# Node-Block based Topology Control Algorithm for Mobile Optical Wireless Communication Networks

**Yan Gao**
Xidian University
**Tao Shang**
Xidian University
**Yuanhao Liu** (✉ 562675696@qq.com)
Xidian University https://orcid.org/0000-0002-3591-6049
**Peiheng Qian**
Xidian University

**Research**

**Keywords:** Topology control, Node-Block, OWC networks, connectivity of network, node degree.

**Posted Date:** November 13th, 2020

**DOI:** https://doi.org/10.21203/rs.3.rs-106012/v1

# Node-Block based Topology Control Algorithm for Mobile Optical Wireless Communication Networks

**Yan Gao [a], Tao Shang [a], Yuanhao Liu [a,*], and Peiheng Qian [a]**

[a]Xidian University, School of Telecommunications Engineering, Taibai South Road, Xi'an, China, 710071

*Liu Yuanhao, E-mail: 562675696@qq.com

**Abstract:** Topology control is an efficient strategy to improve robustness and connectivity in networks. The mobility of nodes, the limited node degree and fragile links in optical wireless communication (OWC) networks make topology control a great challenge. In this paper, the node-block (NB) based topology control algorithm is proposed. Firstly, the proposed algorithm uses the prediction of the contact time between the nodes as the link weight to form a stable tree structure that is called node-block; secondly, the quantized value based on Gamma-Gamma channel model is used as the link weight between node-blocks, and then a multi-link connection is established between any two node-blocks; finally, a connected graph is formed. The performance evaluation parameters, such as topological stability, algebraic connectivity and average node degree are discussed, and their expressions are given. The related simulations are carried out, and comparing with MST algorithm are also made. The results show that our proposed topology control algorithm can ensure the connectivity and stability of the OWC networks, meanwhile, the available node degree are reserved is applied to the large-scale networks.

**Keywords:** Topology control; Node-Block; OWC networks; connectivity of network; node degree.

## *1 Introduction*

Recently, optical wireless communication (OWC) has attracted more and more attentions due to high bandwidth, resistance to electromagnetic interference, high security, flexible networking etc. [1-3]. With the higher demand for QoS and capacity of mobile communications, it is necessary to construct OWC networks to guarantee communications in time among multiple users. Topology control is an important part to design and plan OWC networks.

In traditional wireless networks, there are mainly two perspectives of topology control [4]. The first one controls power to regulate the transmission range in order to reduce the interference between transmission links. An optimized topology control algorithm is proposed to create a power-efficient topology for wireless multi-hop networks [5]. Local minimum spanning tree (LMST) algorithms based on MST are proposed to increase the network capacity as well as reduce the energy [6], [7]. The other one is the selection of potential links. In [8], a Gabriel graph (G graph) algorithm is proposed to discuss the problem of describing geographic variation data and develop statistical method for categorizing sets of populations sampled from different localities. In [9], localized Delaunay triangulation with application is proposed in Ad Hoc wireless networks.

In optical wireless communication networks (OWCNs), the topology control is significantly different than that in traditional wireless networks. An OWC node can carry only a limited number of transceivers due to size, weight, and power, which means a small degree of node [10]. And the small divergence angle of laser make it must be considered that an APT (acquisition, pointing and tracking) subsystem needs to track the fast moving target. Therefore, it is becoming more difficult to set up OWCNs.

This paper mainly focuses on how to rapidly and efficiently construct network among multiple nodes in the middle and high-speed mobile applications. Considering the small divergence angle of laser, small degree of node, node-block (NB) based topology control algorithm (NB algorithm for short) is proposed. The algorithm includes two sub algorithms:

1) node-block formation algorithm; 2) topology formation algorithm among node-blocks. According to these two sub algorithms, the NB algorithm is divided into two steps to form a whole connected graph. First, taking the prediction of contact time as the link weight, a high stability tree structure is formed by a number of nodes with similar properties (such as moving trail, velocity, and position) and it is called node-block. Second, thinking of the node-block as a special node, the multi-link topology among these node-blocks is formed according to the link reliability weights, and then a connected graph is formed. The corresponding criteria are presented, including topological stability, algebraic connectivity, and average node degree, and they are used to measure network robustness, reliability, and connectivity. The related simulations are executed, and comparing with MST algorithm are also made in aspect of feasibility, effectiveness and stability. The results show that the proposed NB algorithm can achieve a high topology connectivity to form a stable topology structure in the case of limited node degree.

The remainder of this paper is organized as follows. Section II introduces the system model and performance measures. Section III presents and analyzes the NB algorithm. Our simulation results and performance discussions are given in Section IV. Finally, Section V concludes this paper.

## 2 System Model and Performance Measures

### *2.1 Network Model*

Network topology is the arrangement of various elements (links, nodes, etc.) of a communication network, and it is the topological structure of a network. The topology use two types of nodes, the fixed node and mobile node, and it can be grouped into two categories: static and dynamic [11]. In a static topology, the links between two nodes are passive and cannot be reconfigured for direct connections to other nodes. The dynamic topology network is generally characterized by distributed control topology, including planar structure and hierarchical structure, also known as layered structure and fully distributed structure. In this paper, we use a fully distributed structure model of wireless communication network in which all the nodes have the same status in the factors of network control, routing and traffic management, as shown in Fig. 1.

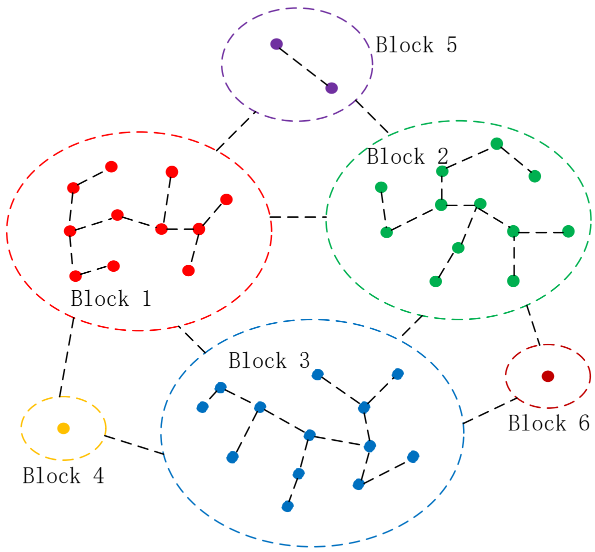



**Fig. 1.** Fully distributed structure of mobile nodes in OWCNs

In Fig. 1, the node-block is a tree structure formed by a number of nodes with similar properties (such as moving trail, velocity, and position), and it can reduce the number of physical links between nodes to save their node degree. Due to the small relative motion range of nodes, the node-block has a stable topology.

For facilitating the analysis, we make the following hypotheses about nodes:

- Each node has a unique node ID (identification) and a node-block ID.
- Each node has K transceivers, also known as the node degree K of physical layer nodes.

- The transmission power of each transceiver is adjustable to conserve energy and the longest transmission distance of each one is less R meters.
- The APT technology has been able to meet the needs of the model, which means that a link between the transmitter and the receiver can be established within the given time.
- Each node can obtain the local information from the neighboring nodes, including node ID, node-block ID, position, velocity, and acceleration.
- All nodes have the capability of host and router so that packets can be transmitted and received.

## 2.2 *Node Mobility Model*

In this paper, the smooth random mobility (SRM) model is used. Unlike other random models, SRM model better captures the individual movement behavior of the mobile users using a combined modeling of direction and speed control [12], [13]. The node in the model has its respective moving trail without interference with each other. The SRM model can correspond to the real-life scenarios as well as possible, such as the smooth speed and direction rather than the mutation.

### *2.2.1 Speed control*

The concept for modeling the speed behavior of nodes is based on the use of target speed (the speed that a node intends to achieve) and linear acceleration [12]. The node moves at the current speed until a new target speed is determined by the random process; after that, it speeds up or slows down to the target speed, and then it moves at the constant speed. In this process, the motion state of a node at time $t$ can be described by the following three parameters:

- $v(t)$: the current speed in $\frac{m}{s}$;
- $a(t)$: the current acceleration in $\frac{m^2}{s}$;
- $v^{'}(t)$: the current target speed in$\frac{m}{s}$.

