A Hierarchical Autonomous System Based Topology Control Algorithm in Space Information Network

Wei Zhang[1], Gengxin Zhang[1], Liang Gou[1], Bo Kong[1] and Dongming Bian[1]
[1] College of Communication Engineering, PLA University of Science and Technology
Nanjing, China
[e-mail: zev@msn.com, satlab@126.com, gouliang880@163.com, kbvx_123@163.com, bian_dm@163.com]
*Corresponding author: Gengxin Zhang

Received April 23, 2015; revised July 12, 2015; accepted July 18, 2015; published September 30, 2015

Abstract

This article investigates the topology control problem in the space information network (SIN) using a hierarchical autonomous system (AS) approach. We propose an AS network topology control (AS-TC) algorithm to minimize the time delay in the SIN. Compared with most existing approaches for SIN where either the purely centralized or the purely distributed control method is adopted, the proposed algorithm is a hybrid control method. In order to reduce the cost of control, the control message exchange is constrained among neighboring sub-AS networks. We prove that the proposed algorithm achieve logical *k*-connectivity on the condition that the original physical topology is *k*-connectivity. Simulation results validate the theoretic analysis and effectiveness of the AS-TC algorithm.

Keywords: Space information network, topology control, autonomous system, time delay, strong connectivity

This work was supported by NSF of China under Grant No. 91338201 and 91438109.

http://dx.doi.org/10.3837/tiis.2015.09.016 **ISSN : 1976-7277**

1. Introduction

The space information network (SIN) is a new type of self-organizing system constituted by information systems of space, air, land and sea [1-4]. It is proposed to solve the problems that current space systems' lacking of universality and the inconvenience in cooperation between different missions. SIN accepts various kinds of platforms as network nodes, such as satellites, high altitude platforms (HAPs), aerial vehicles and terrestrial sensors [5]. Due to the distinguishing characteristics of SIN such as large scale, high component complexity and dynamic, how to make an efficient and reliable topology control of SIN is challenging.

Existing works about network topology control mainly focus on maintaining a specified topology to achieve a set of network-wide objectives such as reducing energy consumption, guaranteeing the robustness, increasing the network capacity and reducing end-to-end delay, e.g., [6-14]. But in most studies, either the purely centralized or the purely distributed control method is adopted. Centralized algorithms rely on a central entity which knows the conditions of all the nodes in order to calculate the optimal topology [15-17]. However, these algorithms are not suitable for large scale networks such as SIN where excessive amount of control messages need to be collected by the central entity. On the other hand, in distributed algorithms, each node collects the information from its neighboring nodes and autonomously decides which link should be preserved [17-19]. Considering that the information each node obtains is limited, the final topology usually cannot achieve global optimization for large scale networks. Thus, it is important to develop a special topology control algorithm for the large scale SIN.

Considering the scale of SIN, a substantial proportion of the links are extremely long. Excessive use of these long-distance links data not only brings additional delay but also reduces the system efficiency [20, 21]. Thus in the SIN topology control, end-to-end delay should be considered more. Instead of using long links, nodes in SIN collaboratively determine which links should be used and define the network topology through forming proper neighbor relations. That is, topology control algorithms actually remove unnecessary links. As a result, the network topology is susceptible to unpredictable events such as hardware failures. Therefore, to design robust topology control algorithm, k-connectivity of the network is considered, where a k-connected network is k-1 fault-tolerant, i.e., the failure of less than k-1 nodes will not disconnect the whole network.

In this article, we study the topology control problem in SIN using a hierarchical autonomous system (AS) approach. An AS network is a collection of nodes have similar properties. The reasons for using hierarchical AS approach are twofold. Firstly, the complex SIN is decoupled into a series of AS networks, and each AS network could have different topology control strategy. Secondly, AS network is divided into small sub-AS networks to ensure strong connectivity after topology control. We propose an AS network topology control (AS-TC) algorithm using a hybrid approach. AS-TC preserves k-connectivity and is *min-max* delay optimal. A min-max algorithm tries to minimize the maximum end-to-end delay between any pair of nodes in the network. Briefly, the AS-TC algorithm consists of three phases: (i) Nodes in AS network of SIN autonomously form sub-AS networks and elect sub-AS cores. (ii) With the topology information gathered from the members of its sub-AS network, each sub-AS core minimize the maximum delay between any pair of nodes in the sub-AS network and guarantee strong connectivity using centralized method. (iii) Each sub-AS core selects a set of border nodes, shares topology information with neighboring sub-AS cores, and computes min-max delay links between neighboring sub-AS networks using distributed method. The main contributions of this article are summarized as follows:

1) An AS network model of SIN is proposed. The large scale, complex and dynamic SIN is decoupled into AS networks with similar nodes, which makes the control of the SIN easier.

2) An AS network topology control (AS-TC) algorithm is proposed to achieve low time delay and strong connectivity. It is a hybrid algorithm within a sub-AS network and among neighboring sub-AS networks.

3) The strong connectivity of AS-TC algorithm is proved that the algorithm could achieve logical *k*-connectivity on the condition that the original physical topology is *k*-connectivity.

The rest of this article is organized as follows. In Section 2, we define the network model and provide some definitions. In Section 3, we propose a hierarchical AS based algorithm AS-TC to achieve low time delay and strong connectivity. Then, the validity of AS-TC is proved in Section 4, and the message complexity of our algorithm is analyzed in Section 5. In Section 6, simulation results and discussion are presented. Finally, we make conclusion in Section 7.

2. Network Model

In this section, the network model of AS network is defined. As presented above, SIN is a new type of self-organizing system constituted by various kinds of nodes. As demonstrated in **Fig. 1**, the nodes distribute at different altitude, and work in different environment with either mobile (e.g., in HAPs networks) or static (e.g., in terrestrial wireless sensor networks) statuses. If we apply a unified strategy to manage the whole SIN, it will induce low efficiency, and even cannot maintain the normal operation of the network with too much control information. So, as shown in **Fig. 2**, we divide the SIN into a series of AS networks according to the property of the nodes. Each AS network can adopt an independent topology control strategy. In this way, the complex SIN is decoupled into sub-networks constituted by similar nodes, and the control of them is easier to carry out. AS-1 contains all the satellites of SIN and the topology of it is usually high dynamic but predictable. Topology control of AS-1 is usually based on satellites constellations design. We will not discuss the topology control of AS-1 in this paper. AS-2 contains the HAPs, AS-3 contains the airplanes and AS-4 contains the ground nodes. Properties of AS-2/3/4 are similar: nodes in them move slowly, and they connect to each other autonomously. In this article, we mainly focus on the topology control problem in AS-2/3/4.

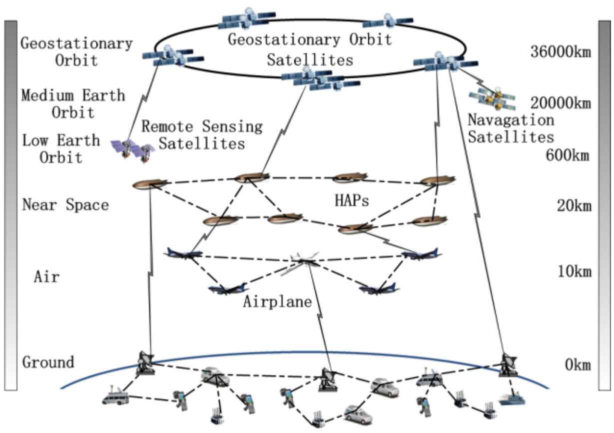


Fig. 1. The architecture of SIN.

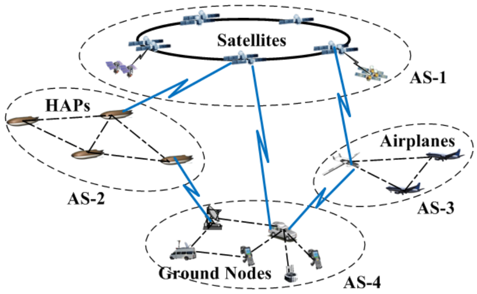


Fig. 2. The whole SIN is divided into a series of AS networks according to the property of the nodes.

Considering that the properties of nodes in an AS network are similar, we assume that nodes in the same AS network are homogeneous. They have the same maximal transmission range $R_{\max}$. Let an AS network topology be represented by an undirected simple graph $G=(V,E)$, where $V=\{u_1,u_2,\cdots,u_n\}$ is the set of nodes (or equivalently, vertices) and $E=\{(u_i,u_j)|(u_i,u_j\in V)\wedge(r(u_i,u_j)\le R_{\max})\}$ is the set of links (edges). $r(u_i,u_j)$ is the distance between nodes u_i and u_j. Each node is assigned a unique identifier (ID) according to its property, such as MAC address.

