A Bypass-Ring Scheme for a Fault Tolerant Multicast

V. Dynda

We present a fault tolerant scheme for recovery from single or multiple node failures in multi-directional multicast trees. The scheme is based on cyclic structures providing alternative paths to eliminate faulty nodes and reroute the traffic. Our scheme is independent of message source and direction in the tree, provides a basis for on-the-fly repair and can be used as a platform for various strategies for reconnecting tree partitions. It only requires an underlying infrastructure to provide a reliable routing service. Although it is described in the context of a message multicast, the scheme can be used universally in all systems using tree-based overlay networks for communication among components.

Keywords: distributed systems, fault tolerance, message multicast, tree-based networks, tree recovery, repair algorithm.

1 Introduction

Recently, many distributed applications, particularly peer-to-peer internet-based systems, have become increasingly popular. As these systems need to transport data to a set of participating nodes, an efficient and reliable multicast is critical for their success.

Since multicast message routing is usually realized by overlay structures connecting members of often sparsely distributed multicast groups, it is necessary to deal with failures not only at the level of routers of the underlying infrastructure but also at the overlay level to keep the routing structure in an operational state.

The most efficient topology for message propagation in a multicast group is a tree, which is scalable and can be easily reconfigured according to the current network state to keep message dissemination cost minimal. However, tree topologies may have a reliability problem, since without any enhancement a single point of failure causes tree partitioning and thus prevents the messages being delivered to all receivers.

Generally, there are two approaches for tree recovery. The optimistic approach does not care about partitioning before it occurs, and only when partitioning is detected does the system attempt to rejoin the partitions into a connected graph. This on-demand fault tolerance usually forces affected members to abandon the existing connections and rejoin the tree ([1], [8]). However, the long recovery latency could be undesirable for many applications.

The alternative is the pessimistic approach where backup routes are set beforehand, and they are activated when a failure is detected [5], [6].

We present a scheme for failure repair in tree-topology structures based on virtual cyclic backup paths used to bypass faulty nodes and repair the tree without traffic interruption. Since our protocol ensures that all partitions are reconnected and that no cycle is formed during recovery, no matter how and where faults are detected, it can be used as a platform for various strategies of partition reconnection that influence the resulting tree topology. Thus, the tree optimization preferences of other application levels can be involved in the repair process. The scheme can be used generally in applications using overlay tree-topology networks for message propagation.

In this paper, we aim at message multicast, since multicasting is one of applications where trees can be easily and efficiently used, and fault tolerance is the critical issue at the same time. Other applications may include unicasting, routing, object location, replica management, etc.

The rest of paper is structured as follows. In section 2 the related work is summarized. Section 3 presents the model and the notations used further in the text. In section 4 we introduce the fault tolerant scheme, describe the principles of the scheme, single and multiple failure repair mechanisms and deal with practical issues. Section 5 briefly describes three different repair methods for tree partition reconnection. Section 6 discusses the properties of our protocol and section 7 contains conclusion and sets some future directions.

2 Related work

Several fault tolerant schemes for multicast trees have been reported based on preplanned failure restoration. Some of them ([5], [12], [11]) use schemes similar to path restoration or link restoration, originally proposed for unicast fault tolerance in self-healing ATM networks ([7], [9]) where pre-computed backup virtual paths either protect an entire individual end-to-end virtual path or are used to reroute the traffic originally carried by a failed link.

In the Dual-Tree Scheme [5], a secondary tree providing alternative delivery paths is built and it is activated when node or link failure is detected in the primary tree. Unfortunately, this scheme as well as schemes proposed in]7], [12], [11] assume a single failure model in which there is only a single link or node that fails at the time.

An approach dealing with multiple failures is the Efficient Fault Tolerant Multicast Routing Protocol [6], where bypass paths connect nodes and their grandparents in the tree and are used to reconnect partitions when the parent node of these nodes fails. This solution is not efficient in multi-source networks, where the parent-child relation often alters.

A different approach to ensure multicast fault tolerance is used in Bayeux architecture [14]. The messages are routed in a tree determined by a common prefix of addresses with small salt values, enabling delivery even in the case that a particular intermediate address is not available. Bayeux architecture requires Tapestry [13] as its underlying infrastructure.

3 Model and notations

The underlying network that we consider in this paper is a network providing a routing service (e.g., IP network, Tapestry [13], Pastry [10], or just a set of virtual inter-process connections with routing capability) modeled as a graph $SN = (V, E)$ where V is a finite set of vertices representing nodes; E is a finite set of edges, representing links between nodes in the network.

A multicast group MG, i.e. a set of nodes receiving multicast messages, is an arbitrary subset of nodes from V; $MG = \{n_i; i = 1, \ldots, k; n_i \in V\}$ where n_i are nodes from SN that are to exchange information and $k = |MG|$ is multicast group size.

