Electric Power Systems Research 187 (2020) 106484

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

A network design algorithm for multicast communication architectures in smart transmission grids

Charles M. Adrah*,[a], David Palma[a], Øivind Kure[b], Poul E. Heegaard[a]

[a] Department of Information Security and Communication Technology, Norwegian University of Science and Technology, NTNU, Trondheim, Norway
[b] Department of Technology Systems, University of Oslo, UiO, Oslo, Norway

ARTICLE INFO

Keywords:
Smart transmission grids
Heuristics algorithms
Communication networks
Multicast
Wide-area networks
Substation automation

ABSTRACT

In future smart transmission grids, there are distributed applications that will benefit from the deployment of Internet Protocol (IP) multicast technology for communication. Sharing of Routable-Sample Values (R-SV) and Routable-GOOSE among the digital substations for wide-area monitoring, protection, and control (WAMPAC) applications will be needed. Using multicast for distribution of R-SVs is resource-efficient and offers a simpler configuration with only the interested substations needing reconfiguration. However, the demands for such concurrent delivery of R-SV data will put constraints on the underlying supporting networking infrastructure. For example, it must be ensured that the paths taken to route data traffic are within the bounds of delay to achieve the aims of the WAMPAC application. In this paper, we look at the problem of network topology augmentation through link additions. We present a heuristic algorithm that finds a set of links to be added to a network topology such that the multicast distribution tree for a multicast configuration is bounded by latency, which is set as the hop-count threshold. Our results show that by adding a few new links to the network topology, the delay incurred by the multicast traffic from sources to destinations can be reduced.

1. Introduction

Critical infrastructures such as the smart transmission grid (STG) will depend on supporting communication networking infrastructures. In STG operations, deploying Internet Protocol (IP) multicast technologies as a solution for wide-area monitoring, protection, and control (WAMPAC) applications have been proposed [1,2]. This will become increasingly common and relevant as more substations adopt the IEC 61850 standardization in substation automation [3,4]. The IEC technical report 61850-90-5 [5] specifies how the data models of Generic Object Oriented Substation Events (GOOSE) and Sampled Values (SV) can be routed beyond the substation into wide-area networks, with the addition of UDP/IP headers. These new data models are referred to in the literature as Routable-GOOSE (R-GOOSE) and Routable-Sampled Values (R-SV) messages.

From the Information and Communication Technology (ICT) network perspective, multicast offers several benefits. One of these is bandwidth efficiency as only one copy of R-SV is sent over a link into the network from a source substation to the numerous interested receivers, instead of sending multiple copies from the source substation. R-SVs have delay restrictions to satisfy WAMPAC application requirements. Hence, quality of service (QoS) guarantees, such as multicast admission control and resource reservation techniques, will be needed [6]. Therefore, there is a better utilization of the network resources as maintaining one reservation for the one copy of R-SV sent over the link instead of maintaining reservation for each of the multiple copies. Also, there is the benefit of simplified network configurations. With multicast, only the interested receivers need to change their configurations whenever they want to join or leave a multicast session. The source substation does not need to alter its configurations. If unicast were used, both the source substation and interested receivers would have to be reconfigured whenever a new receiver is interested in R-SVs.

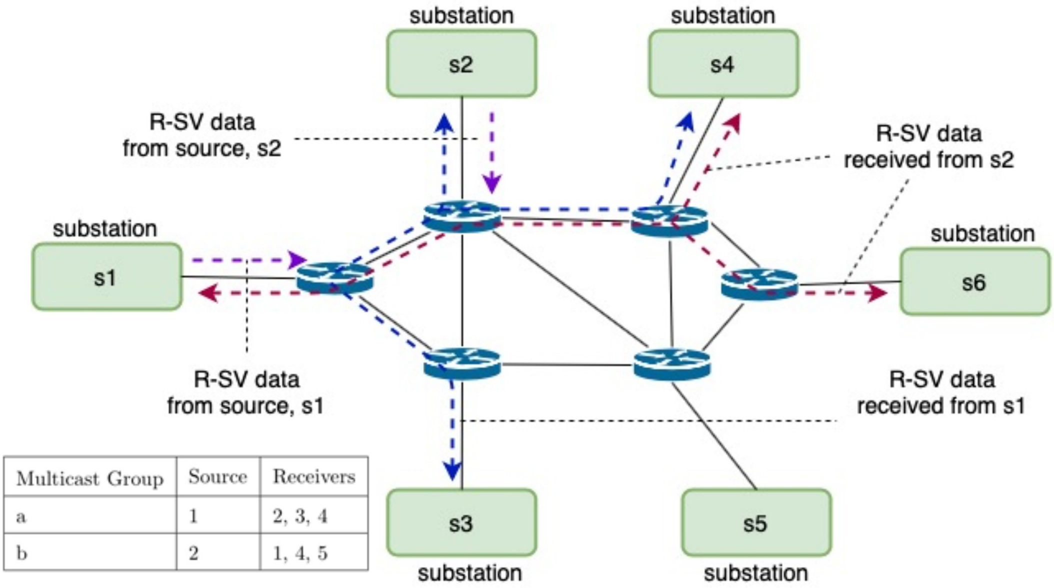
Fig. 1 illustrates how multicast technology is deployed for a WAMPAC application in the STG. It depicts a transmission grid consisting of some substations interconnected by a supporting wide-area communication infrastructure. The substations will want to share and receive R-SV data from other substations. For example, multicast group *a* with substation s1 as a multicast source publishes R-SV data into the network, which are transparently shared using IP multicast to substations *s*2, *s*3, and *s*4. Similarly, substation *s*1 subscribes to receive R-SV data from multicast group *b*, with substation *s*2 as the multicast source.