The time between two-time steps is denoted as $\Delta t$. Then the term $1/\Delta t$ represents the time step number [12]. According to Newton's law of motion, in each time step ($\Delta t$), the new current speed can be expressed as:

$$v(t) = v(t - \Delta t) + a(t) \cdot \Delta t \tag{1}$$

until $t^{'}$ time to reach the target speed, that is$v(t^{'}) = v^{'}(t)$.

*2.2.2 Direction control*

The principle for direction control is similar to the speed control principle and each node has an initial direction $\varphi_0 = \varphi(t = 0)$ which is chosen from a uniform distribution over$[0,2\pi]$. The direction behavior of a node at time $t$ can be controlled by the following three parameters:

- its current direction $\varphi(t)$;
- its current direction changes $\frac{\Delta\varphi(t)}{\Delta t}$ in $s^{-1}$;
- its current target direction $\varphi^{'}(t)$.

During the free space motion, the real-time direction of the node is:

$$\varphi(t) = \varphi(t - \Delta t) + \Delta\varphi(t) \tag{2}$$

until $t^{'}$ time to reach the target direction, that is$\varphi(t^{'}) = \varphi^{'}(t)$.

In the SRM model, the moving path of single node is shown in Fig. 2.

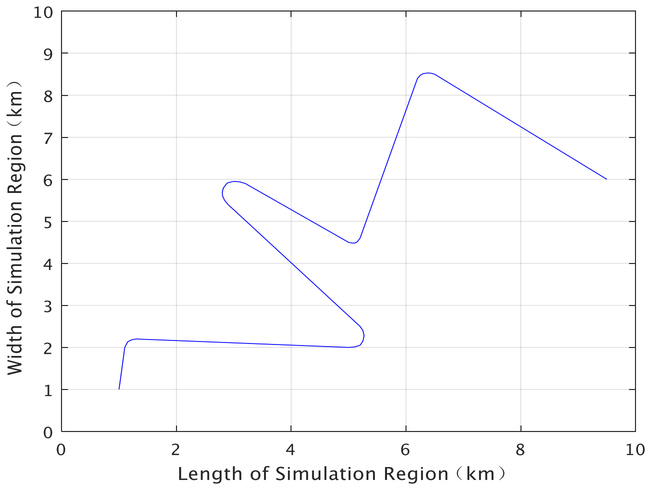



**Fig. 2.** Single node moving path in SRM model

### *2.3 Performance Measures*

We evaluate the performance of OWCNs through three parameters: topological stability, algebraic connectivity, and average node degree.

#### *2.3.1 Topological stability*

It is worthwhile to study whether a link disconnect before the next topology update in the node-blocks. We measure the topological stability by using the predictive and actual links among nodes. Specifically, we define the predicted edge set $E$ of a graph, and $n(E)$ denotes the number of elements in $E$. A subset $E^{'}$ of $E$ is defined to express these edges in actual operation and $n(E^{'})$ is used to represent the number of elements in $E^{'}$. The topological stability can be expressed as:

$$\rho = \frac{n(E^{'})}{n(E)} \tag{3}$$

From the above equation, the greater $\rho$ is, the more stable the topology is.

#### *2.3.2 Algebraic Connectivity*

Connectivity is defined as the minimum number of nodes that must be removed to disconnect a network [10]. A common measure of network reliability is connectivity. As other invariants reflecting the capability of graph connectivity, the algebraic connectivity is considered as a quantitative measurement of graph connectivity. According to Fiedler, the algebraic connectivity of a graph $G$ can be expressed as the second smallest eigenvalue of its Laplacian Matrix$L(G)$ [14][15], which is also called$\lambda_2(L)$. The Laplacian Matrix$L(G)$ can be expressed as [16]

$$L(G) = D(G) - A(G) \tag{4}$$

where $A(G)$ is the adjacency matrix of graphs$G$and $D(G)$ is diagonal matrix consisting of vertex degree of graphs$G$. $D(G)$ is expressed as

$$D(G) = diag\big(d(v_1), d(v_2), \cdots, d(v_n)\big) \tag{5}$$

where $d(v_k)$ is the connection degree of the vertex$v_k$.

### *2.3.3 Average Node Degree*

For an OWCN with N nodes, the average node degree is expressed as [10]

$$D_{avg} = \frac{(\sum_{i=1}^{N} d(i))}{N} \tag{6}$$

where $d(i)$ is the degree of node $i$.

We use an average node degree as a parameter to measure the entire network connectivity from the perspective of the network topology.

## 3 Toplogy Control Algorithm

### *3.1 Node-Block Formation Algorithm*

### *3.1.1 Link Weight*

In this section, we define the evaluation criteria of the link establishment in the node-block, which is the link weight. This algorithm is designed for the nodal mobility model to mainly overcome the problem of frequent changes in mobile network. So we consider it is appropriate that the contact time is used as the link weight. The contact time can be expressed as

$$T_{LAB} = min_t\{\|X_A(t) - X_B(t)\| > R\} \tag{7}$$

where, $X_A(t)$ and $X_B(t)$ is the position of node A and node B at time $t$ respectively, $T_{LAB}$ is the time that node A and node B are connected before moving out of the transmission range of each other, and R is the maximum communication distance between nodes.

We define the time threshold $T_{th}$, and if $T_{LAB} > T_{th}$ , it is stated that the link is stable before the next topology update, and the link becomes an alternative link. $T_{th}$ is calculated by

$$T_{th} = \lambda \frac{R}{\hat{v}} \tag{8}$$

where $\lambda$ is the centrifugal coefficient and its value is the ratio of the average distance of all nodes to the maximum communication distance, $\hat{v}$ is the average reverse speed between two nodes.

There is a special equation for calculating the value of$\hat{v}$, if the velocity ($\vec{v}_A$) of node A and the velocity ($\vec{v}_B$) of node B are known.

$$\begin{aligned} \hat{v} &= \left\| \vec{v}_A - \vec{v}_B \right\| \\ &= \bar{v} \int_0^{2\pi} \frac{\sqrt{(1+\cos\theta)^2 + \sin^2\theta}}{2\pi} d\theta \\ &= \frac{4}{\pi} \bar{v} \end{aligned} \tag{9}$$

where $\bar{v}$ is the average speed of the nodes and $\theta$is the movement direction angle of the nodes.

The current parameters of the nodes can be provided as position, speed, target speed and acceleration. The contact time $T_{LAB}$ can be obtained by calculating the link expiration time between the two nodes within the communication range. To calculate the contact time, the distance $L(t)$ between two nodes at movement time $t$ needs to meet the following equation.

$$L(t) = R \tag{10}$$

In our algorithm, we calculate $T_{LAB}$ with two nodes (assuming node A and node B) as an example. We assume that the initial distance is $L_0$ between node A and node B; the initial speed of node A and node B are $v_{A0}$ and $v_{B0}$ respectively; the acceleration are $a_A$ and $a_B$; the target speed are $v_{At}$ and $v_{Bt}$; the time required for these two nodes to accelerate to the target speed are $t_A$ and $t_B$; and the moving distances of the acceleration phase are $S_{A0}$ and $S_{B0}$. After the acceleration or deceleration, the node will reach the target speed and then it will move at a constant speed. We calculate the $T_{LAB}$ as follows.

**Step1:**$L_0$, $S_{A0}$, $S_{B0}$, $t_A$ and $t_B$ are calculated by:

$$L_0 = \|X_A(t_0) - X_B(t_0)\| \tag{11}$$

$$S_{A0} = \frac{v_{At}^2 - v_{A0}^2}{2a_A}, S_{B0} = \frac{v_{Bt}^2 - v_{B0}^2}{2a_B} \tag{12}$$

$$t_A = \frac{v_{At} - v_{A0}}{a_A}, t_B = \frac{v_{Bt} - v_{B0}}{a_B} \tag{13}$$

**Step2:** $T_{LAB}$ is calculated in the following four cases:

(1) For $S_{A0}+S_{B0}+L_0<R$ and $t_A \geq t_B$, the speeds of node A and node B have been accelerated to the target speed, and then these two nodes bear a constant speed $v_{At}$ and $v_{Bt}$, respectively. The distance $L(t)$ can be obtained by

$$L(t) = S_{A0}+S_{B0}+v_{Bt}(t_A-t_B)+(v_{At}+v_{Bt})t+L_0 \quad (14)$$

where $t$ is the time for node A moving at a constant speed $v_{At}$.