In order to connect members of a given multicast group and allow for efficient message traffic, an overlay multicast tree connecting all MG-members is built and it is used as a source-independent structure for message propagation to the group members. The multicast tree is modeled as a graph $MT = (MG, CE)$ where CE is a set of *core tree edges* – virtual links (built on top of SN) connecting nodes in MG. We expect that the multicast group (thus either multicast tree) dynamically adapts to the current network state and user requirements.

We assume that nodes in MT may fail and that their faulty state can be detected by neighboring nodes. Note that we do not have to deal with link failures, since the multicast tree is an overlay structure. Thus, message delivery across a virtual link connecting two MT-neighbors n_1 and n_2 depends on routing in the underlying network fabric, and we expect it to deliver the message with the best effort if there is any path from node n_1 to node n_2 in SN.

Further in the text, we consider each node $n \in V$ to be assigned with an SN-unique identifier ID_n.

4 Fault tolerant scheme

Message dissemination using a multicast tree is vulnerable to even a single node failure. To prevent network partitioning, we propose a protocol based on deploying *bypass rings* of radius r that are used to bypass either a single faulty node or a cluster of faulty nodes, to reconnect the tree avoiding cycles and thus to enable the communication between remaining group members to continue.

4.1 Basic definitions

Besides SN-unique ID, each member of multicast tree MT may be assigned with an MT-specific *hierarchical identifier* HID. Identifier HID_n^c of node n related to node c in MT tree is a concatenation of IDs of nodes on the only path in MT from node c to n. Functions $pref\left(i, HID_n^c\right)$ and $suff\left(i, HID_n^c\right)$ denote a prefix and suffix of length i of identifier HID_n^c and $gcp\left(HID_{n_1}^c, HID_{n_2}^c\right)$ denotes *the greatest common prefix* of the hierarchical identifiers of nodes n_1 and n_2. For example, in Fig. 1, HID_{2F}^{BA} is BA.18.2F, $pref\left(1, HID_{2F}^{BA}\right)$ is BA and $gcp\left(HID_{2F}^{BA}, HID_{F6}^{BA}\right)$ is BA.18.

Let $MT = (MG, CE)$ be the tree-topology communication network. Bypass ring $BR_c(1)$ of radius 1 centered at node $c \in MG$ is a circuit consisting of an ordered sequence of nodes $n_1, n_2, \ldots, n_{t+1} \in MG$, $n_1 = n_{t+1}$ such that:

(1.1) For all n_i, $i = 1, \ldots, t$, distance $d(n_i, c) = 1$ and

(1.2) $HID_{n_i}^c < HID_{n_{i+1}}^c$ for $i = 1, \ldots, t-1$.

A directed *bypass edge* $be_i = (n_i, n_{i+1})$ (for all $i = 1, \ldots, t$) is a virtual link from node n_i to node n_{i+1}, where n_i is the initial node and n_{i+1} the terminal node of be_i. For convenience, the direction is further indicated by left (L) / right (R) symbols, so that the *left ring-neighbor* of node n_i is the initial node of the bypass edge terminated at n_i and the *right ring-neighbor* is the terminal node of the bypass edge initiated at n_i.

The bypass ring $BR_c(r)$ of radius $r > 1$ centered at node $c \in MG$ is a circuit consisting of an ordered sequence of nodes $n_1, n_2, \ldots, n_{t+1} \in MG$, $n_1 = n_{t+1}$ such that:

(2.1) For all n_i, $i = 1, \ldots, t$, distance $d(n_i, c) = r$ or $d(n_i, c) \le r$ if n_i is a leaf of MT and

(2.2) Each sequence of all nodes n_i, n_{i+1}, ..., n_{i+j} having $gcp\left(HID_{n_i}^c, \ldots, HID_{n_{i+j}}^c\right)$ equal to $pref\left(r, HID_{n_i}^c\right)$ is ordered equally to the ordering of these nodes in $BR_{pref(r,\, HID_{ni}^c)}(1)$.

Property ensures that all bypass edges constructed between any two nodes are equally oriented, which is important for the repair process.

A *complete bypass ring* is a ring consisting of *all* nodes having property (1.1) or (2.1); a *reduced bypass ring* comprises only a subset of these nodes. An example of complete $BR(1)$ and $BR(2)$ centered at node c is shown in Fig. 1.