There are challenges however as to how the communication network should be designed for such multicast deployments. This is as a

---

* Corresponding author.
E-mail address: charles.adrah@ntnu.no (C.M. Adrah).

https://doi.org/10.1016/j.epsr.2020.106484
Received 5 December 2019; Received in revised form 10 May 2020; Accepted 18 June 2020



Multicast Group	Source	Receivers
a	1	2, 3, 4
b	2	1, 4, 5

Fig. 1. WAMPAC application deploying IP multicast in the smart transmission grid.

result of the strict QoS requirements such as latency, packet delay variations, packet losses, availability and path redundancy, imposed on the WAMPAC applications [7,9,8]. Constraints such as delays incurred through different link costs or processing delays on nodes become limitations affecting the latency path for the multicast traffic, which must be addressed.

A method to address delay constraints is to augment the network topology by adding new links to the topology. This can ensure that the delay incurred on the multicast traffic from the source(s) to their destinations is limited to a maximum number of hops. The approach known as *network topology design* has received significant attention in graph theory and network science communities. The challenges of minimizing the diameter of such graphs or network topologies by the insertion of edges or links of bounded costs are problems that exist in practical application fields such as telecommunication networks, information networks, road networks, and flight scheduling [10,11].

In this paper, we look at the problem of network topology augmentation through link additions. We present a heuristic algorithm that finds a set of links to be added to a network such that the multicast distribution trees formed from multicast configurations are bounded by latency. Here, the latency is set as the number of hops in the shortest paths (i.e., hop-count threshold). The algorithm minimizes the maximum shortest path lengths from a group of multicast destinations to one or more multicast sources. We evaluate the performance of our algorithm over some ICT network topologies for an IEEE-39 transmission grid. The results show that only a few new links are needed to be added to the topologies to meet set delay requirements. Hence, communication network design through topology augmentation can improve the delivery of multicast traffic in smart transmission grids.

The rest of the paper is organized as follows: We present related work on network design in Section 2. Our heuristic algorithm is presented in Section 3, and we show how the algorithm works with an example network topology. Performance evaluation of ICT topological data-sets for an IEEE-39 transmission grid is presented in Section 4. We present discussions in Section 5 and finally give concluding remarks in Section 6.

2. Related work

Network graph or topology design and optimization involve improving the network design with some defined objectives, either be rewiring while maintaining constant edges or by adding new links to improve the connectivity of the networks [12,13]. This is shown to be an NP-hard problem [14]. Adding a set of links or nodes to the graph to optimally maximize a certain graph property is known in the literature as graph augmentation [15–17]. Research works have focused on augmentation of network topologies for two purposes. The first involves improving fault tolerance and robustness of networks [18–20]. The second involves analyzing information flow properties such as minimization of eccentricity and diameter [21,22], and average shortest path length [23]. The scope of our work addresses the latter, where we minimize the end-to-end delays of groups of multicast sources and destinations.

In [21], the problem of designing a composite network to minimize the maximum of shortest path lengths from a traffic source to its specified destinations by adding up to B edges to the information flow structure is addressed. The set of possible added edges is a subset of the edges in a complement graph where a complement graph is the all edges not in the initial network. The maximum of the shortest path lengths from a traffic source to its destinations is the maximum of the delay, or hops, required for traffic propagation. Thus, minimizing this maximum implies reducing the end-to-end delay suffered by the message. The approach used in determining the new edges to be added to the network shows that the newly added edges are incident on the source such that any shortest path from source will use at most one of the newly added edges.

In [22], a clustering algorithm (see Section 4.1) is used to minimize the diameter of a network using up to B constant shortcut edges, with the set of allowable edges added being edges in the complementary graph. With a clustering algorithm to minimize edges from a single source, new-formed edges are incident on the source. Also, edges are formed by connections from the source to the center of the formed cluster(s). Furthermore, the paper shows that solving the single source eccentricity minimization problem, as well as the multicast version, can be done with this same approach.

In [23], the problem of adding k shortcut edges of small fixed length to a graph to minimize the weighted average shortest path distance over all pairs of vertices was studied. In the single source version for this, it is shown that there exists an optimal set F^* such that each edge, $e \in F^*$, is incident on the source s, hence for all other vertices, $v \in V$ there exists a shortest $s \leftrightarrow v$ path that uses at most one edge in F^*.

From the reviewed works, the methods that have been suggested in minimizing the eccentricity or diameter of a graph are done mostly by forming new edges that are incident on the source nodes. In minimizing the diameter for multiple single-source multicast groups defined in the same graph, this approach may not always produce the best approximations since there might be links which are not incident on the sources that can better minimize the eccentricity for the multicast groups. The algorithm proposed in this paper suggests new links for multiple single-source multicast configurations defined over a topology without always finding a solution incident on the sources.

3. Minimizing end-to-end delay in the network

In this section, we present a heuristic algorithm, which we call *reduction over minimum set cover* (ROMSC). ROMSC algorithm improves a given network topology end-to-end delay through topology augmentation. As such, the delay incurred in the delivery of multicast traffic in the network is reduced. The algorithm's objective is to find a set of best link(s) to be added in a topology network design to enable the delivery of multicast traffic within delay bounds. The delay is defined as the maximum number of hops (hop-count threshold) in the shortest paths trees that can exist in the multicast distribution tree between a source node and the destination nodes. The end-to-end delay in a network is characterized by several delay components such as transmission, propagation, queuing, and processing delays. Hence, setting a limit on the number of routers passed will reduce the end-to-end delay experienced by the multicast traffic. The algorithm finds the best link(s) for a network that has several single-source multicast configurations deployed.