Substituting $L(t)$ into Eq. (10) to get $t$, $T_{LAB}$ can be obtained by:

$$T_{LAB} = t_A + t \quad (15)$$

(2) For $S_{A0}+S_{B0}+L_0<R\text{and}t_A<t_B$, the distance $L(t)$ can be obtained by:

$$L(t) = S_{A0}+S_{B0}+v_{At}(t_B-t_A)+(v_{At}+v_{Bt})t+L_0 \quad (16)$$

where $t$ is the time for node B moving at a constant speed $v_{Bt}$.

Substituting $L(t)$ into Eq. (10) to get $t$, $T_{LAB}$ can be obtained by:

$$T_{LAB} = t_B + t \quad (17)$$

(3) For $S_{A0}+S_{B0}+L_0 \geq R$ and$t_A \geq t_B$, it is indicated that at least one of the two nodes does not accelerate to the target speed when they are within the scope of direct communication. First, we discuss that node B completes the acceleration and executes uniform rectilinear movements, and that node A will continue to accelerate. So, when node B completes the acceleration, the movement distance $S_{A0}^{'}$ of node A can be obtained by:

$$S_{A0}^{'} = v_{A0}t_B + \frac{1}{2}a_A t_B^2 \quad (18)$$

- For $S_{A0}'+S_{B0}+L_0<R$, the distance $L(t)$ can be obtained by:

$$L(t) = \left(v_{A0}t+\frac{1}{2}a_A t^2\right)+\left(S_{B0}+v_{Bt}(t-t_B)\right)+L_0 \quad (19)$$

where $t$ is the time for node A at the acceleration stage.

Substituting $L(t)$ into Eq. (10) to get $t$, $T_{LAB}$ can be obtained by:

$$T_{LAB} = t \tag{20}$$

- For $S_{A0} + S_{B0} + L_0 \geq R$, the distance $L(t)$ can be obtained by:

$$L(t) = \left(v_{A0}t + \frac{1}{2}a_A t^2\right) + \left(v_{B0}t + \frac{1}{2}a_B t^2\right) + L_0 \tag{21}$$

where $t$ is the time for node A at the acceleration stage.

Substituting $L(t)$ into Eq. (10) to get $t$, $T_{LAB}$ can be obtained by:

$$T_{LAB} = t \tag{22}$$

(4) For $S_{A0} + S_{B0} + L_0 \geq R$ and$t_A < t_B$, in the same way, the movement distance $S_{B0}^{'}$ of node B at the time of the completion of the accelerated node A can be obtained by:

$$S_{B0}^{'} = v_{B0}t_A + \frac{1}{2}a_B t_A^2 \tag{23}$$

- For $S_{A0} + S_{B0}' + L_0 < R$, the distance $L(t)$ can be obtained by:

$$L(t) = \left(S_{A0} + v_{At}(t - t_A)\right) + \left(v_{B0}t + \frac{1}{2}a_B t^2\right) + L_0 \tag{24}$$

where $t$ is the time for node B at the acceleration stage.

Substituting $L(t)$ into Eq. (10) to get $t$, $T_{LAB}$can be obtained by:

$$T_{LAB} = t \tag{25}$$

- For $S_{A0} + S_{B0}' + L_0 \geq R$, the distance $L(t)$ can be obtained by:

$$L(t) = \left(v_{A0}t + \frac{1}{2}a_A t^2\right) + \left(v_{B0}t + \frac{1}{2}a_B t^2\right) + L_0 \tag{26}$$

where $t$ is the time for node B at the acceleration stage.

Substituting $L(t)$ into Eq. (10) to get $t$, $T_{LAB}$ can be obtained by:

$$T_{LAB} = t \tag{27}$$

*3.1.2 Algorithm Description*

The node-block formation algorithm is one of the core algorithms, and its feature is that a node-block with tree structure is formed according to the contact time $T_{LAB}$ between nodes. The node-block formation algorithm is presented in Table I, and the detailed steps are as follows.

**Step1:** Node initialization, i.e. to determine their initial position, velocity, acceleration, and node degree, and to set their own node IDs as the node-block IDs (after the initialization is completed, each node as a node-block and its node-block ID equals its node ID).

**Step2:** Each node (assuming node A) using APT subsystem obtains the information of an available neighbor node (assuming the neighbor node B). The information includes the current position, node degree, velocity, acceleration, and target speed.

**Step3:** According to the acquired information, node A determines whether the node-block ID of Node B is in line with its own. If it is, it means node A and node B belong to a same node-block, and then node A selects the next neighbor node (jump to Step2); otherwise, node A calculates the contact time $T_{LAB}$.

**Step4:** Comparing $T_{LAB}$ with the threshold time$T_{th}$, if$T_{LAB} > T_{th}$, this link is chosen as an alternative link, and then node A continues to proceed to the next step; otherwise, node A selects the next neighbor node (jump to Step2).

**Step5:** Node A sends a building request (BSEQ) to node B, and informs $T_{LAB}$ to node B. In this step, if node A also receives the BSEQ sent by node B at the same time, the two nodes need to use binary back off algorithm to avoid the collision. And then node A continues to proceed to the next step.

**Step6:** Node B calculates the contact time $T_{LBC}$ ($T_{LBC}$ indicates that node B keeps the contact time with its own node-block (assuming Block-C)). Comparing $T_{LAB}$ and $T_{LBC}$, if $T_{LAB} < T_{LBC}$, node B sends a building rejection (BREJ) to node A to reject this request; otherwise it sends a building ACK (BACK) to node A. If receiving BREJ, node A reselects the next neighbor node (jump to Step2); if receiving BACK, node A establishes a link with node B, and continues to the next step.

**Step7:** Compare the node-block ID of node A with node B. If the node-block ID of node B is less than that of node A, change the node-block ID of node A to the node-block ID of node B; vice versa.

**Step8:** Broadcast the data packets of "block ID change instruction" to other members of the same node-block, and other nodes in the node-block to change their own node-block ID after receiving the message.

**Step9:** Node A continues to select the next neighbor node (jump to Step2). Until all the neighbor nodes have been traversed, the algorithm ends.

TABLE I

Node-Block Formation Algorithm

Wait for awakening;
Read neighbor information, including node ID, node-block ID, position, velocity, acceleration, target speed;
**WHILE** (there is still neighbor node for the group)
  **IF** (node does not belong to the same node-block)
    Calculate the contact time $T_{LAB}$;
    **IF** ($T_{LAB}$ is higher than the threshold time $T_{th}$)
      Calculate the contact time $T_{LBC}$ of node B with its own node-block;
      **IF** ($T_{LAB}$ is higher than $T_{LBC}$)
        Establish a link;
        Modify relative information;
      **END**
    **END**
  **END**

14: **END**

### 3.2 Topology Formation Algorithm among Node-Blocks

#### 3.2.1 Link Weight

Through node-block formation algorithm, the links that belong to the same node-block are stable. However, the links among node-blocks may be become brittle, because the nodes that belong to different node-block have different properties (such as moving trail, velocity, and position). In order to ensure the stability of the links among node-blocks, the link reliability will be considered in mobile OWC networks. The link weight is determined by the link reliability of the atmospheric channel model, that is the Gamma-Gamma channel model.