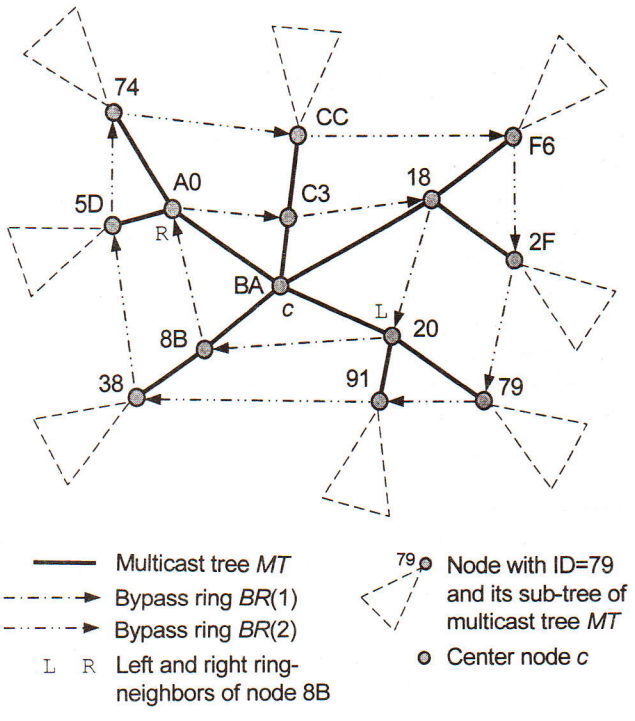

Fig. 1: Example of complete bypass rings $BR(1)$ and $BR(2)$

Integration of all bypass rings and an MT graph creates an *extended multicast tree* $EMT = (MG, CE \cup BE)$ where CE is the set of core tree edges of the original MT and BE is the set of all bypass edges. MT is then a spanning tree of EMT.

Each node $n \in MG$ holds a *BR routing table* containing information about all bypass rings that the node is member of. Each entry in the table consists of the ID of ring center node c, the IDs of the left and right ring-neighbors, $suff\left(2, HID_n^c\right)$, $pref\left(2, HID_n^c\right)$ and the radius of the ring.

Further in this paper, we will consider only complete bypass rings and assume that if $BR_c(r)$ is part of EMT then all $BR_c(q), 1 \le q \le r$ are contained in EMT, too. r_{max} denotes the maximum radius of bypass rings deployed.

4.2 Practical bypass ring construction algorithm

A node c that is to create its bypass ring of radius 1 first sorts all its t MT-neighbors n_i by their IDs to get the sorted list $n_1, \ldots, n_t$ fulfilling property (1.2) and sends to each node n_i a CREATE_BR$\left(pref\left(2, HID_{n_i}^c\right), ID_{n_l}, ID_{n_r}\right)$ message, where $l = (i-2) \bmod t + 1$ and $r = i \bmod t + 1$. Upon receiving a CREATE_BR message, each node n_i saves the ID of the sender together with its own ID as $suff\left(2, HID_{n_i}^c\right)$, $pref\left(1, pref\left(2, HID_{n_i}^c\right)\right)$ as the ID of center node c and ID_{n_l} and ID_{n_r} as left and right ring-neighbors into its BR table.

To build $BR(r+1)$, each member n of $BR(r)$ sorts all its t MT-neighbors by their ID to get the list $n_1, \ldots, n_j, \ldots, n_t$, where $ID_{n_j} = pref\left(1, suff\left(2, HID_n^c\right)\right)$ and creates the sequence n_{j+1}, $\ldots, n_t, n_1, \ldots, n_{j-1}$ to fulfill property (2.2). This sequence is then dealt as sorted list in the case of $BR(1)$ construction. To get $ID_{n_l}^c$ for n_{j+1} and $ID_{n_r}^c$ for n_{j-1}, node n has to communicate with its left and right $BR(r)$-neighbor, respectively.

Construction of rings (namely $BR(1)$) is similar to construction of a sorted circular bidirectional linked list. The important factor here is to prevent incomplete rings in cases when the center node fails during the process. Each node sends a HALLO_BR message to its ring-neighbors upon receiving CREATE_BR from the center node. If it does not receive the same message from both its ring-neighbors within a certain period of time, it assumes that the ring has not been constructed completely and sends a DELETE_BR message around to delete accordant entries in the BR tables of the members of the incomplete ring.

Updating the bypass ring when a member is either added or deleted is again similar to corresponding operations with a sorted circular bidirectional linked list. The center node sends CHANGE_BRR and CHANGE_BRL messages, causing the target node to change the ID of its right and left ring-neighbor, retaining the orientation in the ring. To deal with center node failure that can lead to bypass ring damage, nodes receiving a CHANGE_BRL or CHANGE_BRR message have to agree with each other to perform the change atomically.

4.3 Single failure repair using BR(1)

The aim of the repair process is to create new core tree edges using a bypass ring to connect tree partitions caused by a node failure and restore the multicast tree MT to the connected and consistent state. At the same time, it has to be assured that the repair process operates properly no matter where and at how many nodes the repair of a failed node is simultaneously initiated.

