reliable backbone problem, is by finding a k-connected dominating set. This problem was further extended in [18], where the authors presented two algorithms to construct a k-connected m-dominating set $kmCDS$ in a graph G(V,E) to act as a communication backbone for a WSN. A set $D \subset V$ is an m-dominating set if any node in $V \backslash D$ is a neighbor to at least m-1 nodes in D (a node dominates itself). The centralized algorithm CGA constructs a $kmCDS$ in $O(|V|^{3.5}|E|)$ by adding nodes to the set C (which will be the kmCDS when the algorithm ends) in a non-increasing order of their number of neighbors; breaking ties by the remaining power, and finally breaking ties arbitrarily by the node ID. The finishing step is to optimize C by removing nodes from it such that it remains k-connected and m-dominating. The authors propose a Distributed Deterministic Algorithm DDA to do the same job by first using one of the known distributed algorithms to compute a CDS, then using another known distributed algorithm to compute m-1 MISs (Maximum Independent Sets) in $G \backslash C$, and finally adding nodes to C relying on the fact that if a node has k neighbors in C then it can be added to it and the new C will still be k-connected. The difference between these algorithms and previous ones in the literature is that they allow the case of $k \neq m$.

F. Coding-based protection

In Section II it was indicated that erasure codes and network coding can be used to enhance the scalability of proactive protection schemes. The work in [6] gives an example of using erasure codes in a fashion similar to that discussed in Section II to reduce the overhead of data redundancy. That is, a data packet is first divided into n smaller sub-packets, from which m redundant sub-packets are computed. Then, these n+m sub-packets are sent on n+m node disjoint paths, which will result in less overhead compared to duplication, especially if $m << n$. This enables us to tolerate m failures, since the destination needs only n sub-packets to recover the original data. In [6], the value of n is made equal to the expected number of paths that will be successful with high probability, which can be estimated given a set of paths to the sink and their failure probabilities.

As for network coding, the authors in [7] proposed using network coding to provide proactive protection against link

failures in many-to-one flows. The problem considers a set of wireless mesh routers and the set of wireless mesh clients (or users) associated with them. It is assumed that the user's data units will be relayed to a common gateway (sink) through multihop wireless communication. The set of users contains n users, and the set of routers to which the users are directly connected contains at least $n+1$ routers. To tolerate a single failure, the routers have to create $n+1$ ($n+e$ if e failures are to be tolerated) linear combinations from the user's data units, such that any n of them are linearly independent. These $n+1$ combinations are then forwarded to the sink on $n+1$ (or $n+e$) edge-disjoint paths, which means that a single (e) link failure(s) will at most affect one (e) path(s), and thus the sink will be able to recover the original data units if at most one (e) failure(s) take place, by using the remaining n linear combinations. If the number of available paths from the routers to the sink is less than $n+1$, the n sources cannot transmit together and must be divided into groups. It was shown in [7] that this general problem is NP-complete. For the single failure case, an efficient algorithm was proposed to create a coding tree, which produces the required combinations at the router nodes in $O(n^2)$.

IV. Duplication Vs Coding-based mechanisms

In this section we compare duplication to coding based approaches, i.e., both erasure codes and network coding. We assume a simple topology, where we have a source/destination pair S and T, with k node-disjoint S-T paths in between, where k is an even number, and all the paths have the same number of hops, L. In addition, we assume that protection is to be provided for the S-T information flow against a single failure, and that the one hop propagation delay is τ. We take the transmission conflicts between the nodes along the same path into account, but we ignore the conflicts between nodes on different paths for simplicity. We assume that the interference range equals the transmission range, i.e., a node can only transmit to and interfere with all its 1-hop neighbors on the same path.

To tolerate a single failure in duplication-based approaches, each packet is forwarded on 2 disjoint paths. Therefore, if there are k paths the source can use a different pair of paths for each packet to distribute power consumption. Since a node cannot transmit and receive at the same time, the source can transmit a packet every 3τ because it needs to wait for the 2-hop neighbors to transmit first so that their transmissions do not conflict with its transmission to the 1-hop neighbors. Therefore, the rate of receiving useful information at the destination is $1/3\tau$ (i.e., one data unit every 3τ). Note that the throughput is independent of the number of paths.

In erasure codes, each sub-packet is transmitted alone and needs roughly τ/n time units (since its size is smaller). We assume an optimal erasure code in which only a single redundant sub-packet is generated, i.e., a total of $n+1$ sub-packets are transmitted to provide protection against a single failure. In this case, the source can transmit the $n+1$ sub-packets every $(n+1+2)\tau/n = (n+3)\tau/n$ time units.