Given a threshold of channel state h0, the link reliability is expressed as [8]

$$\begin{aligned} L_{ij} &= \int_{h_0}^{\infty} \frac{2(\alpha\beta)^{\frac{\alpha+\beta}{2}}}{\tilde{h}\Gamma(\alpha)\Gamma(\beta)} \left(\frac{h}{\tilde{h}}\right)^{\frac{\alpha+\beta}{2}-1} K_{\alpha-\beta}\left(2\sqrt{\alpha\beta\frac{h}{\tilde{h}}}\right) dh \\ &= \frac{\alpha\beta}{\tilde{h}\Gamma(\alpha)\Gamma(\beta)} \left(\alpha\beta\frac{h_0}{\tilde{h}}\right)^{\frac{\alpha-\beta}{2}} G_{1,3}^{3,0}\left(\alpha\beta\left(\frac{h_0}{\tilde{h}}\right)\middle|\begin{matrix} -\frac{\alpha+\beta}{2} \\ 1-\frac{\alpha+\beta}{2}, \frac{\alpha-\beta}{2}, \frac{\beta-\alpha}{2} \end{matrix}\right) \end{aligned} \tag{28}$$

where, $K_{\varphi}(.)$ is the modified Bessel function of the second kind of order; $\Gamma(.)$ is the gamma function; α and β are parameters that are related to distance, aperture diameter, and atmospheric conditions; h denotes the channel state; and $\tilde{h}$ is a scale parameter that is related to geometric spread loss and path attenuation. The Meijer G-function can be used to solve above equation.

We assume that the links are symmetric, *i.e.*, $L_{ij} = L_{ji}$ for all $i \neq j$. According to the adopted channel model, the link weight $\omega_{ij}$ is as follows:

$$\omega_{ij} = \begin{cases} L_{ij}, & L_{ij} \geq I_{th} \\ 0, & others \end{cases} \tag{29}$$

where $I_{th}$ is the connection threshold and its value is determined by the topology update cycle, the relative speed of nodes, and numbers of the links among node-blocks.

### *3.2.2 Algorithm Description*

$$P = \begin{vmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MM} \end{vmatrix} \qquad p_{ij}\text{:}\ \overbrace{\boxed{\mathrm{L}(1)}\boxed{\mathrm{L}(2)}\ \cdots\ \boxed{\mathrm{L}(x)}}^{x} \quad \omega_1 \le \omega_2 \le \ldots \le \omega_x$$

**Fig. 3.** Priority queue matrix $P$ and its priority queue $p_{ij}$

In this algorithm, the node-block is considered as a special node and the links among node-blocks are established, and then a connected graph is formed ultimately. We assume that the maximum number of links between arbitrary two node-blocks is x, and that each node-block knows its neighbor node-block information. In order to select these links between arbitrary two node-blocks, firstly the priority queue matrix $P$ is set as shown in Fig. 3, where $M$ equals the number of all the node-blocks. The matrix element is a priority queue $p_{ij}$ size of x, where $i$ and $j$ are the node-block ID, and all the links $L(1), L(2) \cdots L(x)$ in $p_{ij}$ sort by its link weight $\omega_1, \omega_2 \cdots \omega_x$ from smallest to largest.

The topology formation algorithm among node-blocks is presented in Table II. We take Block-A for example, and all the nodes $A_1, A_2 \cdots A_i \cdots A_k$ inside Block-A execute this algorithm, where $k$ is the number of nodes inside Block-A. The detailed steps are as follows:

**Step1:** Node $A_i$ determines its node degree. If node $A_i$ has no remaining node degree, node $A_i$ has been executed, jump to step5; otherwise continue to the next step.

**Step2:** Node $A_i$ finds all the neighbor node-blocks, and then selects the neighbor node with the maximum link weight from each neighbor node-block, and adds them to its own neighborhood node set ($V_A$).

**Step3:** Node $A_i$ determines whether $V_A$ is empty. If it is empty, node $A_i$ has been executed, and the operation of the priority queue $p_{AB}$ has completed to get the links between Block-A and Block-B, jump to Step5. Otherwise node $A_i$ takes a node (such as node B belonging to Block-B) from $V_A$ with the link weight $\omega_{AB}$.

**Step4:** Node $A_i$ compares $\omega_{AB}$ and the minimum link weight in $p_{AB}$. If $\omega_{AB}$ is greater than this minimum value, this link between Block-A and Block-B is added to $p_{AB}$. Otherwise, back to Step3.

**Step5**: Until all the nodes inside Block-A are traversed, all the links in $p_{A1} \cdots p_{AB} \cdots p_{AM}$ between Block-A and other node-blocks can be obtained, and then connect all the links. The topology formation between Block-A and other node-blocks has completed.

TABLE Ⅱ

Topology Formation Algorithm among Node-Blocks

**IF** (Node $A_i$ has its remaining node degree)<br>
  Finds available neighbor nodes and add them to its neighbor node set $V_A$;<br>
    **WHILE** ($V_A$ is not null)<br>
        Take a node from $V_A$ with the maximum link weight $\omega_{AB}$;<br>
        **IF** ($\omega_{AB}$ is greater than the minimum value of link queue $p_{AB}$)<br>
            Add the link to the link queue $p_{AB}$;<br>
        **END**<br>
    **END**<br>
**END**

## 4 Simulations and Analysis

Performance measures are discussed and defined in Section II. In the section, we show the simulation and analysis results. All 100 nodes generated with random are located in a 10km*10km square. Each node has K (3~23) transceivers that can be used to build links with other nodes and its maximum transmission distance is 2km. The time threshold *Tth* = 60s and the simulation time T = 600s. The moving paths of the 100 nodes in SRM model are shown in Fig. 4.

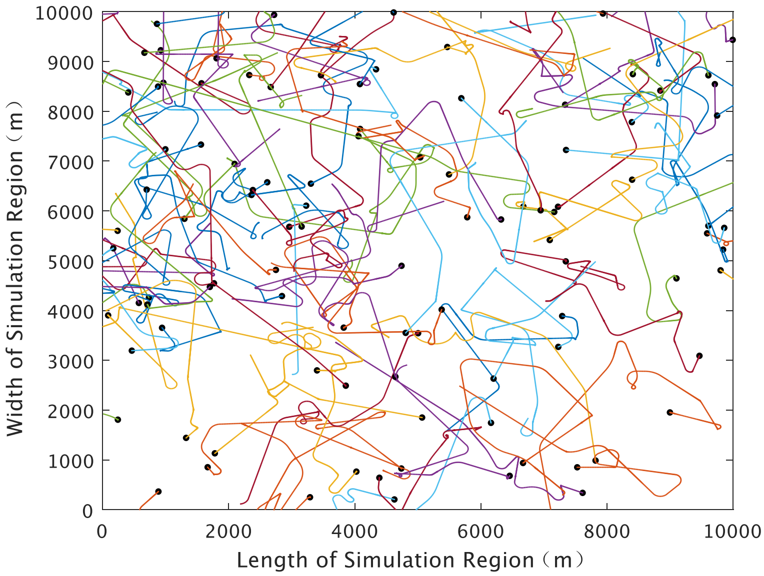



**Fig. 4.** Moving paths of 100 random nodes in SRM model

From Fig. 4, the black point represents the initial position of the nodes. The results show that all the nodes are moving independently and have no mutation direction. Thus, this model can basically meet the need for simulation work to adapt to the real environment.

### *4.1 Node-Block Formation Algorithm*

#### *4.1.1 Simulation Results*

In topology update period T=60s, the topology of node-block is shown in Fig. 5. The internal topology of the node-block is a tree structure or a single node. For examples, in Fig. 5, the nodes whose node IDs are 6, 18, 68, 63, 11, 32, 96 belong to the same node-block, and the single node whose node ID is 69 belongs to a node-block. And there is only one path between any two nodes in the same node-block, which can save the number of limited transmitter and receiver. Due to the tree structure and the small relative motion between the nodes in the same node-block, the topology structure is constructed with high robustness and more residual node degree.