Let node c be the faulty node in $EMT = (MG, CE \cup BE)$ and $BR_c(1)$ its bypass ring consisting of nodes $n_1, \ldots, n_t$. Define function $R(n_i)$ for all n_i, $i = 1, \ldots, t$ as follows:

(3.1) $R(n_i) = HID_{n_f}^c$, if node n_i has been first notified about node c failure by node n_f. If n_i detects the failure itself then $f = i$.

(3.2) $R(n_i)$ is undefined if node n_i does not know about node c failure yet.

Define relation $\rightarrow$: $n_i \rightarrow n_j$ if and only if:

(4.1) (n_i, n_j) is the bypass edge of $BR_c(1)$ and

(4.2) $R(n_i)$ is defined and

(4.3) $R(n_i) < R(n_j)$ or $R(n_j)$ is not defined.

The basic idea behind the repair process is that each neighbor $n_f = n_i$ of faulty node c detecting the failure consecutively iterates along the ring $BR_c(1)$ in the direction of the bypass edges through the nodes $n_{i_1}, n_{i_2}, \ldots \in BR_c(1)$ (n_{i_1} is the right ring-neighbor of $n_i = n_{i_0}$, n_{i_2} is the right ring-neighbor of n_{i_1}, ...) until it reaches a node n_{i_p} that has already been notified about failure (i.e., $R(n_{i_p})$ is defined). At each hop n_{i_q}, $1 \le q \le p$, it is determined if $n_{i_{q-1}} \rightarrow n_{i_q}$ and n_{i_q} is notified about failure of node c such that $R(n_{i_q}) = n_i$. Termination of this process is guaranteed, since $BR(r)$ consists of a finite number of nodes (each node has a finite set of neighbors in MT).

After node n_{i_q} has been notified and if $n_{i_{q-1}} \rightarrow n_{i_q}$ then a new core tree edge $ce_{i_q} = (n_{i_q}, n_{i_r})$ is constructed (and EMT modified appropriately) with the following properties:

(5.1) $r < q$ and

(5.2) $n_{i_{q-1}} \rightarrow n_{i_q}$ and

(5.3) There is a path $n_{i_q}, n_{i_r}, \ldots, n_i$ in the repaired MT.

Several methods can be deployed for selection of node n_{i_r}. They are discussed in section 5.

After all nodes $n_1, \ldots, n_t \in BR_c(1)$ have been notified, the relation $\rightarrow$ between the incident nodes of all bypass edges be_i is known. With the definition of the bypass ring and $\rightarrow$ relation, it can be proven that relation $\rightarrow$ on the bypass ring has the following properties:

(6.1) Relation $\rightarrow$ is not cyclic. That is, there is no subset of nodes (say, $n_1, \ldots, n_j \in BR_c(1)$) such that $n_1 \rightarrow n_1 \rightarrow n_1 \rightarrow \ldots \rightarrow n_j \rightarrow n_1$.

(6.2) There is only one bypass edge $be_i = (n_u, n_v) \in BR_c(1)$ such that $n_u \rightarrow n_v$ is not true.

Together with properties (5.1)–(5.3) of newly constructed core tree edges, we can get the following:

(7.1) The repaired MT connects all partitions induced by a node failure.

(7.2) The repaired MT is a tree graph.

Moreover, the described technique is independent of the repair initiating node and also prevents collisions in cases when multiple nodes initiate repair simultaneously (which can easily happen in a distributed system).

4.4 Practical single failure repair algorithm

A practical repair algorithm may incorporate several performance improvements:

- The iteration along the ring from the failure detecting node n_i is performed in both directions (further referred to as a *two-way algorithm*).
- The iteration is not performed directly by n_i, it is rather delegated by the ring members.
- The new core tree edges are constructed on the fly, immediately after node n_{i_q} is notified and relation $n_{i_{q-1}} \rightarrow n_{i_q}$ determined.

The algorithm works as follows:

Each node n_i that during message routing through the multicast tree realizes that one of its neighbors, say, node c, is down, excludes node c from its bypass ring $BR_{n_i}(1)$ and sends REPAIR(ID_c, ID_{n_i}) messages to the both its ring-neighbors on the ring $BR_c(1)$.

```
procedure OnRecv_REPAIR ( ID of failed node IDc,
                          ID of initiating node IDni)
(performed at niq, REPAIR received from ring-neighbor niq-1)
1.  remove node c from BR(1) centered at niq
2.  if R (IDniq) is undefined then
3.    define R (IDniq) = IDni
4.    select node nir to create core tree edge ceiq = (niq, nir)
5.    send CREATE_CE to node nir // Create a new core edge
6.    add node nir to BR(1) centered at niq
7.    add node niq to BR(1) centered at nir
8.    send REPAIR (IDni, IDc) to the ring-neighbor niq+1
    else  // R (IDniq) is defined
9.    if R (IDniq) == IDni then
10.     send REJECT to node niq-1 and goto step 18 end
11.   if R (IDniq) < IDni then
12.     if niq-1 is right ring-neighbor of niq then
13.       perform steps 4 – 7 // Create a new core edge
14.     else send REJECT to node niq-1 end
        else
15.     if niq-1 is left ring-neighbor of niq then
16.       perform steps 4 – 7 // Create a new core edge
17.     else send REJECT to node niq-1 end
        end
    end
18. delete entry concerning BRc(1) in BR table
end
```