We assume that G is a general graph, that is, if $(u,v)\in E$, u and v can exchange information with each other. We also assume that the link is symmetric and obstacle-free, and each node is able to obtain its location information by some means (e.g., GPS). We then define several graphs related terms in the following, which will be used both in algorithms and proofs. For all definitions, we refer to graph $G=(V,E)$ and sub-graphs $G_i=(V_i,E_i)$ and $G_j=(V_j,E_j)$.

Definition 1: Weight Function. For an edge $e=(u,v)$, the weight function is $w(u,v)=(d(u,v),\min(ID(u),ID(v)),\max(ID(u),ID(v)))$, where $d(u,v)$ is the time delay between u and v when exchanging information, and $ID(\cdot)$ denotes the unique identifier of a node. Node's ID is included in the weight calculation to break ties. Given $(u_1,v_1),(u_2,v_2)\in E$, the relationship between $w(u_1,v_1)$ and $w(u_2,v_2)$ is given as

$$\begin{aligned}&w(u_1,v_1)>w(u_2,v_2)\Leftrightarrow d(u_1,v_1)>d(u_2,v_2)\\&or\big(d(u_1,v_1)=d(u_2,v_2)\big)\wedge\big(\min(ID(u_1),ID(v_1))>\min(ID(u_2),ID(v_2))\big)\\&or\big(d(u_1,v_1)=d(u_2,v_2)\big)\wedge\big(\min(ID(u_1),ID(v_1))=\min(ID(u_2),ID(v_2))\big)\wedge\\&\big(\max(ID(u_1),ID(v_1))>\max(ID(u_2),ID(v_2))\big)\end{aligned}. \quad (1)$$

It is obvious that edges with the same vertices have equivalent weights. However, edges with different end-vertices have different weights.

Definition 2: *k*-connected. In graph (topology) G, node u is said to be connected to node v, if there exists a path $p=\overline{ux_1x_2\cdots x_{m-1}x_mv}$, where $x_i\in V$ and $(u,x_1),(x_i,x_j),(x_m,v)\in E$. And for any $u,v\in V$, if there exists at least k disjoint paths between them. Graph G is *k*-connected, and denoted by $CON(G,k)$. If G is *k*-connected, it follows that, there does not exist a set of $k-1$ vertices, whose removal will partition G into two or more connected components.

Definition 3: Neighboring *k*-connected sub-graphs. For two disjoint sub-graphs G_i and G_j of G, if $\exists u \in V_i, v \in V_j$ and $\exists (u,v) \in E$, G_i and G_j are neighboring sub-graphs, denoted by $NBR_G(G_i,G_j)$. If $CON(G_i,k) \wedge CON(G_j,k)$, and $\exists (u_1,v_1),\cdots,(u_k,v_k) \in E$, where $u_1,\cdots,u_k \in V_i$ and $v_1,\cdots,v_k \in V_j$, G_i and G_j are neighboring *k*-connected sub-graphs, denoted by $NBR_G(G_i,G_j,k)$.

Definition 4: Multihop *k*-connected sub-graphs. Let $G_1,G_2,\cdots,G_n$ be a partitioning of G. If $\exists G_l$ subject to $NBR_G(G_i,G_l,k) \wedge NBR_G(G_l,G_j,k)$, G_i and G_j are multihop *k*-connected sub-graphs, denoted by $MCON_G(G_i,G_j,k)$.

3. Algorithms for Topology Control

Recall from the introduction, the design aims of the AS-TC algorithm in each AS are twofold: 1) to provide min-max delay optimal through a hierarchical AS approach, and 2) to achieve strong connectivity in the resulting network. The AS-TC algorithm does not require the global topology of the AS network to be known by any entity. On the contrary, AS-TC relies on sub-AS network where nodes autonomously form groups in AS, and select a core for each sub-AS network. It is a hybrid of centralized algorithm and distributed algorithm. A centralized topology control algorithm is applied to each sub-AS network to achieve the desired connectivity within the sub-AS, while the desired connectivity between adjacent sub-AS networks is achieved via localized information sharing between adjacent sub-AS cores. The following subsections detail the three phases of the AS-TC algorithm.

3.1 Phase 1: Sub-AS network Formation

The main function of *Phase* 1 is to select a minimal number of nodes as cores that dominate the sub-AS networks by using only 1-hop transmission. And these cores will take the main responsibility for the subsequent two phases.

Step 1: Broadcasting hello messages. When starting up, each node broadcasts *hello* messages periodically in order to let them discover each other in the surrounding area. A hello message is of the form $(NodeID, Location, CoreID, Degree, Delay)$. The explanation of each field is as follows. 1) $NodeID$: the unique ID of each node. 2) $Location$: the location of each node. 3) $CoreID$: the ID of the core with which the sending node is currently associated. If the sending node does not associate to any core, it is zero. Note that a core node uses its own ID for this field. 4) $Degree$: the degree of connectivity (the number of neighbors). 5) $Delay$: time delay to each neighbor when exchange information. It may contain processing, transmission and propagation delay in practice. In order to facilitate the analysis, we only consider propagation delay in this paper.

Step 2: Core selection process. The core selection process of each node begins after it has broadcasted hello messages for a certain waiting time. The waiting time should be long enough to allow this node receive at least one hello message from every immediate neighbor.

In this process, every node will decide whether it is suitable as a core of a sub-AS, or become a member of a sub-AS by checking for its local optimality. Each node computes its own *height* from its current states. The height metric should be chosen to suit the design goals of the AS topology control algorithm. Recall that the design goals of the AS-TC algorithm in each AS are twofold: 1) to provide min-max delay optimal through a hierarchical AS approach, and 2) to achieve strong connectivity in the resulting network. To achieve the first part of this goal, it is appropriate to choose the node with lower delay to each neighbor. On the other hand, to achieve the second part of the goal and to select a minimal number of nodes as cores, the nodes which have a higher degree should be considered. Therefore, we define the height metric as $\left(Delay, Degree, NodeID\right)$. $NodeID$ is included in the metric calculation to break ties. The height function is $height\left(u\right)=\left(h\left(u\right), ID\left(u\right)\right)$. In order to balance the factor of $Delay$ and $Degree$, we formulate $h\left(u\right)$ as

$$h\left(u\right)= f\left(Degree\left(u\right), Delay\left(u,v_i\right),\alpha\right). \tag{2}$$

In (2), $f\left(\cdot\right)$ denotes the balance function, v_i are the neighboring nodes of node u, $i=1,\cdots,Degree\left(u\right)$, and α denotes the balance factor. There can be many methods to formulate $f\left(\cdot\right)$, here we provide one of these methods for instance as

$$h\left(u\right)= Degree\left(u\right)+\alpha\cdot\overline{Delay_u^{-1}}. \tag{3}$$

In (3), $\overline{Delay_u^{-1}}=\frac{1}{Degree\left(u\right)}\sum_{i=1}^{Degree\left(u\right)}\frac{1}{Delay\left(u,v_i\right)}$ is the average of reciprocal delay with every neighboring node. We use reciprocal delay $\frac{1}{Delay\left(u,v_i\right)}$ here to guarantee that when the delay is lower and the degree is higher, we will obtain the higher $h\left(u\right)$. The value of α is choosed with the consideration of the following two aspects: 1) to determine which is more important between delay optimization and number of cores in the network, and 2) to make $Degree\left(u\right)$ and $\overline{Delay_u^{-1}}$ be of the same order of magnitude. The relationship between $hight\left(u\right)$ and $hight\left(v\right)$ is given by

$$\begin{aligned} &hight\left(u\right)>hight\left(v\right)\Leftrightarrow h\left(u\right)>h\left(v\right)\\ &\text{or }\left(h\left(u\right)=h\left(v\right)\right)\wedge\left(NodeID\left(u\right)>NodeID\left(v\right)\right). \end{aligned} \tag{4}$$

Then, if a node has the highest height among its neighbors, it is considered as a local optimal node and should serve as a core. After this process, the first batch of cores is selected, and all consequent hello messages will be changed accordingly. Note that, nodes will continue broadcasting hello messages at other steps to maintain the algorithm. The frequency of the *hello* messages can be divided into two types. Firstly, at each step, the hello message is broadcast with a fixed period, e.g. every one second. Secondly, when the status ($CoreID$ in the hello message) of a node changes, it will broadcast one hello message immediately.

Step 3: Supplement of cores. After Step 2, each node checks if there are cores in the range $R_{\max}$. If there exists, it will regard the core who has the least $Delay$ between them as its parent. That is, this node will be the member of the sub-AS dominated by its parent core. Then nodes update the $CoreID$ in their hello messages with their parent cores' ID. Note that, a core node uses its own ID for this field. However, each node only has the local information. After Step 2, some nodes may remain without parent. As shown in **Fig. 3**, the height values of node *A*, *B*, *C*, *D* and E are 2, 3.1, 3, 4 and 1 respectively. *A* is the highest height neighbor of *E*; *C* is the height neighbor of *A*; and *D* is the height neighbor of *C*. So after the first core selection step, node *A* and *E* remain without parent.