3.1. Algorithm

ROMSC uses the greedy approximation technique for the minimum set cover problem [24] to find the minimum set of links. There are three inputs used in our algorithm: an input network topology, G_i, multicast configurations, *MC*, and the hop-count threshold value, *thresh*. The input topology is defined by nodes and links, while the multicast configuration is the sets of multicast source and destination nodes. This algorithm uses three functions: find_exceeding_pairs(), find_candidate_links() and find_new_links(). The pseudo-code is presented in Algorithm 1.

The function find_exceeding_pairs() takes an input topology G_i, multicast configurations *MC*, hop-count threshold *thresh*, and returns the set of source-destination pairs that exceed *thresh* in all multicast configurations. Multicast trees are built using the reverse shortest path that builds shortest paths from multicast destinations towards the multicast source. We use a Breadth-First Search (BFS) algorithm [25], which is a graph traversal algorithm to find destination nodes that exceed the hop-count threshold. We call the set of source-destination pairs found with this function, the universal set *U*.

The function find_candidate_links() returns a set of candidate links, *C*, from which we find the final solution of new links to be added to our topology. The candidate links are formed by either connecting links directly between the exceeding pairs in *U* or their neighbor nodes. Firstly, the function stores in a table the distances (i.e., number of hops) of all neighbor nodes for all the unique elements in *U*, which are within the hop-count threshold. That is for all neighbor nodes, *i*, that are within the hop-count threshold for each node, *i*, in *U*, we find the distances, *val*, The value of the distances are stored as such; $distMap[i][j] = val$. Secondly, the function forms candidate links from the table for an exceeding pair *k* in *U*, satisfying the condition;

- $distMap[a][k.\,src] + distMap[b][k.\,dest] < thresh$.

Fig. 2 illustrates how candidate links are formed. That is for an exceeding pair *k* in *U* with a multicast source *k.src* and multicast destination *k.dest*, a candidate link *a*↔*b* is formed. The nodes *a* and *b* are either the source and destination nodes, i.e., *k*, or their neighbor nodes

```
Result: S:= list of links to be added to topology
input : $G_i$ := input network topology; $MC$:= multicast configurations or groups; $thresh$ := hop-count threshold;
Functions(find_exceeding_pairs(), find_candidate_links(), find_new_links())
begin
    $U$ = find_exceeding_pairs($G_i$, thresh, $MC$);
    $C$ = find_candidate_links($G_i$, $U$);
    $S$ = find_new_links($U, C$);
    return $S$
end
Function find_new_links(U, C)($F$):
    $S = \{\}$                                             ▹ empty solution list
    while $U$ is not empty do
        select a link, $l_i$, in $C$ that covers most elements $e$, in $U$, starting with the element, $e$, with the highest hop-count away
        from the source; $S$.add($l_i$); for $e$ in $U$ covered by $l_i$ do
            remove $e$ in $U$
        end
    end
    return $S$
End Function
```

Algorithm 1. ROMSC algorithm.

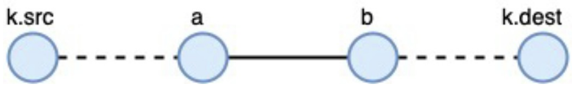


Fig. 2. Formation of candidate links.

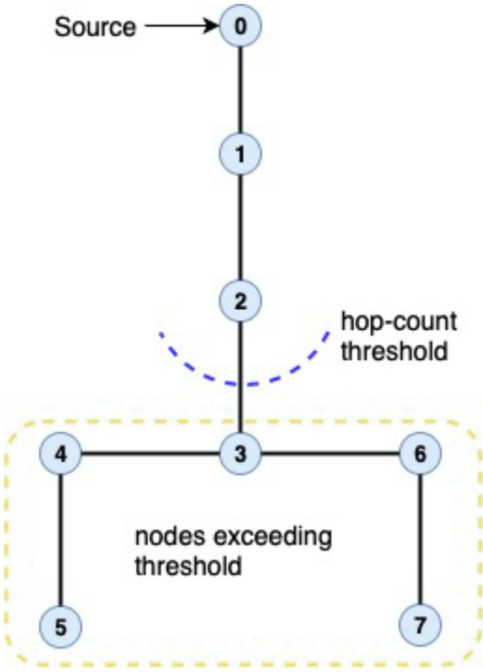


Fig. 3. 8 node network.

within the hop-count threshold.

The find_new_links() function returns the set of new links that can be added to the topology to satisfy all the source-destination node pairs exceeding the set threshold, i.e., U. The strategy used is the minimum set cover approximation, starting by finding a solution for source-destination pairs having the highest hop-count from the sources. This is followed by the link that covers the source-destination pair with the next highest hop-count and the remaining elements in U. The process is repeated until all elements in U are covered, and a set of links returned as the solution.

3.2. ROMSC algorithm evaluation

In this section, we explain how the ROMSC algorithm works by deploying it on a small size topology. Fig. 3 shows a network topology consisting of 8 nodes and 7 edges. Let us assume the hop-count threshold is set as 2 and that a multicast configuration is defined as follows; source node is 0, and destinations nodes are {1, 2, 3, 4, 5, 6, 7}.