Alg. 1: Single failure repair process

Upon receiving a REPAIR message from its ring-neighbor, each node n_{i_q} runs the OnRecv_REPAIR() procedure (see Alg. 1). This algorithm determines relation $n_{i_{q-1}} \rightarrow n_{i_q}$ (steps 2, 11) and forwards a REPAIR message along the ring (step 8). If there is relation $n_{i_{q-1}} \rightarrow n_{i_q}$, node n_{i_q} selects n_{i_r} according to the chosen repair method and creates a new core tree edge $ce_{i_q} = (n_{i_q}, n_{i_r})$ (steps 4, 5). When ce_{i_q} is created, the relevant bypass rings have to be updated (steps 6, 7).

An example of the distributed construction of relation $\rightarrow$ on BR(1) is shown in Fig. 2. In the first step, node 18 finds that node c is not available so it uses $BR_c(1)$ to eliminate tree partition and sends REPAIR(c, 18) messages to its ring-neighbors (step 2) to determine the $\rightarrow$ relation. In step 4, another node (8B) similarly sends a REPAIR(c, 8B) message to its ring-neighbors. Since the first REPAIR message received by node A0 originated at 18 (after step 3 R(A0) = 18) and the second one (from node 8B) was received from its left neighbor, node A0 rejects the message (there is no relation 8B $\rightarrow$ A0) and thus prevents a cycle forming. Each node accepting a REPAIR message establishes a new core tree edge according to the chosen repair method.

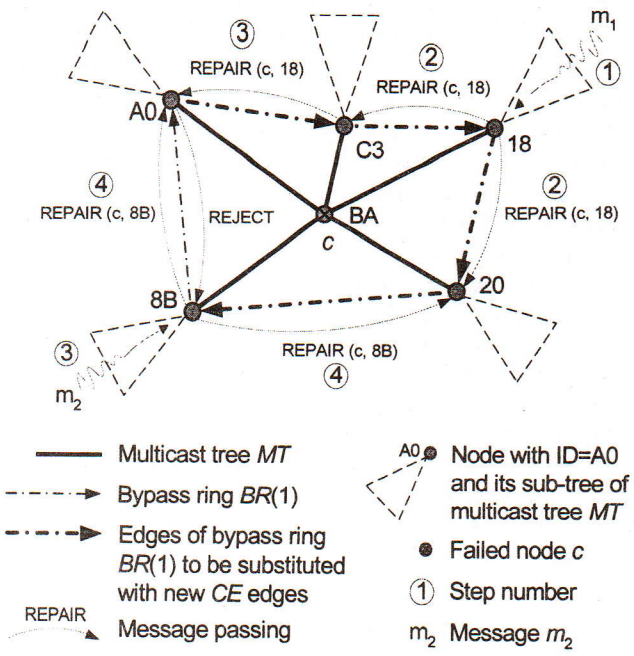

Fig. 2: Example of a two-way single failure repair process

4.5 Multiple failure repair

To deal with failure of a cluster of nodes in an *MT* tree, bypass rings $BR(r)$, $r > 1$, have to be deployed. Essentially, the repair process is similar to the single failure repair described in section 4.3, as it is also based on relation $\rightarrow$ determination between each two neighbors on some repair route. The difference is, of course, in the repair message routing and in relation $\rightarrow$ determination, since the nodes on the route may not be sorted in general.

Let $FC = \{c_i;\ i = 1, 2, \ldots, s;\ c_i \in MG\}$ be the faulty cluster (i.e., set of faulty nodes), where $s = |FC|$ is the cluster size. The repair message is routed as follows. The repair process always begins at the member of $BR(1)$, since the failure is detected by the *MT*-neighbor of some faulty node. Each node n_i (detecting failure of its neighbor c_i) iterates through nodes along $BR_{c_i}(1)$ in the direction of the bypass edges in the same way as in single failure repair, except that it can reach another faulty node $n_{i_{q+1}} = c_{i+1} \in BR_{c_i}(1)$. In this case, $ID_{c_{i+1}}$ is appended to the list of faulty nodes and the iteration continues from node n_{i_q} along the ring $BR_{c_{i+1}}(r)$, where $r = d(n_{i_q}, c_{i+1})$, provided that a bypass ring with radius r centered at c_{i+1} is constructed ($r \le r_{max}$). This 'switch' to another ring is done whenever the next regular node in the repair route is faulty. From each member of a $BR_c(r)$, $r > 1$, the repair message is routed along a core tree edge to the *MT*-neighbor node $m \in BR_c(r-1)$ (if it is not faulty) in order to keep the iteration path as close as possible to the failed cluster of nodes.