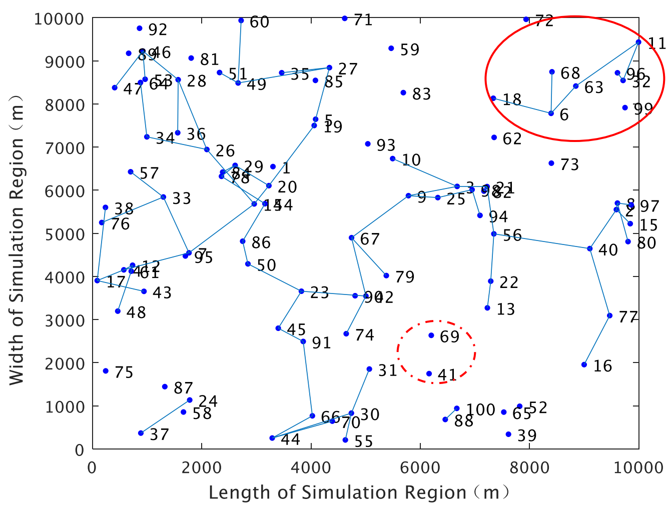



**Fig. 5.** Topology of node-block

### *4.1.2 Topological Stability of Node-Blocks*

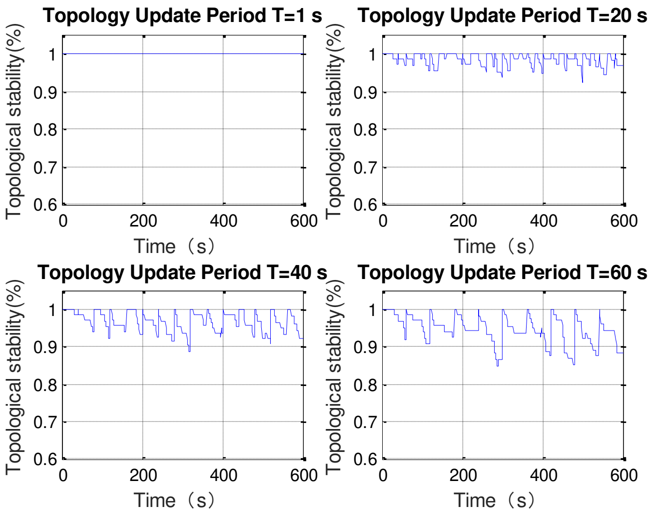



**Fig. 6.** Topological stability of node-blocks

The time threshold $T_{th}$ = 60s, and the update period of topology are set as T= 1s, 20s, 40s and 60s. The topological stability of node-block is obtained by simulation, as shown in Fig. 6. The results show that when T= 1s, the most stable node-block is obtained. With the increase of T, the stability of the node-blocks is decreased. Even if the topology update cycle is 60s, the topological stability of node-blocks is also above 0.8. It means that more than 80% of the links have not been broken after 60s and this node-block is also stable. However, the topological stability of these node-blocks can increase to 1 after topology update, and it means the topology is still stable after update. We can see that the predicted results are in agreement with the theory results. That is to say, the topology is stable after update. But in the remaining time, the nodes in the node-block may move out of the node-block, namely some links in the original node-block disconnect due to the node movement, and the topological stability of node-block becomes small. This is the impact of the prediction, but it will not be accumulated over time and will be eliminated after update.

### *4.2 Topology Formation Algorithm among Node-Blocks*

### *4.2.1 Simulation Results*

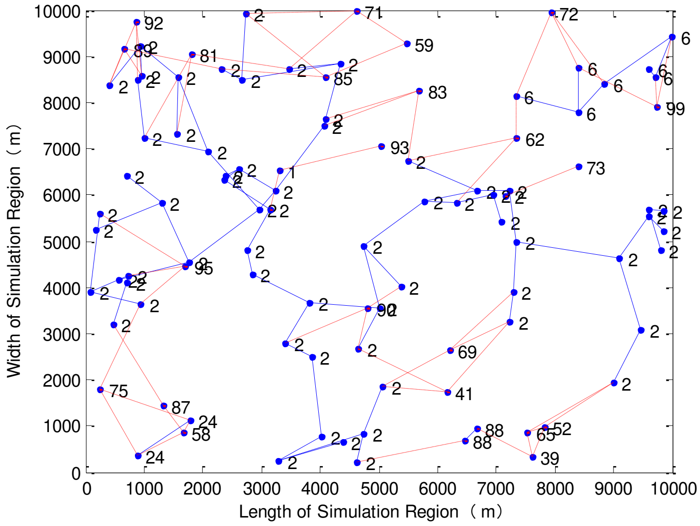



**Fig. 7.** Topology of connected graph

After the first step of the "node-block formation algorithm", all of the node-blocks have been formed. Each node in the node-block has the same node-block ID that is the minimum ID for member nodes. The second step "topology formation algorithm among node-blocks" will be executed between arbitrary two node-blocks, and ultimately a complete connected graph is formed, as shown in Fig. 7. The number represents the ID of node-block. The solid lines are all the links of nodes in the same node-block, and the dotted lines are the topological connections among the node-blocks.

### *4.2.2 Topological stability of connected graph*

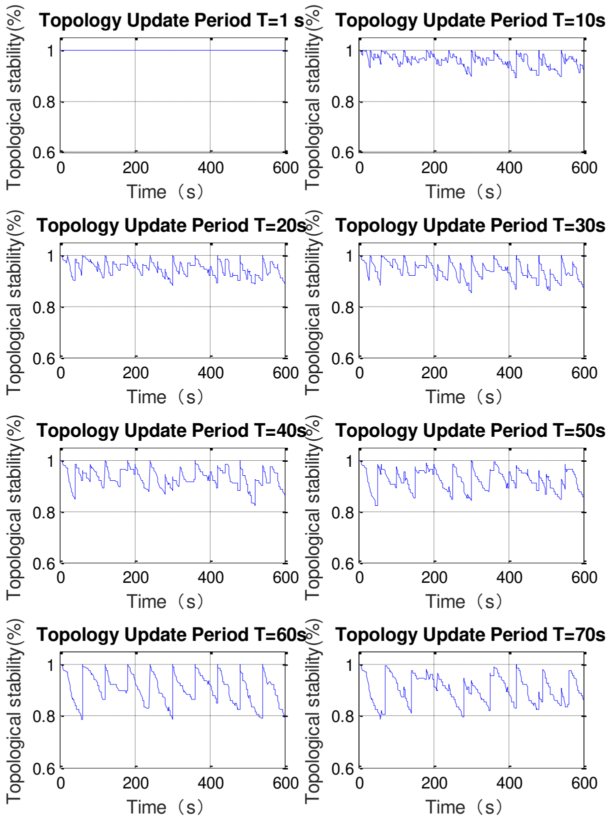



**Fig. 8.** Topological stability of connected graph

We analyze the topological stability of the connected graph with topology update period T=1s, 10s, 20s, 30s, 40s, 50s, 60s and 70s respectively, as shown in Fig. 8. The results show that the topological stability decreases with the increase of T, but the topological stability is still high. When T=70s, its topological stability still maintains at more than 0.8.

## *4.3 Algorithm Analysis*

### *4.3.1 Algebraic Connectivity*

We simulate the algebraic connectivity of the connected graph formed by the NB algorithm, and compare it with the typical topology formation algorithm, MST (Minimum Spanning Tree) algorithm, as shown in Fig. 9. The node degree for the simulation is K=3, and the number of nodes is N=100, 150 and 200. Simulation results show that the algorithm presented in this paper has a great advantage over the MST algorithm in algebraic connectivity, which is about 0.02. As known that the higher the algebraic connectivity is, the higher the topology connectivity is. At the same time, the relationship between the number of nodes and the algebraic connectivity is also compared, which shows that the algebraic connectivity gradually decreases with the increase of the size of the network. That is to say, the nodes far apart will not be able to establish a direct link with the increasing of nodes number. At this time, the network no longer guarantees that any two nodes have a direct link as the small network is.