First, by using the function *find_exceeding_pairs()* the list of destination nodes exceeding the threshold returned is: {3, 4, 5, 6, 7}. Hence, the universal set of source-destination pairs exceeding the threshold is:

- $U = \{(0, 3), (0, 4), (0, 5), (0, 6), (0, 7)\}$

Second, we use the function *find_candidate_links()*. This function first creates the table of distances of neighbor nodes within the hop-count threshold for each unique element in U. Let us call the set of all unique elements from U as $U_{flattened}$. This table takes each element in $U_{flattened}$ and finds distances of neighbor nodes that are within the hop-count threshold, i.e., the elements {0, 3, 4, 5, 6, 7} are used to create this table. Table 1 shows a square matrix of the 8-node network, which illustrates how the table is generated. For example, node 0 from $U_{flattened}$ has neighbor nodes 1 and 2 with distances 1 and 2 stored. Another example is node 7, which has neighbor nodes 3 and 6 with distances 2 and 1 respectively stored in the table. Each node in $U_{flattened}$ has also the distance 0 from itself stored in the table. The function then next finds the candidate links of each source-destination pair in U. For example, taking the pair (0,3) from U, we form candidate links, (i, j), if:

- $distMap[i][0] + distMap[j][3] < 2$

Table 1
$distMap[i][j] = val$, where i is a neighbour node of j, j is a node in $U_{flattened}$, val is hop-count from j.

i \ j	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	4	5
1	1	0	1	2	3	4	3	4
2	2	1	0	1	2	3	2	3
3	3	2	1	0	1	2	1	2
4	4	3	2	1	0	1	2	3
5	5	4	3	2	1	0	3	4
6	4	3	2	1	2	3	0	1
7	5	4	3	2	3	4	1	0

Using this condition produces candidate links that can satisfy the pair (0,3); 0↔2, 0↔3, 0↔4, 0↔6 and 1↔3. Table 2 shows all candidate links for the topology, and the exceeding pairs covered in U by adding such a link.

Third, we use the function *find_new_links()* to find the solution of the best links to be added to the network topology. With the universal set, $U = (0, 3), (0, 4), (0, 5), (0, 6), (0, 7)$, the function uses a set cover approximation strategy to find a set of links that covers the elements in U, starting first with elements with the highest hop-count.

From Fig. 3, nodes 5 and 7 have the highest hop-count (i.e., 5), away from the source, 0. Hence, taking node 5, we find that the link 0↔4 will cover elements (0,3),(0,4),(0,5). The remaining elements in U are then (0, 6), (0, 7). Again we find a solution for the node with the next highest hop-count away from the source. It is node 7. Hence, we find the best link that covers the remaining elements in U, inclusive of the pair {0, 7}. The link 0↔6 is selected to cover the rest of the elements in U. The final solution of new links to be added to the network graph will thus be 0↔4 and 0↔6.

4. Performance evaluation and discussion

In this section, we apply our algorithm presented in Section 3.1 by

Table 2
Candidate links (C) and exceeding pairs (U) satisfied.

C \ U	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)
0↔2	x				
0↔3	x	x		x	
0↔4	x	x	x		
0↔5		x	x		
0↔6	x			x	x
0↔7				x	x
1↔3	x				
1↔4		x			
1↔5			x		
1↔6				x	
1↔7					x

evaluating sets of communication network topologies that can be deployed in the STG. The physical power grid for transmission networks is usually well planned, and as such, operations involving WAMPAC applications in the grid should be carefully planned before deployment. Therefore, when deploying multicast configurations, the sources of multicast traffic (i.e., substations sending R-SV data), and the multicast traffic receivers (i.e., substations receiving R-SV data) should also be predetermined.

The evaluation is done by setting different values of hop-count thresholds required to achieve multicast delivery for a multicast configuration in a network topology. Our algorithm, ROMSC, determines the number of links to be added to the network topology to meet the latency demands for the delivery of multicast traffic. We compare ROMSC and clustering algorithm from Demaine et al. [22] in our evaluations. Both algorithms were implemented in C++ and our code compiled with gcc-5.4.0. The tests were done on a 32 *bit* Ubuntu Linux machine equipped with 8 *GB* RAM, and a 2.2 *GHz* Intel core. A summary of the clustering algorithm and its implementation is presented next.

4.1. Clustering algorithm (CLUS)

Given a distance or hop-count $dist(x) \geq 0$, the CLUS algorithm [22] involves partitioning the vertices, V, and edges, E, of an undirected weighted graph, $G = (V, E)$ into clusters of diameters at most $2x$. A subset of vertices S, from G, is selected as the centers of the clusters. The set S satisfies the following properties; 1) the distance between any pair of vertices in S should be greater than $2x$. 2) for every vertex $u \ni S$, there should be a vertex $v \in S$ whose distance to u is at most $2x$, where $dist(u, v)$ is the distance between u and v. Otherwise, vertex u is added to the set S as a cluster center.

In the implementation of the single-source version of the multicast problem, only a subset of the nodes is minimized. That is for the subset of vertices, $V' \subseteq V$, and a source node s, we want to add k shortcut edges to minimize the maximum distance between the nodes in V' from the source s. In this case, the centers of the clusters are selected from the vertex set V'. Hence any vertex outside is not chosen as a center in the algorithm. The CLUS stops when all vertex nodes of the set V' are exhausted or cannot be selected.