This process is performed until iteration reaches a node $n_{i_p} = n_{i+1}$ that has already been notified about failure of at least one node c_i from the list of faulty nodes (i.e., $R(n_{i_p})$ is defined). The number of steps of this iteration is finite, since all $BR(r)$ have finite length, the number of faulty nodes also has to be finite and all rings have a uniform orientation.

Let P_i be iteration paths initiated at nodes n_i, $i = 1, \ldots, h$, where h is the number of nodes initiating multiple failure repair. Since every path P_i is terminated at node n_{i+1} initiating path P_{i+1}, the union $P_1 \cup P_2 \cup \ldots \cup P_h$ forms a continuous path surrounding the faulty cluster, where the terminating node of P_h is at the same time the initiating node of P_1. This cycle is called a *bypass cycle BC*. A bypass cycle has similar properties to a bypass ring, and in the case of single failure repair of node c, the bypass cycle is formed solely by $BR_c(1)$.

To enable the multiple failure repair algorithm to work properly with bypass cycles, paragraph (4.1) in the definition of relation $n_i \rightarrow n_j$ introduced in section (4.3) has to be modified:

(8.1) (a) $n(n_i, n_j)$ is a bypass edge of $BR_c(r)$ or

(b) $(n_i, n_j) \in MT$ and $n_i \in BR_c(r)$ and $n_j \in BR_c(r+1)$ or

(c) $(n_i, n_j) \in MT$ and $n_j \in BR_c(r)$ and $n_i \in BR_c(r+1)$

To achieve comparability between nodes on BC (paragraph (4.3)), paragraph (3.1) of the function R definition also has to be modified:

(9.1) $R(n_i) = HID_{n_f}^{c_m}$, where $ID_{c_m} = \min(\{ID_{c_i};\ c_i \in FC\})$, if node n_i has been first notified about node c failure by node n_f.

Node c_m, $HID_{n_f}^{c_m}$ and also $HID_{n_i}^{c_m}$ are determined during repair message routing. To get the correct node c_m considering all nodes $c_i \in FC$ (not only the information known locally to the message routed along a particular path P_i), a technique similar to the Chang-Roberts leader election algorithm [2] is used.

After these modifications, the relation $\rightarrow$ between each two neighboring nodes on a bypass cycle has similar properties to the relation $\rightarrow$ on $BR(1)$. In particular, relation $\rightarrow$ on the bypass cycle is acyclic and there are only two neighbors n_u, n_v on the cycle such that neither $n_u \rightarrow n_v$ nor $n_v \rightarrow n_u$ is true. With these properties, it can be shown that all partitions induced by simultaneous failure of multiple adjacent nodes in MT are reconnected into one tree.

5 Repair methods

Bypass rings used to repair the multicast tree network MT together with the $\rightarrow$ relation create a platform that can be used as a basis for failure repair strategies responsible for constructing new core tree edges as they select node n_{i_r} (see Alg. 1) influencing the topology of the repaired MT. For convenience, the set of new core tree edges constructed during the repair process is further denoted as SE.