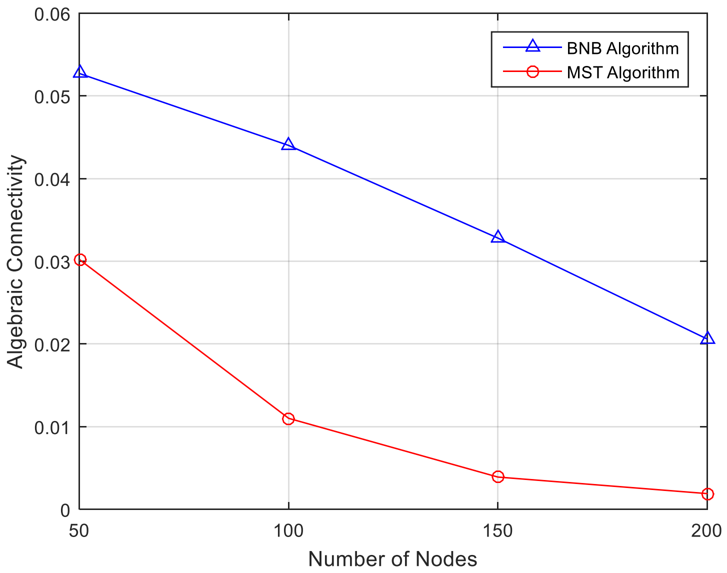



**Fig. 9.** Comparison of algebraic connectivity between NB and MST Algorithm

We also simulate the influence of the maximum degree of nodes on the algebraic connectivity by the NB algorithm, and compare it with MST algorithm, as shown in Fig. 10. The node degree for the simulation is K=3~23, and the number of nodes is N=100. The topology formed by MST is tree structure and no loop, and it has one, and only one path between any two nodes, moreover, its algebraic connectivity is relatively small and unchanged with the increase of the maximum degree of nodes. However, the algebraic connectivity of NB algorithm increases with the maximum degree of nodes. When the node degree is relatively small (less than 10), the algebraic connectivity increased faster, but when the node degree increases to a certain number, the algebraic connectivity almost no longer increases. The reason is that all the adjacent nodes have been connected with the increasing maximum degree of nodes. Based on the result of algebraic connectivity analysis, the NB algorithm can guarantee the connectivity of the topology.

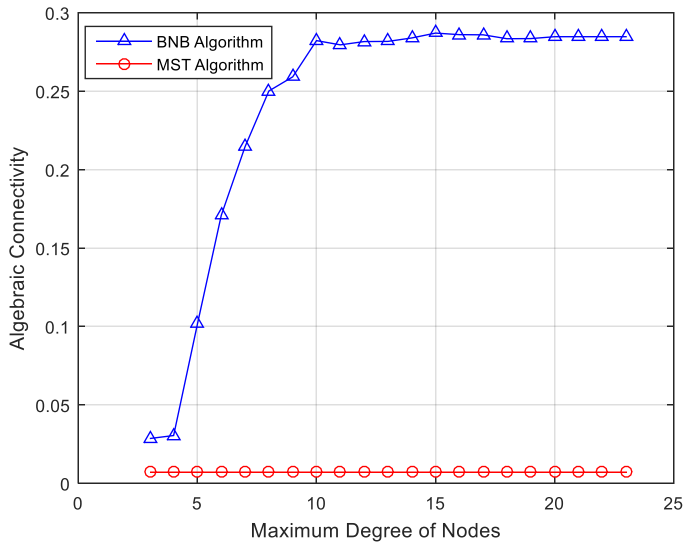



**Fig. 10.** Comparison of algebraic connectivity between different node degrees

*4.3.2 Average Node Degree*

The average node degree of connected graph is calculated according to Eq. (6), and it is closely related to the network topology connectivity. The node degree for the simulation is K=3, and the number of nodes is N=100, as shown in Fig. 11. It can be seen that the average node degree of the MST algorithm is 0.99, and that of the NB algorithm maintains on the domain (2, 2.5), which is smaller than the upper bound of the node degree. From this simulation results, the available node degrees are reserved for the nodes to join, and the node degree also has not been wasted to enhance the connectivity of whole networks, so the NB algorithm is better than the MST algorithm.

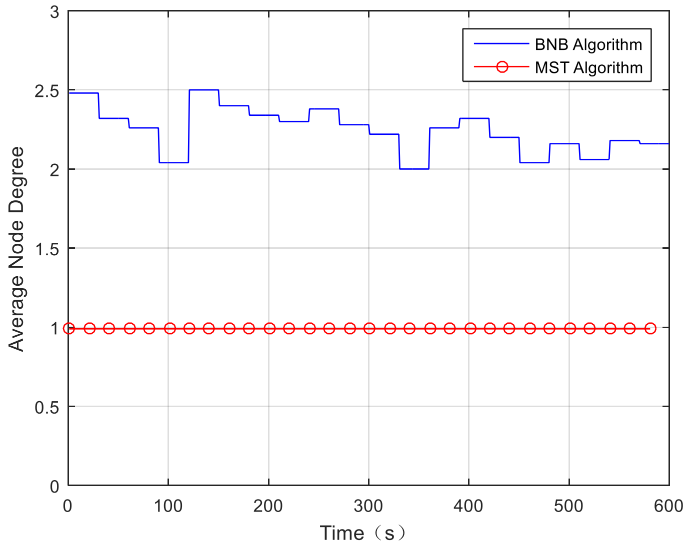



**Fig. 11.** Comparison of average node degree between NB and MST algorithm

## 5 Conclusions

Due to the limited divergence angle of laser beam, a few node degrees, and the performance limitations of APT, the study of fast and efficient networking for mobile OWCNs faces many challenges. In this paper, an OWCN topology control algorithm, NB algorithm, is proposed. According to the link weight determined by the predictive value of the contact time between any two nodes, the first sub algorithm (i.e. node-block formation algorithm) is presented to form a relatively stable node-block composed of a number of nodes with similar properties (such as moving trail, velocity, and position). And then the second sub algorithm (i.e. topology formation algorithm among node-blocks) is presented, where the quantized value based on Gamma-Gamma channel model is used as the link weight between node-blocks. That is to say, due to the different properties between two nodes that belong to different node-blocks, the link becomes fragile and a multi-link connection is adopted between any two node-blocks.

Compared to the Random Waypoint Model, the SRM model is used in this algorithm to ensure the facticity and authenticity. In order to evaluate the stability of the OWCNs, topological stability is used as a measure of robustness. The simulation results show that the topological stability can be maintained at more than 0.8. In order to analyze the reliability of the OWCNs, algebraic connectivity and average node degree are adopted. Comparison with MST algorithm is also made. Simulation results show that our NB algorithm has an obvious advantage over the MST algorithm in aspect of algebraic connectivity, which is about 0.02.

Theoretical analysis and simulation results show that the NB algorithm has obvious advantages on the conditions of mobile environment and the limited node degree, which can effectively solve the problem of uneven distribution of topology connectivity degree. Therefore, our proposed algorithm is feasible and effective for mobile OWCNs.

## 6 Declarations

*Acknowledgment*

This work is supported in part by the 863 High Technology Plan of China (2013AA013402), the National Natural Science Foundation of China (61172080, 61771357). Corresponding author: Tao Shang (email: shtsun_sjtu@hotmail.com, 562675696@qq.com).

*Funding*

The authors acknowledge the 863 High Technology Plan of China (2013AA013402), and the National Natural Science Foundation of China (61172080, 61771357).

*Authors' contributions*

YG and TS are the main writers of this work. They proposed the main idea and described the whole relaying process. YL derived the formulas. YL, YG and ZD preformed the simulations and edited the manuscript. All authors read and approved the final manuscript.

*Availability of data and materials*

Not applicable.

*Competing interests*

The authors declare that they have no competing interests.

*References*


[1] Uysal M, Nouri H. Optical wireless communications — An emerging technology[C]// Transparent Optical Networks (ICTON), 2014 16th International Conference on. IEEE, 2014:1-7.

[2] Zhou H, Babaei A, Mao S, et al. Algebraic connectivity of degree constrained spanning trees for FSO networks[C]// Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013:5991-5996.

[3] Shang T, Yang Y, Ren G, et al. Topology control algorithm and dynamic management scheme for mobile FSO networks[J]. Journal of Optical Communications & Networking, 2015, 7(9):906-917.

[4] Sun F, Shayman M. Minimum interference algorithm for integrated topology control and routing in wireless optical backbone networks[C]// IEEE International Conference on Communications. 2004:4232-4237 Vol.7.

[5] Hajiaghayi M T, Immorlica N, Mirrokni V S. Power Optimization in Fault-Tolerant Topology Control Algorithms for Wireless Multi-hop Networks[J]. IEEE/ACM Transactions on Networking, 2010,

15(6):300-312.