4.2. IEEE 39-bus transmission grid and communication network topology

In Adrah et al. [2], we defined a method of constructing physical-level communication topologies for an IEEE 39-bus transmission test system. These were assumed fiber-level topologies having a general structure of substations grouped into rings connected to a common core network. Each substation is represented by an edger router that connects to other substation edge routers to form the wide-area communication topology. Four different communication topologies were defined, as shown in Figs. 4 and 5, and will be used in our evaluation. The main differences between these topologies are the number of groups of rings formed, and the additional links added to the core node for improved connectivity.

Also, we define two sets of multicast configurations to be deployed on these topologies. The first multicast configuration (MC-1) consists of all 39 nodes acting as multicast sources, and multicast receivers defined as all neighbor nodes directly connected to a multicast source. The average number of receivers per multicast group is 2. For the second multicast configuration (MC-2), 39 nodes acting as multicast sources and the multicast receivers are defined to include all nodes that are at most 2 nodes away from their respective multicast sources. MC-2 has the average number of receivers per multicast group as 6. The details of the topologies with the number of nodes, links, the average number of hops per multicast configuration, and maximum candidate links of the topologies are presented in Table 3. The defined multicast configurations are shown in the Appendix (Tables 4 and 5).

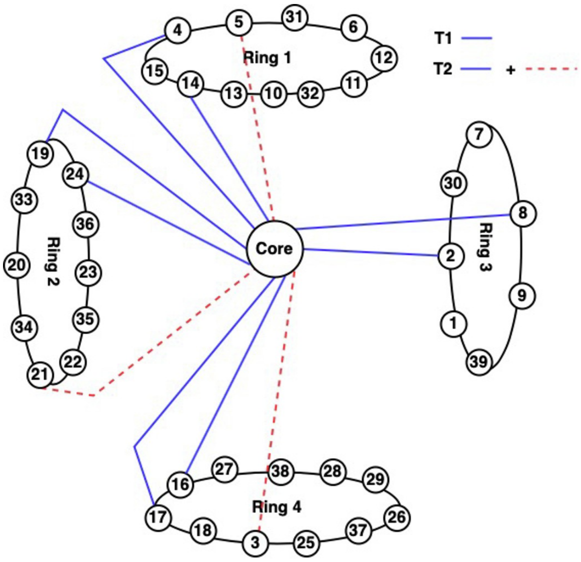


Fig. 4. Communication network topologies T1 and T2.

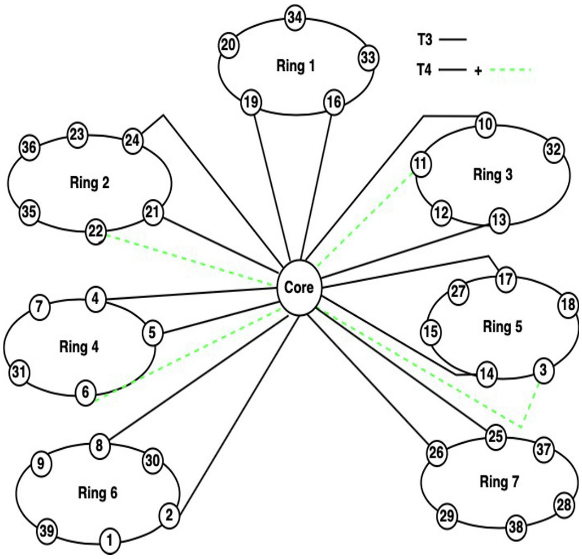


Fig. 5. Communication network topologies T3 and T4.

Table 3
Topological Data-set.

Network	Nodes	Links	Avg. Degree	Avg. no. of hops		Candidate links
				MC-1	MC-2	
T1	40	47	2.35	1.91	2.79	733
T2	40	50	2.5	1.69	2.44	730
T3	40	53	2.65	1.69	2.38	727
T4	40	57	2.85	1.65	2.29	723

4.3. Results

Figs. 6 and 7 show plots of the minimum number of links to be added to the four network topologies deployed with multicast

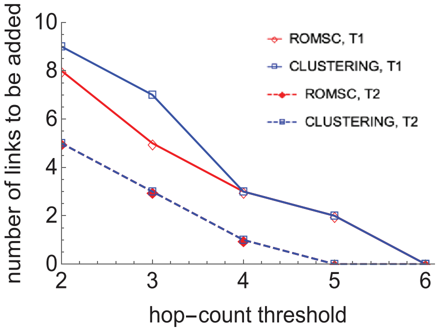


Fig. 6. T1 and T2 with MC-1.

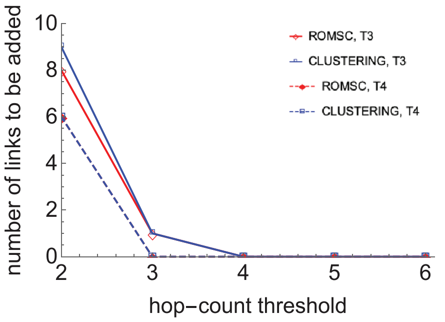


Fig. 7. T3 and T4 with MC-1.

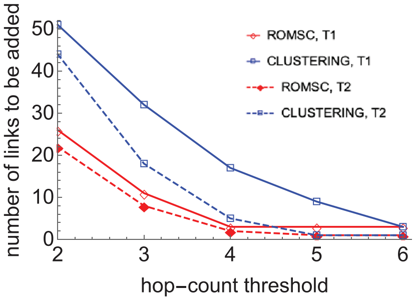


Fig. 8. T1 and T2 with MC-2.