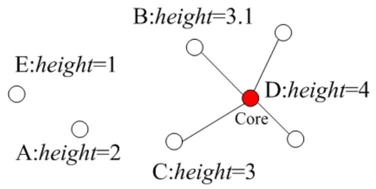


Fig. 3. Node A and E remain without parent.

Hence, supplement of new cores is needed. Nodes whose *CoreID* are zero （without parent） calculate their height functions (e.g., node *A* and E in **Fig. 3**). And the node who has the highest height among its neighbors without parent in the range $R_{\max}$ should serve as a core (e.g., node *A* in **Fig. 3**). If there are still nodes without parent, another cycle of supplement starts. After λ cycles of supplement, there are only two kinds of nodes, cores and members.

Step 4: Optimization and maintenance process. Considering nodes' mobility, and in order to keep the number of cores as low as possible, if a core detects there are other cores in the range $R_{\max}$ (from the hello process), it will check whether it has the highest height among these cores. If not, it will turn into a member of the highest height core, and its member nodes will turn into nodes without parent. If there exist nodes without parent in the AS, process will turn to Step 3. Finally, there are only two kinds of nodes, cores and members. And this optimization and maintenance process will keep monitoring the AS network. For instance, if a new node joint in the network, the process will take this node as a without parent node, and turn to Step 3.

3.2 Phase 2: Intra-sub-AS Topology Control

In this phase, we present a centralized algorithm for intra-sub-AS network. Each core will calculate the links for all of the members of its sub-AS such that the resulting topology of the sub-AS meets the given topology constraint (min-max delay and *k*-connectivity). The intra-sub-AS topology control algorithm is described in Algorithm 1, where G represents the AS, and let $G_1, G_2, \cdots, G_n$ (sub-AS) be a partitioning of G.

Algorithm 1 *：Intra-sub-AS Topology Control*

Input: (at sub-AS $G_s = (V_s, E_s)$)
  k (required connectivity)
Output:
  $G_k = (V_k, E_k)$
Begin:
  $V_k \leftarrow V_s$, $E_k \leftarrow \phi$
  Sort all edges in E_s in ascending order of *weight* (as defined in Definition 1)
  for all edge (u_i, v_i) in the order **do**
    if u_i is not *k*-connected to v_i **then**
    $E_k \leftarrow E_k \cup (u_i, v_i)$
    end if
  end for
Return G_k

For each sub-AS, Algorithm 1 ensures that G_k preserve the *k*-connectivity of G_s, i.e., $CON(G_s,k) \Rightarrow CON(G_k,k)$. And the maximum end to end delay among all edges in the sub-AS network is minimized by Algorithm 1, i.e., let $D_{\max}(G_k)$ be the maximum delay of all edges in the sub-AS minimized by Algorithm 1, and let $S_k(G_s)$ be the set of all kinds of *k*-connected sub-graphs of G_s, then we have $D_{\max}(G_k) = \min\{D_{\max}(G_i) | G_i \in S_k(G_s)\}$. The strong connectivity of Algorithm 1 is provided in section 4.

3.3 Phase 3: Inter-sub-AS Topology Control

In this phase, connectivity between adjacent sub-AS networks is considered. In order to allow adjacent sub-AS networks to discover each other, every node continue broadcasting *hello* message $(NodeID, Location, CoreID, Degree, Delay)$ as in *Phase* 1 periodically. When a node u receives a hello message from a node v that belongs to a different sub-AS (e.g., they have different $CoreID$), u will place v's information in its border list. Then this border list is reported to the node's parent core. With these border lists, we present a distributed algorithm for inter-sub-AS. This algorithm is described in Algorithm 2. Where G represents the AS, and let $G_1, G_2, \cdots, G_n$ (sub-AS) be a partitioning of G.

In this algorithm, the core of a sub-AS *A* checks whether there exist *k* disjoint links from this sub-AS to each adjacent sub-AS *B*. That is accomplished by applying an algorithm ($MaxMatching$) [22] that computes a matching of maximum cardinality in a *bipartite graph* defined by the nodes in respective sub-AS networks and the edges with one vertex in each sub-AS. If *k* does not exceed the size of maximum cardinality matching, the core of sub-AS *A* select *k* disjoint links that meet the min-max delay optimal. When there does not exist *k* disjoint links between *A* and *B* (only k_m disjoint links), the core preserve the k_m-connectivity between these two sub-AS networks and minimize the maximum delay between them. Note that this connectivity preservation (k_m-connectivity) cannot guarantee *k*-connectivity between sub-AS *A* and *B*. However, global *k*-connectivity can be guaranteed after *Phase* 3 is completed when connectivity with other neighboring sub-AS networks is already established. This will be proved in section 4.

Parameter $D_{IA}(G_1, G_2)$ in Algorithm 2 is used to perform an optimization which removes unnecessary links between certain adjacent sub-AS networks, while preserving the connectivity of the resulting topology. $D_{IA}(G_1, G_2)$ is the maximum delay of the selected *k* links. However, when the number k_m of disjoint links between two adjacent sub-AS networks is less than *k*, $D_{IA}(G_1, G_2)$ is ∞. Then, as shown in **Fig. 4**, sub-AS *A* will not connect to a neighboring sub-AS *B* directly if it observes that there exists another sub-AS *C*, where *C* is also a neighbor of *B*, and both $D_{IA}(G_A, G_C)$ and $D_{IA}(G_B, G_C)$ are less than $D_{IA}(G_A, G_B)$.

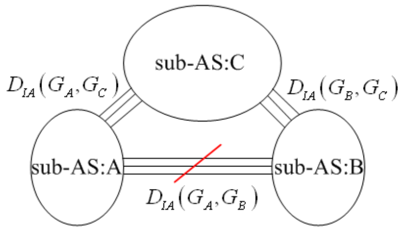


Fig. 4. Inter-sub-AS links between A and B are not necessary if there exist an neighboring sub-AS C such that both $D_{IA}(G_A, G_C)$ and $D_{IA}(G_B, G_C)$ are less than $D_{IA}(G_A, G_B)$.

***Algorithm 2** : Inter-sub-AS Topology Control*

Input: (at sub-AS $G_s = (V_s, E_s)$)
  k (required connectivity)
Output:
  Links for all nodes in G_s 's border list
Begin:
  $G_{si} = (V_{si}, E_{si}), V_{si} \leftarrow V_s, E_{si} \leftarrow \phi$
  for all G_i subject to $NBR_G(G_s, G_i)$ **do**
    $V' \leftarrow \{v | (v \in G_i) \wedge (v \text{ is adjacent to } G_s)\}$
    $V_{si} \leftarrow V_{si} \cup V'$
    $E_{si} \leftarrow \{(u,v) | (u \in V_s) \wedge (v \in V') \wedge (r(u,v) \le R_{\max})\}$
    $M \leftarrow \phi$
    $E_a \leftarrow$ sort all edges in E_{si} in ascending order of *weight* (as defined in Definition 1)
    $k_m \leftarrow |MaxMatching(G_{si})|$
    % $|MaxMatching(G_{si})|$ is the number of edges in $MaxMatching(G_{si})$
    if $k_m \ge k$ **then**
      for all edges $e_t = (u_t, v_t) \in E_a$ in the order **do**
        $M \leftarrow MaxMatching(G_t = (V_{si,} E_a(t)))$
        Find the smallest t, subject to $|M| \ge k$
        % $|M|$ is the number of edges in M , $E_a(t) = \{e_1, \cdots, e_t\}$
      end for
      $D_{IA}(G_s, G_i) \leftarrow |e_t|$, where $|e_t|$ is the weight of e_t
      $L(G_s, G_i) \leftarrow M$
    else
      for all edges $e_t = (u_t, v_t) \in E_a$ in the order **do**
        Find the smallest t, subject to $|M| \ge k_2$, and, and $M \leftarrow MaxMatching(G_t = (V_{si,} E_a(t)))$
      end for
      $D_{IA}(G_s, G_i) \leftarrow \infty$
      $L(G_s, G_i) \leftarrow M$
    end if
      Send $D_{IA}(G_s, G_i)$ to neighbor sub-AS
  end for
  Collect D_{IA} from neighboring sub-AS
  $LIST \leftarrow \phi$
  for all G_p subject to $NBR_G(G_s, G_p)$ **do**
      if there does not exist G_q subject to
      $NBR_G(G_s, G_q) \wedge NBR_G(G_s, G_p) \wedge (D_{IA}(G_s, G_q) < D_{IA}(G_s, G_p)) \wedge$
      $(D_{IA}(G_p, G_q) < D_{IA}(G_s, G_p))$
      then
        $LIST \leftarrow LIST \cup L(G_s, G_i)$
      end if
  end for
Return $LIST$

After *Phase* 3 is completed, each node is assigned a link list, and nodes connect each other according to these lists. This topology will be maintained by every node with *hello* message periodically, and always preserve the objective connectivity of the network.