[6] Li N, Hou J C, Sha L. Design and analysis of an MST-based topology control algorithm[J]. IEEE Transactions on Wireless Communications, 2005, 4(3):1195-1206.

[7] Gallager R G, Humblet P A, Spira P M. A Distributed Algorithm for Minimum-Weight Spanning Trees[J]. Acm Transactions on Programming Languages & Systems, 1983, 5(1):66-77.

[8] Gabriel K R, Sokal R R. A New Statistical Approach to Geographic Variation Analysis[J]. Systematic Zoology, 1969, 18(3):259-278.

[9] Li X Y, Calinescu G, Wan P J, et al. Localized Delaunay triangulation with application in ad hoc wireless networks[J]. IEEE Transactions on Parallel & Distributed Systems, 2003, 14(10):1035-1047.

[10] Hu Z, Verma P, Sluss , J. Improved reliability of free-space optical mesh networks through topology design[J]. Journal of Optical Networking, 2008, 7(5):436-448.

[11] Feng T. A survey of interconnection networks[J]. Computer, 1981, 14(12): 12-27.

[12] Roy R R. Smooth Random Mobility[M]// Handbook of Mobile Ad Hoc Networks for Mobility Models. Springer US, 2011:125-165.

[13] Ghouti L. Mobility prediction in mobile ad hoc networks using neural learning machines ☆[J]. Simulation Modelling Practice & Theory, 2016, 66:104–121.

[14] Fiedler M. Algebraic connectivity of graphs[J]. Czechoslovak mathematical journal, 1973, 23(2): 298-305.

[15] Jamakovic A, Uhlig S. On the relationship between the algebraic connectivity and graph's robustness to node and link failures[C]//Next Generation Internet Networks, 3rd EuroNGI Conference on. IEEE, 2007: 96-102.

[16] Hong Y, Zhang X D. Sharp upper and lower bounds for largest eigenvalue of the Laplacian matrices of

trees[J]. Discrete Mathematics, 2005, 296(2): 187-197.

[17] Jaromczyk J W, Toussaint G T. Relative neighborhood graphs and their relatives[J]. Proceedings of the IEEE, 1992, 80(9):1502-1517.

[18] Nistazakis H E, Karagianni E /, Tsigopoulos A D, et al. Average Capacity of Optical Wireless Communication Systems Over Atmospheric Turbulence Channels[J]. Journal of Lightwave Technology, 2009, 27(8):974-979.

# Figures

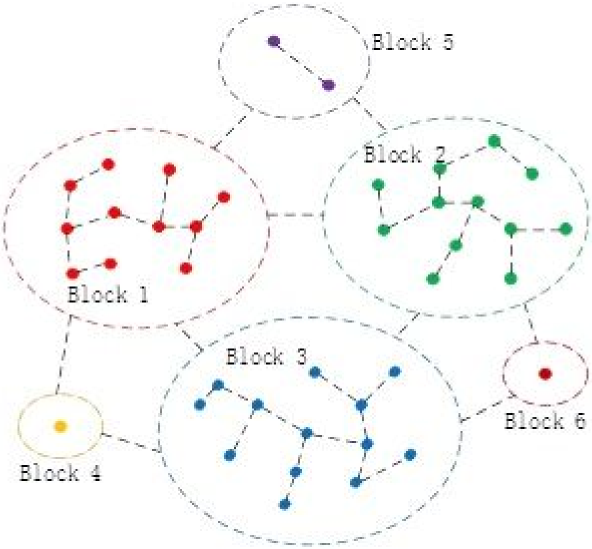



**Figure 1**

Fully distributed structure of mobile nodes in OWCNs

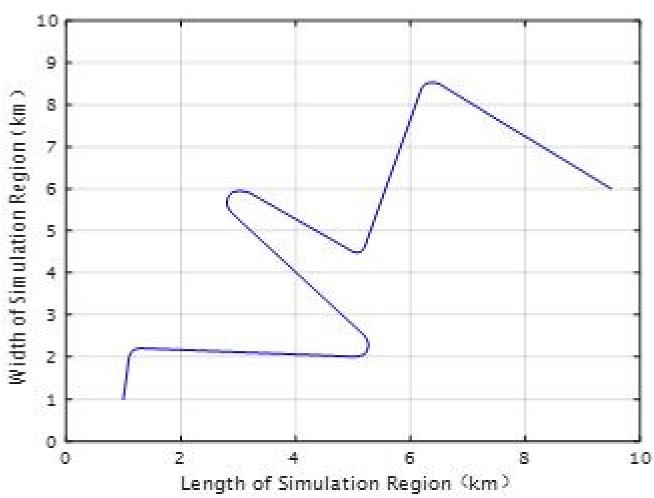



**Figure 2**

Single node moving path in SRM model

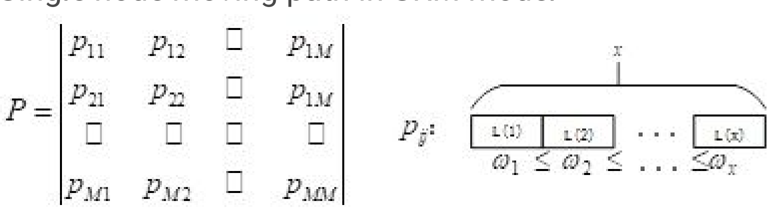
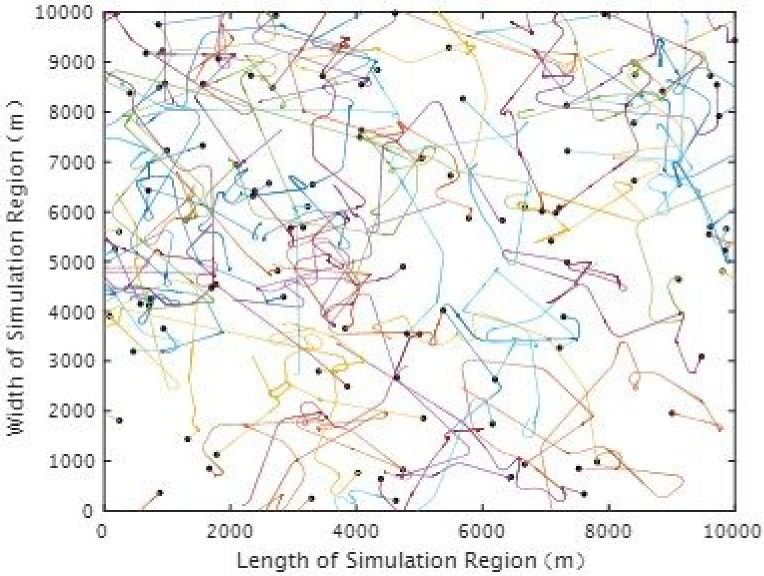