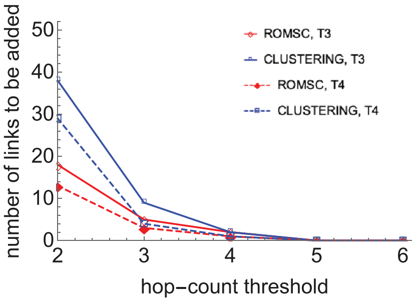


Fig. 9. T3 and T4 with MC-2.

configuration 1 (i.e., Table 4). The number of hops in the shortest path or hop-count threshold is varied from 2 to 6. As can be seen, ROMSC performs better than or equal to the CLUS in approximating the number of links to be added for all values of the hop-count threshold. The topologies T2 and T4 are improvements of T1 and T3, respectively, in terms of design, and were observed to have equal performance using both algorithms. This can be attributed to the additional links in T2 and T4, which improved their average degree compared to T1 and T3.

We also evaluate the performance of ROMSC and CLUS over the same topological data-set deployed with multicast configuration 2 (i.e., Table 5). Figs. 8 and 9 show the number of minimum links to be added with the hop-count threshold varied from 2 to 6. Similarly, we observe that ROMSC performs better than or equal to the CLUS. However, the difference in the number of minimum links suggested by ROMSC and CLUS is more significant in these cases. This is because of the complexity of the multicast configuration used. MC-2 is more complex than MC-1, having a higher average number of multicast receivers per group. MC-2, when deployed on the topologies generally resulted in a higher number of links required to be added to the network, compared to using MC-1. It is observed that ROMSC suggests a significantly fewer number of links compared to CLUS for small hop-count thresholds. For example, when the hop-count threshold is 2 ROMSC gives 26 links against 51 links with CLUS for T1 and likewise ROMSC gives 22 links against 44 links with CLUS for T2. A similar trend is observed with T3 and T4, where ROMSC gives 18 and 13 links against 38 and 29 links using the CLUS.

With a more complex multicast configuration, the percentage of the number of new links to all source-destination pairs exceeding the hop-count threshold is less for ROMSC compared to CLUS. For T1 with MC-2, source-destination pairs U, was 109 when the hop-count threshold was set to 2. ROMSC generated 23.85% new links per U while CLUS generated 46.79% new links per U.

Furthermore, the underlying physical topology also influenced the number of new links to be added to the network topology. For topologies with a higher average degree of connectivity, there were fewer new links needed to be added for improved multicast delivery. In our analysis, T4 had the highest average degree. Using the ROMSC algorithm, when T4 was deployed with MC-1, and hop-count threshold = 3, no new links were required to achieve the end-to-end delay of multicast traffic. When T4 was deployed with MC-2, and hop-count threshold = 4, only 1 link was needed to be added to the topology. It is also noticeable that the performance of both algorithms tends to converge as the hop-count threshold increases.

4.4. *Analysis of IEEE-39 bus communication network with no initial core*

In this section, we consider a utility whose communication network topology is without a core network. For such a scenario, we assume the communication network topology of the utility grid is geographically grouped into ring structures. We aim to find the minimum number of links to be added to enable connectivity among the rings for the multicast groups to achieve the desired end-to-end delay paths. We do this by modifying the topologies T1, having four-rings and T3, having seven-rings, which are the groupings of the substation edge routers. All links from the ring structures going towards the core network are removed. We call the modified T1 and T3: *T1-no-core* and *T3-no-core* respectively. We deploy multicast configurations MC-1 and MC-2 on *T1-no-core* and *T3-no-core*. Both ROMSC and CLUS algorithms are used in

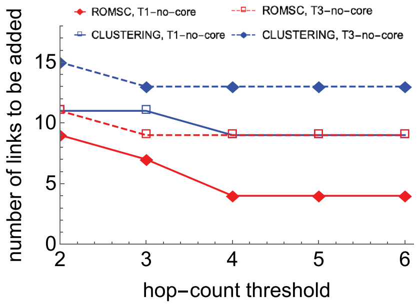


Fig. 10. T1-no-core and T3-no-core with MC-1.

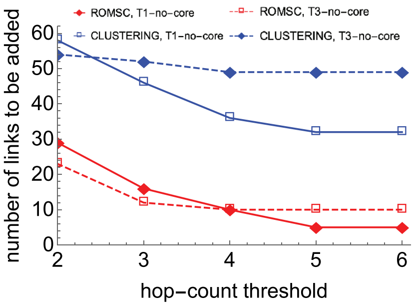


Fig. 11. T1-no-core and T3-no-core with MC-2.

the evaluations.

Figs. 10 and 11 show the number of minimum links added to the topologies when deployed with MC-1 and MC-2, respectively. Again the results show that ROMSC performs better than the CLUS in producing the number of minimum links to be added to the network. We compare the performance of ROMSC on T1-no-core and T3-no-core to study the effects of ring sizes. For MC-1, T1-no-core with a four ring structure consistently produced the minimum number of links for all hop-count thresholds compared with T3-no-core with a seven ring structure. For MC-2, T3-no-core performed better than T1-no-core for the hop-count threshold between 2 and 4. Beyond hop-count = 4, it is observed that T1-no-core produced a minimum number of links as compared to T3-no-core.

5. Discussion

The ICT topologies used in our evaluation were based on IEEE 39-bus STG. Since the STG used in actual deployments are of a limited size, the range of ICT topologies will be equally limited. Our use-case topologies, and cost metric of a hop-count threshold, provide an insight on the expected performance of our algorithm in actual deployment. From the evaluations, the performance of ROMSC is always the same or better, and in some cases, significantly better than the CLUS in reducing the number of links to be added to the network topology.