4. Proof of Strong Connectivity

In this section, we prove that the strong connectivity of Algorithm 1 and Algorithm 2 [23]. The results are given as the following theorems.

4.1 Strong connectivity of Algorithm 1

Theorem 1. *Algorithm 1 can preserve k-connectivity of sub-AS* G_s *, i.e.,* $CON(G_s,k) \Rightarrow CON(G_k,k)$. *And the maximum delay among all nodes in the network is minimized by Algorithm 1.*

Before proving the correctness of Theorem 1, two lemmas are first provided. Let $p = \overline{ux_1x_2\cdots x_{m-1}x_m v}$ be the path from node u to v (as defined in Definition 2). Let the maximal set of disjoint paths from node u to v be represented by $P_{u,v}(G_s)$, i.e., $\forall p_m, p_n \in P_{u,v}(G_s)$, $p_m \cap p_n = \{u,v\}$. If edge $e_0 = (u,v)$, let $G_s - e_0$ be the resulting graph by removing the edge e_0 from G_s

Lemma 1. *Let* u *and* v *be two vertices in the k-connected graph* G_s*, if* u *and* v *are still k-connected after the removal of edge* $e_0 = (u,v)$*, then* $CON(G_s - e_0, k)$.

Proof of Lemma 1: In order to prove $CON(G_s - e_0, k)$, we prove $G_s' = G_s - e_0$ is connected with the removal of any $k-1$ vertices from G_s'. We already know that u and v are k-connected in G_s' . Thus, considering any two vertices $\{u_1, v_1\}$, we assume that $\{u_1,v_1\} \cap \{u,v\} = \phi$. We only need to prove u_1 is still connected to v_1 after the removal of a set $k-1$ vertices $X = \{x_1,\cdots,x_{k-1}\}$, where $x_i \in (V(G_s') - \{u_1,v_1\})$. If (u_1,v_1) is an edge in G_s', that is obviously true. Hence, we only consider the case that there is no directly edge from u_1 to v_1.

Since $CON(G_s,k)$, we have $|P_{u_1,v_1}(G_s)| \ge k$, where $|P_{u_1,v_1}(G_s)|$ is the number of paths in the set $P_{u_1,v_1}(G_s)$. Let r_1 be the number of paths in $P_{u_1,v_1}(G_s')$ that are broken after the removal of vertices in the set of X, i.e., $r_1 = \{p \in P_{u_1,v_1}(G_s') | (x_i \in X) \wedge (x_i \in p)\}$. We know that paths in $P_{u_1,v_1}(G_s')$ are disjoint, so the removal of any one vertex in X can only break at most one path in $P_{u_1,v_1}(G_s')$. Given $|X| = k-1$, we have $r_1 \le k-1$.

Let G_s'' be the resulting graph by removing X from G_s'. If $|P_{u_1,v_1}(G_s')| \ge k$, we have $|P_{u_1,v_1}(G_s'')| \ge (|P_{u_1,v_1}(G_s')| - r_1) \ge 1$, i.e., u_1 is still connected to v_1 in G_s'' . Otherwise, $|P_{u_1,v_1}(G_s')| < k$, it occurs only if the removal of edge $e_0 = (u,v)$ breaks one path

$p_j \in P_{u_1,v_1}(G_s)$. Without loss of generality, let the order of vertices in the path p_j be $\overline{u_1,\cdots,u,v,\cdots,v_1}$. Since the paths in $P_{u_1,v_1}(G_s)$ are disjoint, the removal of edge e_0 breaks at most one path, i.e., $\left|P_{u_1,v_1}(G_s)-\{p_j\}\right| \geq k-1$. So we have $\left|P_{u_1,v_1}(G'_s)\right| = k-1$.

If $r_1 < k-1$, it is obvious that $\left(\left|P_{u_1,v_1}(G'_s)\right| - r_1\right) \geq 1$. Hence $\left|P_{u_1,v_1}(G''_s)\right| \geq 1$. That is, u_1 is still connected to v_1 in G''_s. Otherwise, if $r_1 = k-1$, every vertex in the set X belongs to the paths in $P_{u_1,v_1}(G'_s)$. We know that $p_j \in P_{u_1,v_1}(G_s)$ is disjoint with the paths in $P_{u_1,v_1}(G'_s)$, so we have $p_j \cap X = \phi$. Hence, no vertex in $\overline{u_1,\cdots,u,v,\cdots,v_1}$ is removed with the removal of X. So, with the removal of e_0, u_1 is still connected to u, and v is still connected to v_1 in G''_s. With the assumption that u and v are still *k*-connected after the removal of edge $e_0 = (u,v)$ in Lemma 1. It is obvious that u is still connected to v in G''_s. So u_1 is still connected to v_1 in G''_s.

We have proved that for any two vertices $\{u_1,v_1\} \in G'_s$, u_1 is connected to v_1 with the removal of any $k-1$ vertices from $V(G'_s)-\{u_1,v_1\}$. Hence, $CON(G'_s,k)$.

Lemma 2. *Let G_s and $\hat{G}_s$ be two graphs, where $CON(G_s,k)$ and $V(G_s)=V(\hat{G}_s)$. If every edge subject to $(u,v) \in \left(E(G_s)-E(\hat{G}_s)\right)$ satisfies that u is still k-connected to v in graph $G_s - \{(u',v') \in E(G_s) \mid w(u',v') \geq w(u,v)\}$, then $CON(\hat{G}_s,k)$.*

Proof of Lemma 2: Without loss of generality, let $\{e_1,e_2,\cdots,e_m\} = E(G_s)-E(\hat{G}_s) = \{(u_1,v_1),(u_2,v_2),\cdots,(u_m,v_m)\}$ be a set of edges subject to $w(e_1) > w(e_2) > \cdots > w(e_m)$. We define a series of sub-graphs of G_s: $G_s^0 = G_s$, and $G_s^i = G_s^{i-1} - e_i$, where $i = 1,2,\cdots,m$. Then we have $G_s^m = G_s - \{e_1,e_2,\cdots,e_m\}$, and $E(G_s^m) \subseteq E(\hat{G}_s)$. Here we prove Lemma 2 by induction.

Base: Obviously, we have $G_s^0 = G_s$, and $CON(G_s^0,k)$.

Induction: If $CON(G_s^{i-1},k)$, we prove that $CON(G_s^i,k)$, where $i = 1,2,\cdots,m$. Since $G_s - \{(u',v') \in E(G_s) \mid w(u',v') \geq w(u_i,v_i)\} \subseteq G_s^{i-1} - (u_i,v_i)$, and from the assumption of Lemma 2 (u_i is *k*-connected to v_i in graph $G_s - \{(u',v') \in E(G_s) \mid w(u',v') \geq w(u_i,v_i)\}$), we obtain that u_i is *k*-connected to v_i in graph $G_s^{i-1} - (u_i,v_i)$. Applying Lemma 1 to G_s^{i-1}, it is obvious $CON(G_s^{i-1} - (u_i,v_i),k)$. That is, $CON(G_s^i,k)$.

By induction, we have $CON(G_s^m,k)$. Since $E(G_s^m) \subseteq E(\hat{G}_s)$, hence $CON(\hat{G}_s,k)$.

Finally, we prove the correctness of Theorem 1.

Proof of Theorem 1: In Algorithm 1, we sort all edges in $G_s = (V_s,E_s)$ in ascending order of *weight*. Whether $(u,v) \in E_s$ should be placed into G_k depending on the connection of u and

v, and edges of smaller weights. $\forall(u_0,v_0)\in E(G_s)-E(G_k)$, the edge (u_0,v_0) is not placed into G_k, but u_0 is k-connected to v_0. Obviously, any edge (u',v') subject to $((u',v')\in E(G_s))\wedge(w(u',v')\geq w(u_0,v_0))$ is not needed to be placed into G_k for the current node u_0 and v_0. Hence, for every edge $(u,v)\in E(G_s)-E(G_k)$, we have u is *k*-connected to v in $G_s-\{(u',v')\in E(G_s)\,|\,w(u',v')\geq w(u,v)\}$. Applying Lemma 2 here, then we can prove that $CON(G_s,k)\Rightarrow CON(G_k,k)$.

Recall that $D_{\max}(G_k)$ is the maximum delay of all edges in the sub-AS minimized by Algorithm 1, and $S_k(G_s)$ is the set of all kinds of *k*-connected sub-graphs of G_s. The maximum delay among all edges in the network is minimized by Algorithm 1 can be described as $D_{\max}(G_k)=\min\{D_{\max}(G_i)\,|\,G_i\in S_k(G_s)\}$.