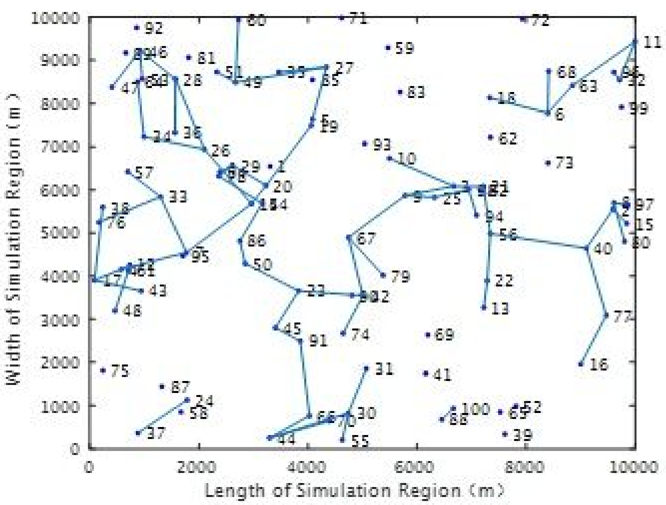
**Figure 3**

Priority queue matrixPand its priority queue pij



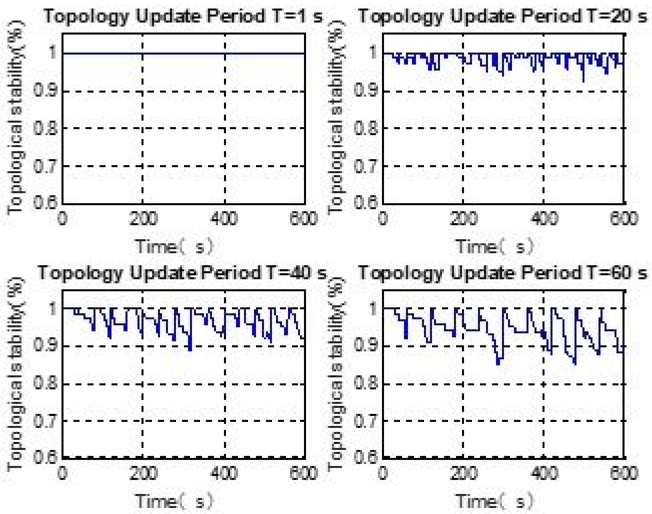

## Figure 4

Moving paths of 100 random nodes in SRM model



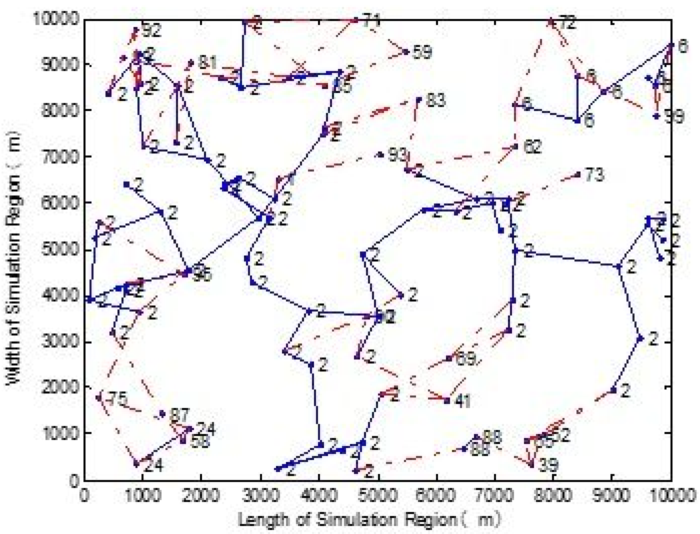

## Figure 5

Topology of node-block



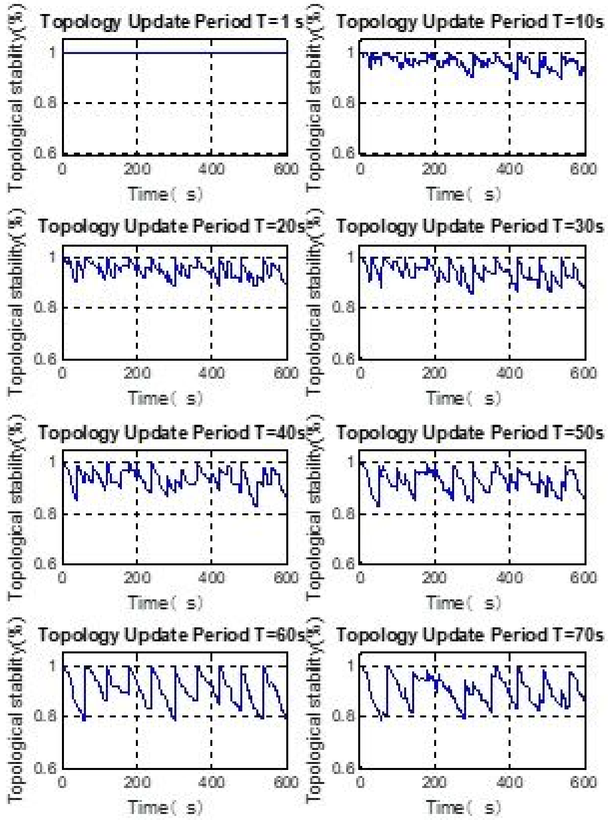

## Figure 6

Topological stability of node-blocks



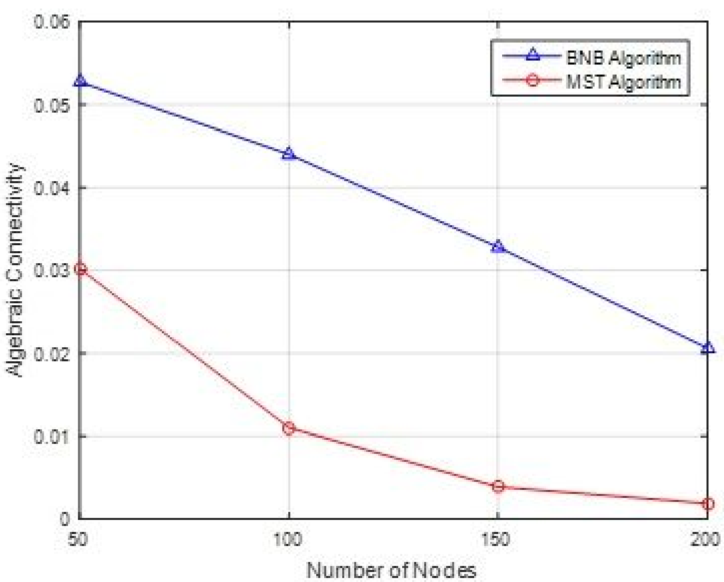

## Figure 7

Topology of connected graph




## Figure 8

Topological stability of connected graph




## Figure 9

Comparison of algebraic connectivity between NB and MST Algorithm

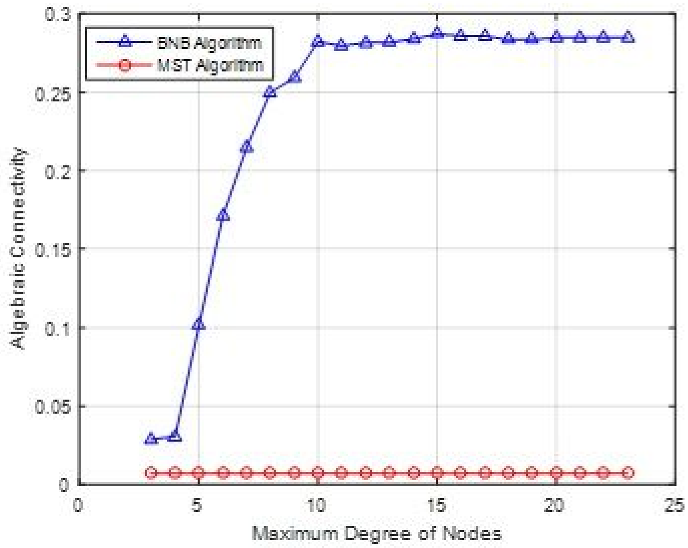



**Figure 10**

Comparison of algebraic connectivity between different node degrees

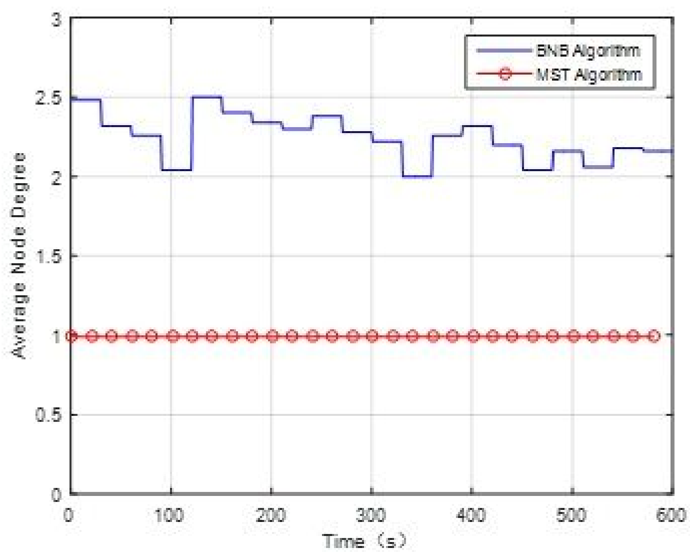



**Figure 11**

Comparison of average node degree between NB and MST algorithm