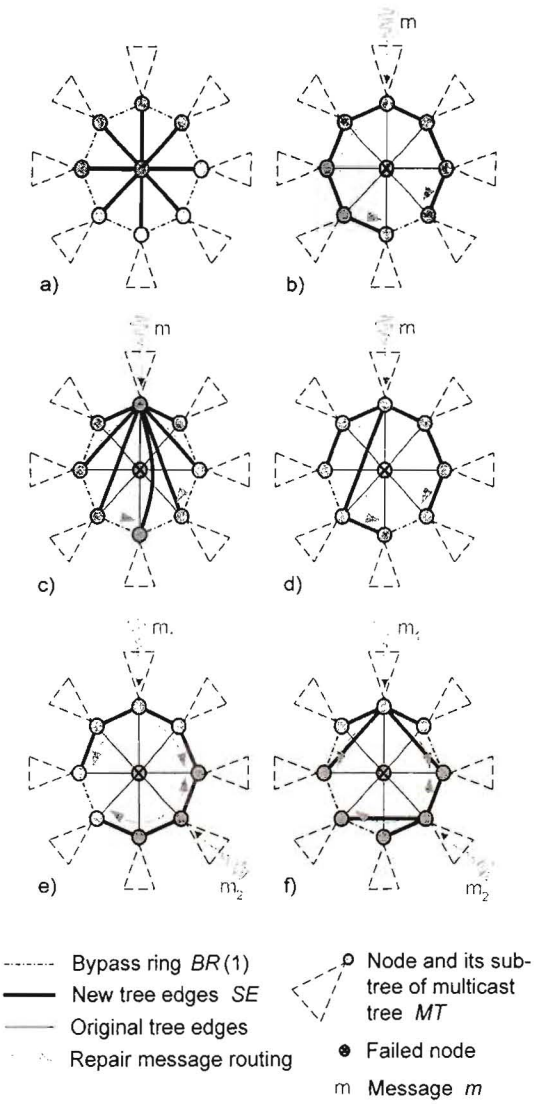

Fig. 3: Methods of multicast tree repair using a two-way single failure repair algorithm
a) Original multicast tree
b), c), d) LRM, TRM and HRM methods of repair initiated at a single node
e), f) LRM and TRM methods, repair initiated simultaneously at two nodes

Based on requirements for optimization of the multicast tree, one of the following approaches to SE creation can be chosen:

- *LRM method.* All member nodes of $BR(1)$ (or the bypass cycle in the case of multiple failure) together with SE form a linear tree (Fig. 3 b), e)). Node n_{i_r} is substituted by $n_{i_{q-1}}$. This approach is preferable if optimization priority is not to increase the degree of nodes in MT (i.e. the number of incident edges).
- *TRM method.* All nodes n_i on BC are connected to $R(n_i)$ and also to $R(n_j)$ if there is a node n_j such that $n_i \rightarrow n_j$ and $R(n_j) \neq R(n_i)$ (Fig. 3 c), f)). Node n_{i_r} is substituted by $R(n_i)$ in Alg. 1. This method is preferable if optimization priority is to retain the mutual distance of nodes n_i and thus not to increase the MT diameter.

- *HRM method.* This method is a combination of both LRM and TRM methods. Specifically, it is similar to TRM except that the path from $R(n_i)$ to n_i may contain one or more intermediate nodes n_j with $R(n_j) = R(n_i)$. The branches of the newly constructed core tree edges rooted at $R(n_i)$ may thus be longer than one hop (Fig. 3 d)).

The appropriate strategy for SE construction can be chosen autonomously by all repair-initiating nodes based on their current local state.

6 Discussion

Bypass rings $BR(1)$ are sufficient for the repair of a single node failure. Generally, a $BR(r)$ scheme is sufficient for the repair of a cluster of faulty nodes with a diameter less or equal to $2(r-1)$. However, because of the repair message routing mechanism 'switching' between bypass rings centered at different faulty nodes, bypass rings $BR(r)$ can repair an even larger cluster of faulty nodes under convenient circumstances. For any repair to be successful, there has to be a path not containing faulty nodes in the underlying routing infrastructure between each two neighboring nodes on a bypass cycle.

The proposed bypass ring scheme is tolerant to failures of nodes on $BR(1)$ (or the bypass cycle) even if the corresponding repair is still in progress. This property is ensured by the fact that core tree edges can be constructed immediately after a node n is notified about failure, and thus it is possible to update $BR_n(r)$ appropriately.

Construction of a complete bypass ring $BR(r)$ (together with all $BR(q)$, $q = 1, \ldots, r-1$) takes $O(r)$ steps and needs $O(b^r)$ messages, where b is the average branching factor in MT. However, ring construction can be done together with MT construction and thus the communication overhead can be substantially reduced. The asymptotical memory overhead of $BR(r)$ is $O(b^r)$, but it can be reduced to $O(br)$ by suitable implementation of BR routing tables.

Failure repair can be done in the worst case in $O(s^2b)$ steps. The time to perform a repair is inversely proportional to the number of nodes simultaneously initiating the repair, and it is further reduced if the repair message is routed in both directions in $BR(1)$ or a bypass cycle (two-way algorithm).

Reduced bypass rings may lower the memory overhead and construction cost to $O(b)$ at the expense of the fault-tolerance level. Note that even complete $BR(2)$ rings provide a substantial level of fault-tolerance for many applications. Additionally, the higher the average branching factor is, the higher is the probability that clusters with a diameter greater than $2(r_{max}-1)$ will be successfully repaired.

7 Conclusion

We have proposed a fault-tolerant scheme for tree-topology communication networks based on bypass rings of optional radii that are used to repair the tree when a node or cluster of nodes fails. For a single failure model, a practical repair algorithm was presented.

As this scheme guarantees that all tree partitions induced by faulty nodes are reconnected and that the repaired network does not contain cycles, it can be used as generic platform for various strategies for reconnecting tree partitions. Three strategies for creating new core edges were briefly discussed.