In our analysis in Section 4.4, with the 39 substation edge routers used to form groups of ring structures, we observed the most significant reduction in the number of added links using ROMSC. Therefore for transmission grid networks that already have existing supporting communication infrastructures, using ROMSC provides a method of finding the minimum links to augment the topology to attain the required end-to-to delay for the multicast traffic in the network. However, such augmented topologies may be sub-optimal. In cases where the network topology is designed just for multicast delivery, it may be possible to achieve fewer links needed to satisfy the latency requirements without consideration for an already existing core network.

Furthermore, the size of the topology ring structures to be created for such networks, together with the multicast configuration complexity, will also affect the number of links required for the topology augmentation. For small multicast configurations of less complexity, large groupings of nodes in rings tend to produce the fewer number of links. With more complex multicast configurations, and for high end-to-to end delay requirements (i.e., small hop-count threshold), the smaller sized rings performed better. However, with lower delay thresholds (i.e., large hop-count threshold), it is observed that the large-sized rings tend to perform better than the small sized ring networks.

In ROMSC, the set of candidate links is reduced. This is because we generate the candidate links only from nodes exceeding the hop-count threshold and their close neighbor nodes that are within the hop-count threshold. The set of all possible candidate links is calculated by $\frac{n(n-1)}{2} - k$ [18], where n is the number of nodes in the topology and k is the existing links in the initial topology. In our 8-node example network with 7 existing links shown in Fig. 3, the maximum number of candidate links is 21. With ROMSC, the candidate links size is reduced to 11, as shown in Table 2. For topology T1 running MC-1, the number of maximum candidate links is 733 since the existing network already has 47 links. When the hop-count threshold is set for the values {2, 3, 4, 5}, ROMSC reduces the candidate link size needed to find the minimum links to {48, 110, 148, 180}, respectively.

Using a brute force approach to obtain the number of minimum links is computationally expensive. If the number of candidate links is n, with existing links k, the maximum iteration to find the number of minimum links using brute force is $\sum_{k=1}^{l} \frac{n!}{(n-k)!k!}$, with a computational complexity of $O(n^n)$. For example, in our 8-node case topology, approximately 2.1 million running iterations will be required with a brute force approach. It was shown in [21] that a brute force enumeration algorithm that determined the eccentricity for a 75-node graph, on adding up to 4 edges and higher, took months to run on the provided hardware.

6. Conclusion

Current and future networking topologies for transmission grids will need to be re-designed as the demand for more complex power system applications arise. For example, WAMPAC applications that require synchronous operations will largely leverage IP multicast technology as a communication solution.

In this paper, we have presented a novel heuristic algorithm, ROMSC, that adds a minimum number of links to the network topology. When we assume a constant delay per hop, the maximum delay for a multicast configuration running in the network can be set. It is shown that by adding a few new links to the network topology, the hop-count threshold is fixed for all multicast traffic in the network.

We demonstrate that ROMSC is more efficient when compared to a CLUS algorithm. Also, ROMSC finds minimum links for multiple multicast configurations defined over the network topology. Hence, it better approximates a minimum number of links to be added to the network. Furthermore using ROMSC, larger network topologies can be easily augmented, which will enable the concurrent delivery of multicast traffic for WAMPAC applications.

CRediT authorship contribution statement

Charles M. Adrah: Conceptualization, Methodology, Investigation, Software, Writing - review & editing. **David Palma:** Conceptualization, Validation, Writing - review & editing. **Øivind Kure:** Conceptualization, Methodology, Writing - review & editing. **Poul E. Heegaard:** Conceptualization, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Table 4
Multicast configuration 1 (MC-1) with sources S, and receivers R.

S	R	S	R	S	R
30	2	15	14,16	11	6,10,12
31	6	18	3,17	13	10,12,14
32	10	20	19,34	14	4,13,15
33	19	21	16,22	17	16,18,27
34	20	24	16,23	19	16,20,33
35	22	27	17,26	22	21,23,35
36	23	28	26,29	23	22,24,36
37	25	39	1,9	25	2,26,37
38	29	3	2,4,18	29	26,28,38
1	2,39	4	3,5,14	2	1,3,25,30
7	6,8	5	4,6,8	6	5,7,11,31
9	8,39	8	5,7,9	26	25,27,28,29
12	11,13	10	11,13,32	16	15,17,19,21,24

Table 5
Multicast configuration 2 (MC-2) with sources S, and receivers R.

S	R	S	R	S	R
34	20,19	29	25,26,27,28,38	11	5,6,7,10,12,13,31,32
37	25,2,26	28	25,26,27,29,38	15	4,13,14,16,17,19,21,24
32	10,11,13	7	5,6,8,9,11,31	14	3,4,5,10,12,13,15,16
33	19,16,20	8	4,5,6,7,9,39	26	2,17,25,27,28,29,37,38
38	29,26,28	18	2,3,4,16,17,27	19	15,16,17,20,21,24,33,34
35	22,21,23	10	6,11,12,13,14,32	3	1,2,4,14,17,18,25,30
36	23,22,24	23	16,21,22,24,35,36	24	15,16,17,19,21,22,23,36
31	6,5,7,11	1	2,3,9,25,30,39	5	3,4,6,7,8,9,11,14,31
30	2, 1,3,25	22	16,21,23,24,35,36	4	2,3,5,6,8,13,14,15,18
39	1,2,8,9	21	15,16,17,19,22,23,24	25	1,2,3,26,27,28,29,30,37
20	16,19,33,34	27	16,17,18,25,26,28,29	2	1,3,4,18,25,26,30,37,39
9	1,5,7,8,39	13	4,10,11,12,14,15,32	17	3,15,16,18,19,21,24,26,27
12	6,10,11,13,14	6	4,5,7,8,10,11,12,31	16	14,15,17,18,19,20,21,22,23,24,27,33

References

[1] M. Seewald, Building an architecture based on ip-multicast for large phasor measurement unit (pmu) networks, 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), (2013), pp. 1–5.