Let (u_m,v_m) be the last edge that is placed into G_k, i.e., $w(u_m,v_m)=\max_{(u,v)\in E(G_k)}\{w(u,v)\}$. Let $G'_k=G_k-(u_m,v_m)$, then we obtain that $|P_{u_m,v_m}(G'_k)|<k$. Now, we assume that there is a graph $H_s=(V(H_s),E(H_s))$, where $V(H_s)=V(G_s)$, and $E(H_s)=\{(u,v)\in E(G_s)\,|\,w(u,v)<w(u_m,v_m)\}$. If we can prove that $CON(H_s,k)$ is not true, we will obtain that any $G_i\in S_k(G_s)$ should have at least one edge equal to or heavier than (u_m,v_m). That is, $D_{\max}(G_k)=\min\{D_{\max}(G_i)\,|\,G_i\in S_k(G_s)\}$. We prove $CON(H_s,k)$ is not true by contradiction in the following.

Assume that $CON(H_s,k)$, hence, $|P_{u_m,v_m}(H_s)|\geq k$. We have $H_s-G'_k\neq\phi$. Since all edges are placed into G'_k in the ascending order, $\forall(u,v)\in H_s-G'_k$ should satisfy that u is *k*-connected to v in $H_s-\{(u',v')\in E(H_s)\,|\,w(u',v')\geq w(u,v)\}$. Applying Lemma 2 here, we obtain that $CON(G'_k,k)$. That is, $|P_{u_m,v_m}(G'_k)|\geq k$, which is a contradiction.

4.2 Strong connectivity of Algorithm 2

Theorem 2. *Let* $G=(V,E)$ *be the initial topology of an AS. Let* $G'=(V,E')$ *be the topology after Algorithm 2 is completed. Then we have* $CON(G,k)\Leftrightarrow CON(G',k)$.

Before proving the correctness of Theorem 1, several lemmas used in that proof are first provided.

Lemma 3. *Let* $G_i=(V_i,E_i)$ *and* $G_j=(V_j,E_j)$ *be two sub-graphs of graph* G. *If* $NBR_G(G_i,G_j,k)$, *then* $CON(G_i\cup_G G_j,k)$.

Proof of Lemma 3: In order to prove $CON(G_i\cup_G G_j,k)$, we prove $G_i\cup_G G_j$ is connected with the removal of any *k*-1 vertices from it. Since $NBR_G(G_i,G_j,k)$, we have $CON(G_i,k)$

and $CON\left(G_j,k\right)$, i.e., consider any $u,v\in G_i$ or $u,v\in G_j$, u is k-connected to v. Then we only need to consider the case $\left(u\in G_i\right)\wedge\left(v\in G_j\right)$.

Since $NBR_G\left(G_i,G_j,k\right)$, $\exists u_0\in G_i,v_0\in G_j$, u_0 is connected to v_0 with the removal of any k-1 vertices from $V_i\cup V_j-\{u_0,v_0\}$. With $CON\left(G_i,k\right)$ and $CON\left(G_j,k\right)$, we know that u is connected to u_0, and v is connected to v_0. Hence u is connected to v. That is, $G_i\cup_G G_j$ is connected with the removal of any k-1 vertices from it.

Corollary 1. *Let sub-graphs $G_1,G_2,\cdots,G_n$ be a partitioning of G. Let S_m be the maximal set of sub-graphs subject to $\forall G_i,G_j\in S_m$, $\exists MCON_G\left(G_i,G_j,k\right)$. Then, $\cup_G\left\{G_i\middle|G_i\in S_m\right\}$ is k-connected.*

Lemma 4. *Let G_s be a sub-graph of G, and let G_s' be edges reduction of G_s. Let $G''=\left(V,E'\right)=\left(G-G_s\right)\cup_G G_s'$. If $CON\left(G_s,k\right)\wedge CON\left(G_s',k\right)\wedge CON\left(G,k\right)$, then $CON\left(G'',k\right)$.*

Proof of Lemma 4: In order to prove $CON\left(G'',k\right)$, we prove $\forall u,v\in G''$ is connected with the removal of any k-1 vertices from G''. Without loss of generality, three cases are considered in the following.

1) $u,v\in V_s$: It is obvious true because of $CON\left(G_s',k\right)$.

2) $u\in V_s$ and $v\in V-V_s$: Since $CON\left(G,k\right)$, u is connected to v in path p with the removal of any k-1 vertices in G. If $p\subseteq E-E_s$, p is also exist in G'', u is connected to v by removing those k-1 vertices. Otherwise, $\exists\left(a\in p\right)\wedge\left(a\in V_s\right)$ and a is connected to v in $G-G_s$. Since $CON\left(G_s',k\right)$, u is connected to a by removing those k-1 vertices. Then u is connected to v with the removal of any k-1 vertices in G''.

3) $u,v\in V-V_s$: Similarly, since $CON\left(G,k\right)$, u is connected to v in path p with the removal of any k-1 vertices in G. If $p\subseteq E-E_s$, u is k-connected to v in G''. Otherwise, $\exists\left(a_1,a_2\in p\right)\wedge\left(a_1,a_2\in V_s\right)$, u is connected to a_1 and a_2 is connected to v in $G-G_s$. Since $CON\left(G_s',k\right)$, a_1 is connected to a_2 by removing those k-1 vertices. Then u is connected to v with the removal of any k-1 vertices in G''.

Corollary 2. *Let $G_1,G_2,\cdots,G_n$ be k-connected sub-graphs of k-connected graph G. Let $G_1',G_2',\cdots,G_n'$ be edges reduction of $G_1,G_2,\cdots,G_n$, and $G_1',G_2',\cdots,G_n'$ are k-connected. Then, $G''=\left(G-\cup_{G\,i=1}^{n}G_i\right)\cup_G\left(\cup_{G\,i=1}^{n}G_i'\right)$ is k-connected.*

Lemma 5. *Let $G=\left(V,E\right)$ be the initial topology of an AS. Let $G'=\left(V,E'\right)$ be the topology after Algorithm 2 is completed. Let $G_i=\left(V_i,E_i\right)$ be the sub-AS networks resulting from Phase 1 in the topology control, where $i=1,\cdots,n$. Let $G_i'=\left(V_i,E_i'\right)$, where $E_i'=E_i\cap E'$. Then, $\forall i,j$ subject to $1\le i\le j\le n$, we have $MCON_G\left(G_i,G_j,k\right)\Rightarrow MCON_{G'}\left(G_i',G_j',k\right)$.*

Proof of Lemma 5: Since nodes of any intra-sub-AS are *k*-connected. We take sub-AS as a *node* here. Formally, we represent graph G as $\bar{G}=(V_S,E_S)$, where $V_S=\{G_1,G_2,\cdots,G_n\}$ and $E_S=\{(G_i,G_j)|NBR_G(G_i,G_j,k)\}$. Actually, edge (G_i,G_j) contains at least *k* disjoint paths between G_i and G_j. Let $\bar{G}'=(V_S,E'_S)$ be the sub-AS level representation of G', where $E'_S=\{(G'_i,G'_j)|NBR_{G'}(G'_i,G'_j,k)\}$. We use V_S to represent the set of sub-AS networks in $\bar{G}'$, because we need not to consider the topology of intra-sub-AS (both G_i and G'_i are *k*-connected). We take all of them as nodes, so we consider (G_i,G_j) and (G'_i,G'_j) as the same edge. Recall that in Algorithm 2, each edge $(G_i,G_j)\in E_S$ has a weight $D_{IA}(G_i,G_j)$.

In order to prove Lemma 5, it suffices to show that $\forall G_i,G_j\in\bar{G}$, G_i is connected to G_j in $\bar{G}'$. We order all edges in $\bar{G}$ in the ascending sequence of weights, and then judge whether an edge should be place into $\bar{G}'$. Without loss of generality, let the ordering be $(e_1,e_2,\cdots,e_m)$, where $m=|E_S|$. Then we prove Lemma 5 by induction.

Base: Obviously, the pair of sub-AS networks corresponding to edge e_1 should always be placed into $\bar{G}'$, i.e., $e_1\in E'_S$.

Induction: $\forall t\le m$, if for all $q<t$, the pair of sub-AS networks corresponding to e_q are connected in $\bar{G}'$ (either directly or indirectly). And suppose $e_t=(G_i,G_j)$. From Algorithm 2, the only reason why $e_t\notin E'_S$ (G_i is not directly connected to G_j in $\bar{G}'$) is that there exists another sub-AS G_l, where both $D_{IA}(G_i,G_l)$ and $D_{IA}(G_l,G_j)$ is less than $D_{IA}(G_i,G_j)$. However, since edges (G_i,G_l) and (G_l,G_j) come before (G_i,G_j) in the ascending order. From path $\overline{G_iG_lG_j}$, G_i is connected to G_j in $\bar{G}'$.

By induction, we prove that G_i is connected to G_j in $\bar{G}'$, and then $MCON_G(G_i,G_j,k)\Rightarrow MCON_{G'}(G'_i,G'_j,k)$.

Finally, we prove the correctness of Theorem 2. In the proof, G_i and G'_i have the same definition in Lemma 5.