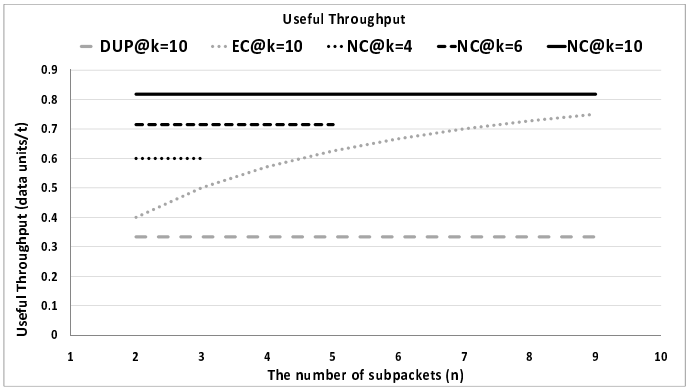

Fig. 6. Useful throughput of the three schemes in (data units/τ). The useful duplication rate is constant. The erasure codes rate is dependent on n. Finally, the network coding case is dependent on the number of paths k, and achieves better rate compared to duplication and erasure codes

Therefore, the rate in this case is $n/\tau(n+3)$. Obviously, the rate is a function of n. However, it also depends on the number of paths indirectly, since $n+1$ cannot be larger than k.

When network coding is used, the k paths carry k combinations in k-1 data units to the destination, such that any k-1 of them are solvable. This can be easily accomplished by sending k-1 native (uncoded) data units to k-1 first-hop neighbors of the source. Because of the wireless multicast advantage, the last 1-hop neighbor will be able to overhear these k-1 transmissions and XOR all the received data units to create the last combination. Since only k-1 packets are transmitted (not sub-packets), only $(k-1+2)\tau$ time units are needed. Therefore, the useful data rate at the destination in this case is $(k-1)/\tau(k+1)$. Note that the rate depends clearly on the number of paths. Figure 6 plots the rate for the three cases, where the x axis is the number of sub-packets n. Note that the performance of duplication and network coding is independent of n. The dashed Gray line is the rate for duplication, which is a constant and independent of n. The Gray dotted line represents the rate for erasure codes, which clearly gets better as the number of sub-packets increases. The erasure codes rate is drawn for the case when k = 10. However, for smaller values of k the rate follows the same trend but the function will be undefined after n=k-1, since the total number of sub-packets (n+1) cannot exceed the number of paths (k). The set of Black lines (dotted, dashed and solid), represent the rate for network coding, where each line represents the rate for a certain k. As in erasure codes, the lines representing the network coding performance for some k cannot extend beyond n=k-1. Note that when an erasure code is used, in each transmission round the source makes k short transmissions, one for each sub-packet. That is, to transmit k-1 packets the source needs $(k \times \tau/n)(k-1)$ time units, and since $n = k-1$ the source will eventually need $k\tau$ time units. However, when network coding is used only $(k-1)\tau$ time units are needed, which establishes the difference in performance between erasure codes and network coding.

V. Summary and research directions

In this survey we covered the different classes of survivability mechanisms for multihop wireless networks, namely,

TABLE I
SURVIVABILITY MECHANISMS. IN THE TABLE, PP=PROACTIVE PROTECTION, RP=REACTIVE PROTECTION, RT=RESTORATION, HY=HYBRID, U=UNICAST, M=MULTICAST, B=BROADCAST, C=CONVERGECAST, A=ASSOCIATION, BK=BACKBONE, L=LINK FAILURES, N=NODE FAILURES AND S=SERVICE NODE FAILURES

Scheme	Network Type	Class	Connection Type	Failure Type
Minimum-Energy Disjoint paths [1]	Ad-hoc	PP	U	N/L
AODVM [8]	Ad-hoc	PP	U	N/L
MP-DSR [9]	Ad-hoc	PP	U	N/L
GRAB [11]	WSN	PP	U	N/L
Angle-based [12]	WSN	PP	U	N/L
Redundant Trees [10]	Ad-hoc	PP	M/B	N/L
FLSS [17]	Any	PP	Bk	N/L
KMDS [18]	WSN	PP	Bk	N/L
CTMP [13]	WSN	PP	C	N/L/S
EC [6]	WSN	PP	U	N/L
NC [7]	WMN	PP	C	L
SMR [14]	Ad-hoc	RP	U	N/L
PAPPS [15]	WSN	RP	C	N/L
Fault-tolerant clustering [2]	WSN	RP	A	S
SWAN [3]	WLANs	Rt	A	S
PBMLSH [4]	Any	Hy	U	N/L
SLR [16]	Ad-hoc	Hy	U	N/L
CHAMP [5]	Ad-hoc	Hy	U	N/L

proactive and reactive protection, restoration, hybrid and coding-based mechanisms. Moreover, we provided a comprehensive discussion on the related issues and challenges in the different survivability mechanisms. Finally, selected examples from the literature were surveyed, which covers most of the survivability mechanisms spectrum. The summary of all these protocols and algorithms is presented in Table I.