Multicasting is not the only application where our scheme can be successfully deployed. We argue that our scheme can be used generally by all applications using tree-topology overlay networks to connect or communicate between their components, since the scheme is independent of message source and traffic direction, and the repair can be done in real-time without a significant delay penalty. The only requirement is an underlying infrastructure providing a routing service.

Our future work in this area will include simulation of standard tree traffic patterns, evaluation of the performance of the scheme under various workloads, and a comparison of the effectiveness of various strategies for partition reconnection in terms of external tree optimization requirements. We would also like to refine repair algorithms for reduced bypass rings and measure the impact of reduction on the fault-tolerance level provided.

Acknowledgement

This work was elaborated as a part of the Gaston project at the Czech Technical University in Prague. Gaston [3], [4] is a peer-to-peer large-scale file system designed to provide a fault-tolerant and highly available file service.

Symbols

BC	Bypass cycle
BE	Set of bypass edges
be_i	Bypass edge i
$BR_c(r)$	Bypass ring of radius r centered at node c
CE	Set of core tree edges
ce_i	Core tree edge i
E	Set of all edges in underlying network SN
EMT	Extended multicast tree
FC	Cluster of faulty nodes
gcp	Greatest common prefix function
HID_n^c	Hierarchical identifier of node n related to node c
ID_n	Identifier of node n
k	Size of multicast group
MG	Set of nodes in multicast group
MT	Multicast tree
n_i	Node i
P_i	Path i
$pref$	Prefix function
R	Function R
r	Radius of bypass ring
r_{max}	The greatest bypass ring radius deployed
s	Size of faulty cluster
SE	Set of new core tree edges constructed during repair process
SN	Underlying network
$suff$	Suffix function
V	Set of all nodes in underlying network SN

References

[1] Ballardie, A., Francis, P., Crowcroft, J.: *Core Based Trees (CBT)*. In Proc. of ACM SIGCOMM '93, 1993.

[2] Chang, E. G., Roberts, R.: *An Improved Algorithm for Decentralized Extrema-Finding in Circular Configuration of Processors*. In Comm. of the ACM, Vol. 22, No. 5, 1979.

[3] Dynda, V., Rydlo, P.: *Fault-Tolerant Data Management in a Large-Scale File System*. In Proc. of ISADS 2002, Guadalajara (Mexico), 2002.

[4] Dynda, V., Rydło, P.: *Large-scale Distributed File System Design and Architecture*. In Acta Polytechnica, Vol. 42, No. 1, 2002.

[5] Fei, A., Cui, J., Gerla, M., Cavendish, D.: *A Dual-Tree Scheme for Fault-Tolerant Multicast*. In Proc. of ICC 2001, Helsinki (Finland), 2001.

[6] Jia, W., Zhao, W., Xuan, D., Xu, G.: *An Efficient Fault--Tolerant Multicast Routing Protocol with Core-Based Tree Techniques*. In Proc. of IEEE ICPP, Wakamatsu (Japan), 1999.

[7] Kawamura, R., Sato, K., Tokizawa, I.: *Self-Healing ATM Networks Based on Virtual Path Concept*. In IEEE J. Selected Areas In Comm., 1997.

[8] Mehra, P., Chatterjee, S.: *Efficient Data Dissemination in Ocean Store*. **http://www-video.eecs.berkeley.edu/~pmehra/classes/cs262/paper.pdf**, 2001.

[9] Murakami, K., Kim, H. S.: *Optimal Capacity and Flow Assignment for Self-Healing ATM Networks Based on Line and End-To-End Restoration*. In IEEE Trans. Networking, 1998.

[10] Rowstron, A. et al: *Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-To-Peer Systems*. In LNCS, 2001.

[11] Wu, C., Lee, W., Hou, Y.: *Back-Up VP Preplanning Strategies for Survivable Multicast ATM Networks*. In Proc. of IEEE ICC '97, 1997.

[12] Wu, C., Lee, W., Hou, Y., Chu, W.: *A New Preplanned Self-Healing Scheme for Multicast ATM Network*. In Proc. of IEEE ICC '97, 1997.

[13] Zhao, B. Y., Kubiatowicz, J. and Joseph, A. D.: *Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing*. U. C. Berkeley Technical Report UCB/CSD-01-1141, **http://www.cs.berkeley.edu/~ravenben/tapestry.pdf**, 2001.

[14] Zhuang, S. et al: *Bayeux: An Architecture for Scalable and Fault-Tolerant Wide-Area Data Dissemination*. In Proc. of NOSSDAV '01, 2001.

Ing. Vladimír Dynda
phone: +420 224 357 616
fax: +420 224 923 325
e-mail: xdynda@fel.cvut.cz

Department of Computer Science and Engineering

Czech Technical University in Prague
Faculty of Electrical Engineering
Karlovo náměstí 13
121 35 Prague 2, Czech Republic