[2] C.M. Adrah, J.R.A.K. Yellajosula, Ø. Kure, D. Palma, P.E. Heegaard, An IP multicast framework for routable sample value communication in transmission grids - Volume 14, no. 9, september 2019 - journal of communications, J. Commun. 14 (9) (2019) 765–772.

[3] R.E. Mackiewicz, Overview of iec 61850 and benefits, 2006 IEEE PES Power Systems Conference and Exposition, (2006), pp. 623–630.

[4] IEC, IEC TR 61850-90-1:2010, Communication networks and systems for power utility automation - Part 90-1: Use of IEC 61850 for the communication between substations, Technical Report, IEC, 2010.

[5] IEC, IEC standard for communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasors information according to IEEE C37.118 IEC 61850-90-5 TR Ed 1.0, Technical Report, IEC, 2012.

[6] A. Neto, E. Cerqueira, A. Rissato, E. Monteiro, P. Mendes, A resource reservation protocol supporting qos-aware multicast trees for next generation networks, 2007 12th IEEE Symposium on Computers and Communications, (2007), pp. 707–714.

[7] P. Kansal, A. Bose, Bandwidth and latency requirements for smart transmission grid applications, IEEE Trans. Smart Grid 3 (3) (2012) 1344–1352.

[8] Y. Deng, H. Lin, A.G. Phadke, S. Shukla, J.S. Thorp, L. Mili, Communication network modeling and simulation for wide area measurement applications, 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), (2012), pp. 1–6.

[9] S.M. Blair, C.D. Booth, B. De Valck, D. Verhulst, K.-Y. Wong, Modeling and analysis of asymmetrical latency in packet-based networks for current differential protection

application, IEEE Trans. Power Delivery 33 (3) (2018) 1185–1193.

[10] S. AhmadBeygi, A. Cohn, Y. Guan, P. Belobaba, Analysis of the potential for delay propagation in passenger airline networks, J. Air Transp. Manage. 14 (5) (2008) 221–236.

[11] F. Frati, S. Gaspers, J. Gudmundsson, L. Mathieson, Augmenting graphs to minimize the diameter, Algorithmica 72 (4) (2015) 995–1010.

[12] A. Sydney, C. Scoglio, D. Gruenbacher, Optimizing algebraic connectivity by edge rewiring, Appl. Math. Comput. 219 (10) (2013) 5465–5479.

[13] H. Wang, P. Van Mieghem, Algebraic connectivity optimization via link addition, Proceedings of the 3rd International Conference on Bio-Inspired Models of Network, Information and Computing Sytems, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, p. 22.

[14] J.M. McQuillan, Graph theory applied to optimal connectivity in computer networks, ACM SIGCOMM Comput. Commun. Rev. 7 (2) (1977) 13–41.

[15] K.P. Eswaran, R.E. Tarjan, Augmentation problems, SIAM J. Comput. 5 (4) (1976) 653–665.

[16] M. Papagelis, F. Bonchi, A. Gionis, Suggesting ghost edges for a smaller world, Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM ’11, ACM, New York, NY, USA, 2011, pp. 2305–2308.

[17] N. Parotsidis, E. Pitoura, P. Tsaparas, Selecting shortcuts for a smaller world, SIAM International Conference on Data Mining 2015, SDM 2015, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015, pp. 28–36.

[18] M.J. Alenazi, E.K. Çetinkaya, J.P. Sterbenz, Network design and optimisation based on cost and algebraic connectivity, International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, IEEE, 2013, pp. 193–200.

[19] M.J. Alenazi, E.K. Cetinkaya, J.P. Sterbenz, Cost-Efficient network improvement to achieve maximum path diversity, Proceedings of 2014 6th International Workshop on Reliable Networks Design and Modeling, RNDM 2014, IEEE, 2014, pp. 202–208.

[20] M.J. Alenazi, J.P. Sterbenz, Evaluation and comparison of several graph robustness metrics to improve network resilience, Proceedings of 2015 7th International Workshop on Reliable Networks Design and Modeling, RNDM 2015, IEEE, 2015, pp. 7–13.

[21] S. Perumal, P. Basu, Z. Guan, Minimizing eccentricity in composite networks via constrained edge additions, Proceedings - IEEE Military Communications Conference MILCOM, IEEE, 2013, pp. 1894–1899.

[22] E.D. Demaine, M. Zadimoghaddam, Minimizing the diameter of a network using shortcut edges, in: H. Kaplan (Ed.), Algorithm Theory - SWAT 2010, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 420–431.

[23] A. Meyerson, B. Tagiku, Minimizing average shortest path distances via shortcut edge addition, Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX ’09 / RANDOM ’09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 272–285.

[24] V. Chvatal, A greedy heuristic for the set-covering problem, Math. Oper. Res. 4 (3) (1979) 233–235.

[25] C.Y. Lee, An algorithm for path connections and its applications, IRE Trans. Electron. Comput. EC-10 (3) (1961) 346–365.