Proof of Theorem 1: For every sub-AS G_i, we know that $CON(G_i,k)$ is true after Algorithm 1. Then we partition those sub-AS networks into sets $A_1,\cdots,A_s$, where each set contains sub-AS networks are multihop *k*-connected in G, i.e., $\forall r=1,\cdots,s$, then, $(G_i\in A_r)\wedge(MCON_G(G_i,G_j,k))\Rightarrow G_j\in A_r$. Then we define sets $A'_1,\cdots,A'_s$, where $\forall i,G_i\in A_r\Rightarrow G'_i\in A'_r$. Applying Lemma 5 here, for every $A'_r=\{G'_{r_1},\cdots,G'_{r_m}\}$, $\forall 1\le i<j\le m$, we have $MCON_{G'}(G'_{r_i},G'_{r_j},k)$. Take A'_r as a sub-graph of G', $A'_r=(V_{A'_r},E_{A'_r})$, where $V_{A'_r}=\{v|v\in A'_r\}$ and $E_{A'_r}=\{(u,v)|(u,v\in A'_r)\wedge((u,v)\in E')\}$. Since A'_r only contains multihop *k*-connected sub-graphs, applying Corollary 1 here, we have A'_r is *k*-connected. Then,

applying Corollary 2 here, we have $G'=\left(G-\left(\cup_{Gr=1}^{s} A_r\right)\right)\cup_G\left(\cup_{Gr=1}^{s} A'_r\right)$ is k-connected. Then $CON(G,k)\Leftrightarrow CON(G',k)$.

5. Control Message Complexity Analysis

We study the control message complexity here by computing the total number of control messages exchanged during the three phases of the AS-TC algorithm. The following terms are used in the complexity analysis.

Let N be the total number of nodes in the AS network. Let S be the number of sub-AS networks, and N_S be the average number of nodes per sub-AS, i.e., $N_S = N/S$. Let R_B be the average probability of nodes that are border nodes in a sub-AS, where $0 < R_B < 1$. Let S_N be the average number of neighboring sub-AS networks for each sub-AS, i.e., $0 < S_N < S$. Let λ be the cycles of cores supplement.

Table 1 shows the average control messages utilized in each phase to complete the topology algorithm for each sub-AS. We partition each phase into major steps. Hence, from **Table 1**, the total number of control messages required in the AS is $S\left(\left(2+\lambda+R_B\right)N_S+2S_N+2\right)$. It can be simplified as $\left(2+\lambda+R_B\right)N+2S_NS+2S$, which is $O(N)+O(S_NS)$ in the worst case.

Table 1. Average message complexity in each phase of a sub-AS

Steps in each phase	Number of control messages
***Phase*1:**	
Nodes announce their existence.	N_s
First batch of cores are selected in AS network using these announcements.	0
All cores in AS network is selected with λ cycles of supplement.	λN_s
Each node announces its current role.	N_s
***Phase* 2:**	
Core node computes the intra-sub-AS topology.	0
***Phase* 3:**	
All border nodes report their border lists to the parent core.	$N_s \cdot R_B$
Core node distribute D_{IA} vector to its border nodes.	1
Border nodes send D_{IA} vector to border nodes of other sub-AS.	S_N
Border nodes of other sub-AS report D_{IA} vector to their parent core.	S_N
Core node sends the link list to the sub-AS members.	1

6. Simulation Results and Discussions

In this section, we present several sets of simulation results to evaluate the effectiveness of the proposed AS-TC algorithm. Recall that the proposed algorithm is a hybrid of centralized algorithm and distributed algorithm. We compare it with typical centralized algorithm FGSS_k [17] and distributed algorithm FLSS_k [17]. We chose these two algorithms because they are also min-max optimal as our algorithm. These simulations were carried out using the NS2 simulator.

In this simulation study, the wireless channel is symmetric (i.e., both the sender and the receiver should observe the same channel fading) and obstacle-free, and each node has an equal maximal transmission range R_{max} =350 km. Nodes are randomly distributed in a $2000 \times 2000 km^2$ region. In order to study the effect of sub-AS size on the resulting topologies, we vary the number of nodes in the region among 125, 150, 175, 200, 225, 250.

For each network, we consider:

1) *k*-connectivity: *k*=1, *k*=2 and *k*=4;

2) algorithms: the proposed hybrid algorithm AS-TC, centralized algorithm $FGSS_k$ and distributed algorithm $FLSS_k$;

3) 1000 Monte Carlo simulations.

Relative to AS-TC, recall that in *Phase* 1 of sub-AS network formation, we configure that each node is at most one hop away from its parent core. In our simulations, algorithm in *Phase* 1 generates sub-AS networks where the average number of nodes per sub-AS is 6.68, 7.67, 8.64, 9.69 and 10.15 (results of 1000 simulations), respectively. Note that, by varying the number of nodes in the network while maintaining other parameters such as the region size and maximal transmission range of nodes, we implicitly adjust the node degree of these topology control algorithms.

Before providing the experimental results regarding time delay, we first observe the actual topologies for one simulated network using AS-TC algorithm. Four figures are given here.

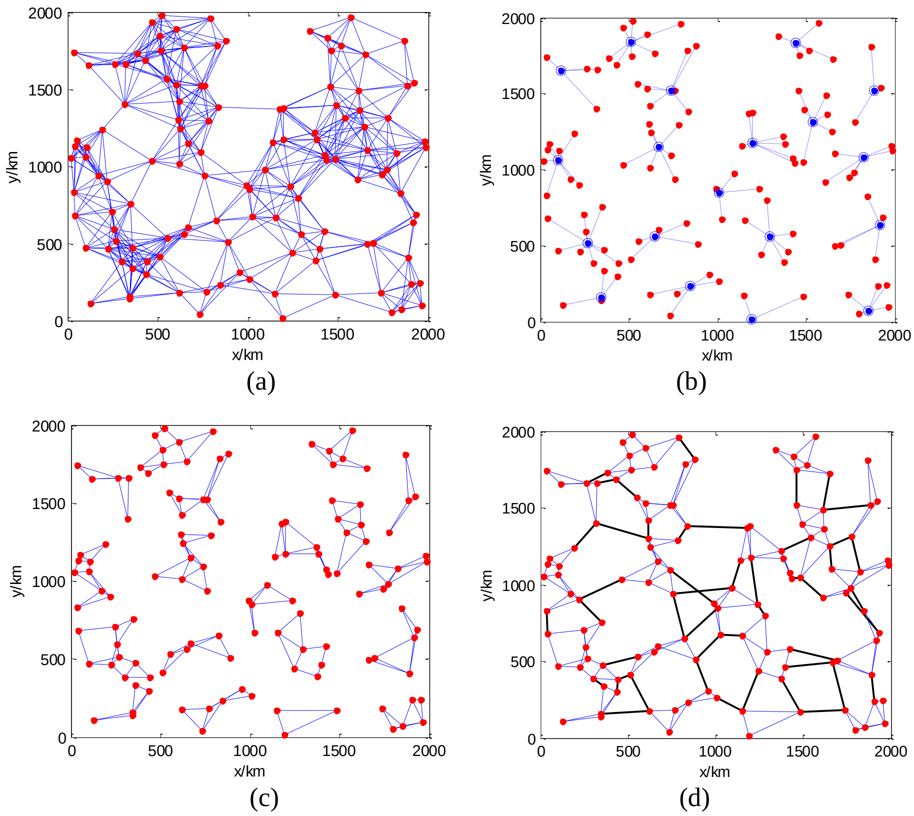


Fig. 5. Network topologies of 125 nodes with different topology control settings. (a) Without topology control. (b) After applying algorithm of *Phase* 1. (c) $k=2$, after applying algorithm of *Phase* 2. (d) $k=2$, after applying algorithm of *Phase* 3.

1) **Fig. 5(a)** shows the original physical topology without topology control. All nodes communicate with the maximal transmission range $R_{\max}$.

2) **Fig. 5(b)** shows the topology after applying algorithm of *Phase* 1. Nodes of the AS are divided into 19 sub-AS networks, where the average number of nodes per sub-AS is 6.58.

3) **Fig. 5(c)** is the topology resulting from the intra-sub-AS topology control algorithm of *Phase* 2, when $k = 2$.

4) **Fig. 5(d)** shows the topology after applying inter-sub-AS Topology control algorithm of *Phase* 3, when $k = 2$. The inter-sub-AS links are represented by black color.