Although many survivability mechanisms exist in the literature, there has been little work done to evaluate and compare their performance against each other on common grounds either quantitatively or by simulation. Such a comprehensive comparison is important to practically assess the performance of different mechanisms. In addition, most of the work in survivability focuses on isolated independent failures. Although studying such failures is important, other types of failures should also receive attention in future work.

An interesting problem is to compute routes that not only satisfies certain QoS metrics, but also that guarantees a certain level of survivability. To the best of our knowledge these two problems have been studied separately, and until now there is no survivability mechanism that jointly addresses them together. Another problem that needs further investigation, is the problem of constructing survivable communication backbones for networks with mobile nodes. This problem is NP-hard in static networks, which makes it even harder when mobility comes into play.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under grants CNS-0626822, CNS-0626741, CNS-0721453 and ECS-0601570 and by a gift from Cisco Systems.

REFERENCES

[1] A. Srinivas and E. Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. In the proceedings of the 9th annual international conference on Mobile computing and networking (Mobicom 2003). Pages: 122 - 133.

[2] G. Gupta and M. Younis. Fault-tolerant clustering of wireless sensor networks. In proceedings of WCNC 2003.

[3] M. Virendra, S. Upadhyaya, V. Kumar, and V. Anand. Sawan: A survivable architecture for wireless lans. In proceedings of the Third IEEE International Workshop on Information Assurance,IWIA 2005.

[4] L. Kant and W. Chen. Service survivability in wireless networks via multi-layer self-healing. In the proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2005). Page(s):2446 - 2452 Vol. 4.

[5] A. Valera, W.K.G. Seah, and S. Rao. Cooperative packet caching and shortest multipath routing in mobile ad hoc networks. In proceedings of INFOCOM 2003.

[6] S. Dulman, T. Nieberg, J. Wu, and P. Havinga. Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. In Proc. WCNC 2003.

[7] O. Al-Kofahi and A. Kamal. Network coding-based protection of many-to-one wireless flows. IEEE Journal on Selected Areas in Communications, VOL. 27, NO. 5, JUNE 2009.

[8] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi. A framework for reliable routing in mobile ad hoc networks. In proceedings of Infocom 2003.

[9] R. Leung, J. Liu, E. Poon, A. C. Chan, and B. Li. Mp-dsr: A qos-aware multi-path dynamic source routing protocol for wireless ad-hoc networks. In proceedings of LCN 2001.

[10] J. Tang, G. Xue, and W. Zhang. Energy efficient survivable broadcasting and multicasting in wireless ad hoc networks. In the proceedings of Military Communications Conference, 2004 (MILCOM 2004).Pages:1165 - 1171 Vol. 3.

[11] F. Ye, G. Zhong, S. Lu, and L. Zhang. Gradient broadcast: A robust data delivery protocol for large scale sensor networks. Wireless Networks 11, 285298, 2005.

[12] W. Choi, S.K. Das, and K. Basu. Angle-based dynamic path construction for route load balancing in wireless sensor networks. In proceedings of WCNC 2004.

[13] P. Thulasiraman, S. Ramasubramanian, and M. Krunz. Disjoint multi-path routing to two distinct drains in a multi-drain sensor network. In proceedings of Infocom 2007.

[14] S. Lee and M.Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In proceedings of ICC 2001.

[15] W. Lou. An efficient n-to-1 multipath routing protocol in wireless sensor networks. In Proc. 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2005).

[16] C. Sengul and R. Kravets. Bypass routing: An on-demand local recovery protocol for ad hoc networks. Elsevier Ad Hoc Networks Journal, Volume 4, Number 3, pp. 380-397, 2006.

[17] N. Li and J. C. Hou. Flss: A fault-tolerant topology control algorithm for wireless networks. In the proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom 2004). Pages: 275 - 286.

[18] Y. Wu, F. Wang, M. T. Thai, and Y. Li. Constructing k-connected m-dominating sets in wireless sensor networks. In proceedings of MILCOM 2007.