In **Fig. 6**, we show average and maximum delay between two nodes which obtained from three topology control algorithm (the proposed hybrid algorithm AS-TC, centralized algorithm FGSS_k [17] and distributed algorithm FLSS_k [17]. Note that, we only consider link propagation delay in the simulation. It is evident from those results that AS-TC is very effective in reducing the delay between nodes. Recall that the maximal transmission range $R_{\max}$ of one node is 350 km. The corresponding delay is 1.167 ms. When $k = 1$ (**Fig. 6(a)**), AS-TC reduces the maximum delay to 0.864 ms when there are 125 nodes in the AS and as low as 0.577 ms when there are 225 nodes. The maximum delay is approximately 9.5% to 18.1% lower than FLSS_1 distributed algorithm, and 10.9% to 18.9% higher than FGSS_1 centralized algorithm. For the average delay, AS-TC reduces the delay to 0.524 ms when there are 125 nodes in the AS and as low as 0.354 ms when there are 225 nodes, which is approximately 9.0% to 12.1% lower than FLSS_1 distributed algorithm, and 10.8% to 13.2% higher than FGSS_1 centralized algorithm.

When $k = 2$ (**Fig. 6(b)**), both the maximum and average delay resulting from AS-TC, FGSS_2 and FLSS_2 are all higher than when $k = 1$. That is expected because 2-connected is a stronger property than 1-connected. What's more, the difference among the three algorithms when $k = 2$ is in a greater range than when $k = 1$. This is the consequence of having to maintain another higher delay link between adjacent sub-AS networks and one more additional disjoint path from each node to other nodes within all sub-AS networks. The maximum delay of AS-TC is approximately 12.9% to 19.8% lower than FLSS_2 distributed algorithm, and 14.9% to 20.8% higher than FGSS_2 centralized algorithm. The average delay of AS-TC is approximately 12.9% to 16.6% lower than FLSS_2 distributed algorithm, and 10.7% to 18.4% higher than FGSS_2 centralized algorithm.

When $k = 4$ (**Fig. 6(c)**), as expected, both the maximum and average delay resulting from the three algorithms are all higher than when $k = 1$ and $k = 2$. However, the differences among the three algorithms are in a small range when there is little number of nodes in the network, e.g., when number of nodes is 125. The main causes of this phenomenon are the following two aspects: 1) The maximal transmission range $R_{\max}$ of each node is 350 km, and the corresponding delay is 1.167 ms (we only consider link propagation delay in the simulation). In order to achieve stronger connectivity ($k = 4$), more links are preserved, but all the links' delay is less than 1.167 ms. This upper bound of link delay make the performance differences among the three algorithms small when most links of the original topology should be preserved. 2) When the density of nodes in the network is relative low, the connectivity of the original network topology (without topology control) may not achieve $k = 4$, i.e., only k'-connectivity, $k' < 4$. In this case, the three algorithms not only do their best to preserve global k'-connectivity but also guarantee local or partly k-connectivity that the original network can achieve. So, a lot of long delay links are preserved. And considering the upper bound of link delay (1.167 ms), the performance differences among the three algorithms become smaller. The maximum delay of AS-TC is approximately 1.3% to 16.0% lower than $FLSS_4$ distributed algorithm, and 3.3% to 12.7% higher than $FGSS_4$ centralized algorithm. The average delay of AS-TC is approximately 5.3% to 12.3% lower than $FLSS_4$ distributed algorithm, and 7.1% to 13.3% higher than $FGSS_4$ centralized algorithm.

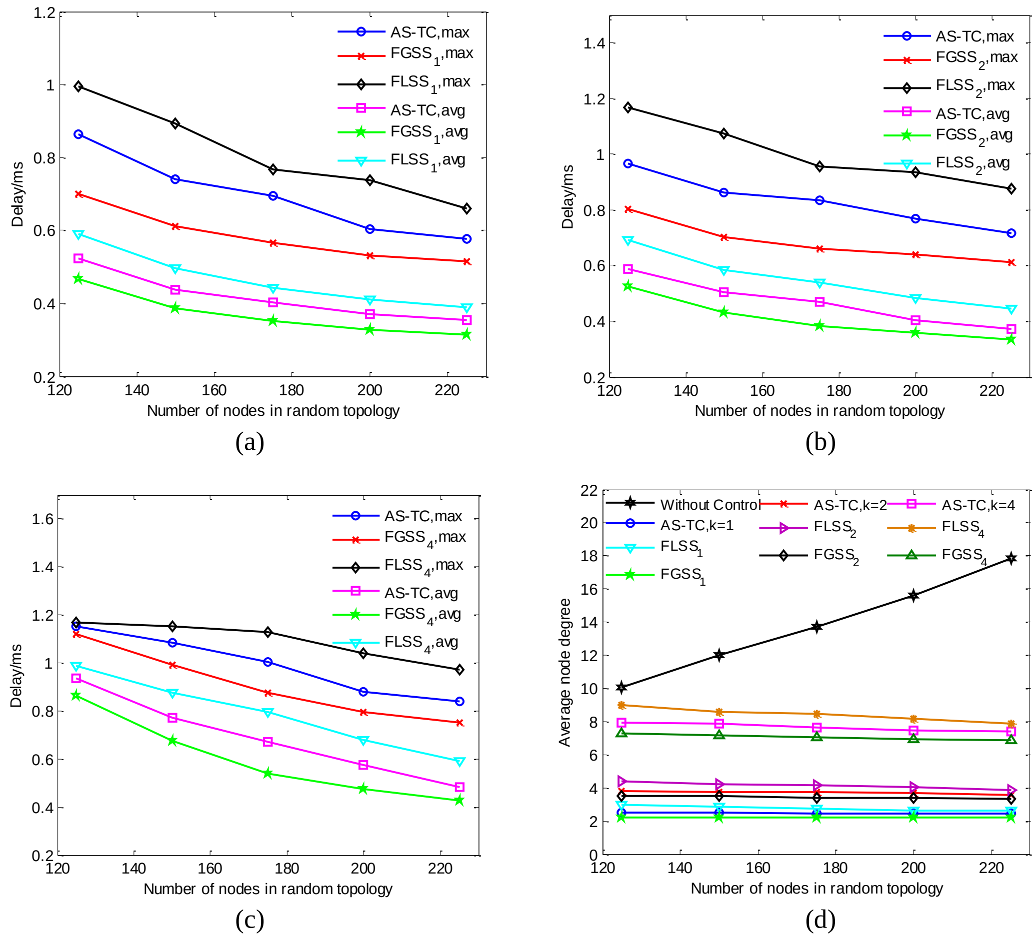


Fig. 6. Results from three topology control algorithms (AS-TC, $FGSS_k$ and $FLSS_k$) showing average and maximum link delay when (a) $k=1$, (b) $k=2$, (c) $k=4$ and (d) average node degree.

The delay performance of the proposed algorithm AS-TC falls in between $FGSS_k$ and $FLSS_k$. This is expected because AS-TC is a hybrid of centralized algorithm and distributed algorithm. Even though centralized algorithm has better delay performance (only 7.1% to 18.4%), they are not suitable for large scale networks. Because excessive amounts of control messages need to be collected by one central entity, and long delay makes the control messages exchanged with remote nodes costly. However, the control message exchange in AS-TC is constrained among neighboring sub-AS networks, and the delay performance is better than distributed algorithm in the simulation result. Thus, the proposed AS-TC algorithm is better than centralized algorithm and distributed algorithm for SIN.

Fig. 6(d) shows the average node degree of the topologies derived under AS-TC, FGSS_k and FLSS_k for k = 1, 2, and 4. The average degree without topology control increases almost linearly with the number of nodes. In contrast, the average degree under all the three algorithms actually slightly decreases. It is obvious that the node degree of a network with AS-TC does not depend on the size or density of the network. What's more, when with the same k, the average degree under AS-TC falls in between FGSS_k and FLSS_k. This is because AS-TC is a hybrid of centralized algorithm and distributed algorithm, and the centralized algorithm usually has the best performance.

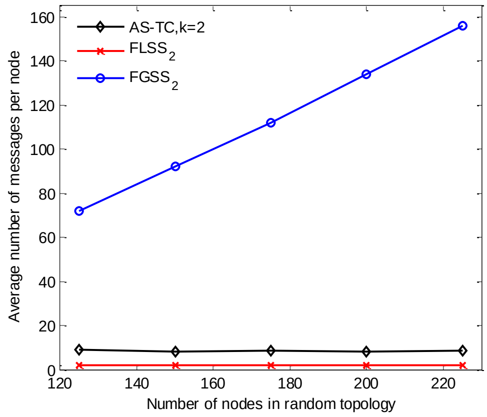


Fig. 7. Number of messages exchanges per node derived under AS-TC, FGSS_k, FGSS_k for $k = 2$.

Fig. 7 illustrates the number of messages exchanges required per node to complete AS-TC, FGSS_k and FLSS_k. Each node in FLSS_k only exchanges control messages with its visible neighborhoods ($R_{\max} = 350$ km), so FLSS_k has the lowest message complexity. Recall that the message complexity of the AS-TC algorithm is $O(N) + O(S_N S)$. For each node, the average number of messages required is $\left(O(N) + O(S_N S)\right)/N = O(1)$. The result validates the analysis. When number of nodes in the AS increases from 125 to 225, the average number of messages required per node in AS-TC does not increase. This shows the AS-TC algorithm has little extra overhead. In contrast, each node in FGSS_k algorithm sends a message to announce its existence to all other nodes in the network. Since no node has global knowledge of the network at the beginning, the message distribution in the simulation is done by flooding. Thus, a message from one node may travel every link of the network. Therefore, the total number of messages is $O\left(N^2\right)$ in the worst case. For each node, the average number of messages required is $O\left(N^2\right)/N = O(N)$. When number of nodes in the AS increases, the average number of messages required per node in FGSS_k also increases.

7. Conclusion

We studied the topology control problem in the SIN using a hierarchical AS approach. The motivation was that the AS network model decouples the complex SIN into simple sub-AS networks, and make the topology control of the SIN easier to carry out. Then we proposed the AS-TC algorithm to minimize time delay in the SIN. Compared with most existing approaches for SIN where either the purely centralized or the purely distributed control method is adopted, AS-TC utilizes a hybrid method. In this way, not only the control message exchange is constrained among local neighboring sub-AS networks, but also the strong connectivity of the network is preserved. Our simulation results validated the theoretic analysis and effectiveness of the AS-TC algorithm.

Although the assumptions stated in Section 2 are widely used in existing topology algorithms, some of them may not be practical. Our future work will focus on how to relax these constraints (e.g. nodes in one AS are homogeneous, and they have the same R_{max}) for AS-TC algorithm so as to improve its practicality in real applications. Furthermore, we will also consider the topology control of AS-1 (constituted by satellites) in the near future.

References

[1] J. Mukherjee and B. Ramamurthy, "Communication Technologies and Architectures for Space Network and Interplanetary Internet," *IEEE Communication Surveys and Tutorials*, vol. 15, no. 2, pp. 881-897, 2013. Article (CrossRef Link)

[2] K. B. Bhasin and J. K. Hayden, "Architecting Communication Network of Networks for Space System of Systems," in *Proc. of the IEEE System of Systems Engineering Conference*, pp. 1-7, 2008. Article (CrossRef Link)

[3] H.-F. Hu and Y.-A. Liu, "A Feasible Mesh-based Architecture and Protocol Model of Space Information Network," in *Proc. of the IEEE Geoscience and Remote Sensing Conference*, pp. 529-531, 2010. Article (CrossRef Link)

[4] F. Ren and J.-L. Fan, "An Adaptive Distributed Certificate Management Scheme for Space Information Network," *IET Information Security*, vol. 7, no. 4, pp. 318-326, 2013. Article (CrossRef Link)

[5] G.-X. Zhang, W. Zhang, H. Zhang and Z.-D. Xie, "A Novel Proposal of Architecture and Network Model for Space Communication Network," in *Proc. of the IAF 65th International Astronautical Congress*, pp. 1-7, 2014. https://iafastro.directory/iac/archive/browse/IAC-14/B2/3/23563/

[6] B.-Y. Guo, Q.-S. Guan, F. R. Yu, S.-M. Jiang and V. C. M. Leung, "Energy-Efficient Topology Control with Selective Diversity in Cooperative Wireless Ad Hoc Networks: A Game-Theoretic Approach," *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6484-6495, 2014. Article (CrossRef Link)

[7] X. Ao, R. Yu, S.-M. Jiang, Q.-S. Guan and V. C. M. Leung, "Distributed Cooperative Topology Control for WANETs with Opportunistic Interference Cancelation," *IEEE Transactions on Vehicular Technology*, vol.63, no.2, pp. 789-801, 2013. Article (CrossRef Link)

[8] L.-F. Liu, Y. Liu and N. S. Zhang, "A Complex Network Approach to Topology Control Problem in Underwater Acoustic Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3046-3055, 2014. Article (CrossRef Link)

[9] D.-Z. Shang, B.-X. Zhang, Z. Yao and C. Li, "An Energy Efficient Localized Topology Control Algorithm for Wireless Multihop Networks," *Journal of Communication and Networks*, vol. 16, no. 4, pp. 371-377, 2014. Article (CrossRef Link)

[10] M.-S. Huang, S.-Y. Chen, Y. Zhu and Y. Wang, "Topology Control for Time-Evolving and Predictable Delay-Tolerant Networks," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2308-2321, 2013. Article (CrossRef Link)

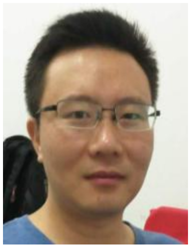
[11] M. Li, Z.-J. Li and A. V. Vasilakos, "A Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues," in *Proc. of the IEEE*, vol. 101, no. 12, pp. 2538-2557, 2013. Article (CrossRef Link)

[12] S. Sardellitti, S. Barbarossa and A. Swami, "Optimal Topology Control and Power Allocation for Minimum Energy Consumption in Consensus Networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 383-399, 2012. Article (CrossRef Link)

[13] O. Awwad, A. Al-Fuqaha, B. Khan and G. B. Brahim, "Topology Control Schema for Better QoS in Hybrid RF/FSO Mesh Networks," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1398-1406, 2012. Article (CrossRef Link)
[14] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick and M. Lvanovich, "A Survey on Distributed Topology Control Techniques for Extending the Lifetime of Battery Powered Wireless Sensor Networks," *IEEE Communication Surveys and Tutorials*, vol. 15, no. 1, pp. 121-144, 2013. Article (CrossRef Link)
[15] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," in *Proc. of IEEE INFOCOM*, pp. 404-413, 2000. Article (CrossRef Link)
[16] J. Yu, H. Roh, W. Lee, S. Pack and D.-Z. Du, "Topology Control in Cooperative Wireless Ad-Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no.9, pp.1771-1779, 2012. Article (CrossRef Link)
[17] N. Li and J. C. Hou, "Localized Fault-Tolerant Topology Control in Wireless Ad Hoc Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no.4, pp. 307-320, 2006. Article (CrossRef Link)
[18] R. Wattenhofer, L. Li, P. Bahl and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. of IEEE INFOCOM*, pp. 1388-1397, 2001. Article (CrossRef Link)
[19] T. M. Chiwewe, G. P. Hancke, "A Distributed Topology Control Technique for Low Interference and Energy Efficiency in Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 11-19, 2012. Article (CrossRef Link)
[20] J. Liu, C. Lin, N. Ye, S.-H. Li and Y. Liu, "Importance Evaluation Algorithm of Links and Nodes in Space Information Networks," in *Proc. of the IEEE Internet Technology and Applications Conference*, pp. 1-4, 2011. Article (CrossRef Link)
[21] N. Ye, Z.-L. Zhu and J. P. Shi, "Distributed Cluster-based Fault-tolerant Topology Control for Space Information Networks," in *Proc. of IEEE Cyber-Enabled Distributed Computing and Knowledge Discovery Conference*, pp. 210-217, 2010. Article (CrossRef Link)
[22] A. Azad, M. Halappanavar, S. Rajamanickam, E. G. Boman, A. Khan and A. Pothen, "Multithreaded Algorithms for Maximum Matching in Bipartite Graphs," in *Proc. of IEEE Parallel and Distributed Processing Symposium*, pp. 860-872, 2012. Article (CrossRef Link)
[23] J. A. Bondy and U. S. R. Murty, *Graph Theory*, Springer, 2008. https://www.springer.com/us/book/9781846289699

Wei Zhang received the B.S. degree (with honors) in communication engineering from Xidian University, Xi'an, China, in 2009 and the M.S. degree from College of Communication Engineering, PLA University of Science and Technology, Nanjing, China, in 2012. He is currently pursuing the Ph.D. degree in communications and information systems in College of Communications Engineering, PLA University of Science and Technology. His research interests include satellite communications, deep space communications, space information networks and wireless sensor networks. He currently serves as a regular reviewer for International Journal of Distributed Sensor Networks.

Gengxin Zhang received his M.S. and Ph.D. degrees from the Institute of Communication Engineer, Nanjing, China in 1990 and 1993 respectively. He is currently a Professor in PLA University of Science and Technology (PLAUST), Nanjing, China. His research interests include the design of communication systems, satellite and deep space communications.



Liang Gou received his B.S. and M.S. degrees from PLA University of Science and Technology (PLAUST), Nanjing, China in 2004 and 2008 respectively. He is currently a Ph.D. candidate in PLAUST. His research interests are focused on satellite communications, deep space communications, network coding and wireless sensor networks.

Bo Kong received the B.S. degree from Xidian University, Xi'an, China in 2010, and his M.S. degree from PLA University of Science and Technology (PLAUST), Nanjing, China in 2013. He is currently a Ph.D. candidate in PLAUST. His research interests are focused on satellite communications, deep space communications, and cooperative communications.

Dongming Bian received his Ph.D. degree from PLA University of Science and Technology (PLAUST), Nanjing, China in 2004. He is currently a Professor in PLAUST. His research interests mainly fall in the area of wireless communications, satellite communications and deep space communications.