The Geodesic Broadcast Scheme for Wireless Ad Hoc Networks*

Dimitrios Katsaros[1,2] and Yannis Manolopoulos[1]
[1]Informatics Dept., Aristotle University, Thessaloniki, GREECE
[2]Computer & Communication Engineering Dept., University of Thessaly, Volos, GREECE
{dimitris,manolopo}@skyblue.csd.auth.gr

Abstract

Broadcasting is an effective means for disseminating information in wireless ad hoc networks. In this paper we propose a novel distributed broadcasting protocol in wireless ad hoc networks, which is based on an highly efficient metric for characterizing the importance of a node, with respect to its contribution in covering the local neighborhood. The protocol is reliable and achieves small communication complexity with linear in the number of nodes computation complexity. Experimental results for a large variety of network topologies show that the proposed algorithm is capable of generating small connected dominating sets, which guarantee a relatively small number of rebroadcasts.

1 Introduction

An ad hoc wireless network is a wireless mobile network in which a set of mobile nodes with wireless connectivity form a temporary network without the existence and support of any infrastructure, e.g., base stations, or centralized administration, e.g., switching centers. Communication in an ad hoc network between any two nodes that are out of one another's transmission range is achieved through intermediate nodes, which relay messages to set up a communication channel between the two nodes.

Broadcasting is an effective means for disseminating information in wireless ad hoc networks, and it forms the basis for implementing many functions, like route discovery. In a broadcasting task, a source node sends a message, which should be delivered to all the nodes in the network, if feasible. Following the mainsteam in literarture, we assume: a) a *one-to-all* broadcasting model, and b) that each communication link is bidirectional.

Straightforward implementation of broadcasting is by the use of *flooding*, but it causes the so-called *broadcast storm problem* [6]. Various approaches have been proposed in relieving this problem: probability-based methods, area-based methods and neighbor-knowledge-based methods are three large families. Probability-based and area-based methods are able to reduce the size of the forward node set, but they can not guarantee full coverage of the ad hoc network, in contrast to the methods that are based on the knowledge of the neighborhood (neighbors identity) of a node.

Neighbor-knowledge-based approaches are based on the concept of identifying a small (as possible) forward node set, which has the property that the nodes of this set form a *connected dominating set* (CDS) for the graph corresponding to the ad hoc network. Recall that a CDS of a graph is a set of nodes, such that any node of the graph either belongs to the CDS or is a neighbor of a node of the CDS. Discovery of the minimum CDS is in NP-complete. Neighbor-knowledge methods can be further categorized based on whether the forwarding status of a node is specified by a broadcasting neighbor of it [4, 2], or whether it is decided by the node itself [9, 5, 7, 1]. All aforementioned algorithms are *localized*, in the sense that they exploit only information available locally at each node.

All broadcasting algorithms proposed so far, present some weaknesses. Some methods rely on node IDs in eliminating potential redundant broadcasting nodes or in defining priorities, e.g., [9, 2, 7]. These approaches suffer from the fact that they can not detect all possible eliminations because ordering based on node id prevents this. As a consequence they incur significantly excessive retransmissions. Other methods rely on a lot of "local" information, for instance knowledge of k-hop neighborhood ($k > 2$), e.g., [8, 10]. Other methods are computationally expensive, incurring a cost of $O(f^2)$ or $O(f^3)$, where f is the maximum degree of a node of the ad hoc network. Examples of this case are the methods reported in [7, 1] and [5], respectively. Although this complexity cost does not seem prohibitive, it is quite high when moderate or high mobility turns the locally computed information (forwarding status) obsolete. Finally, some methods (e.g., [4, 5]) do not *fully* exploit the compiled information; for instance, the use of the degree of a node as its priority when deciding its possible inclusion in the dominating set might not result in the best local decision. Here, we propose a novel broadcasting protocol for ad hoc networks, which:

*Research supported by a *ΓΓΕΤ* grant in the context of the project "Data Management in Mobile Ad Hoc Networks" funded by *ΠΥΘΑΓΟΡΑΣ ΙΙ* national research program.

- is localized, thus distributed; *it can exploit 1-hop, 2-hop or k-hop neighborhood information, presenting different tradeoffs in efficiency vs. communication cost, but for the sake of readability we present it here assuming knowledge of the 2-hop neighborhood of a node*,
- introduces a new measure for capturing a node's significance in being included in the dominating set which has never been described before,
- computes a node's significance in time linear in the number of nodes and linear in the number of edges of the network neighborhood of the node, irrespectively of the degree of each node,
- computes forwarding decisions on-the-fly, thus (in combination with the above feature) is appropriate for moderate and high mobility networks.

The rest of the paper is organized as follows: In Section 2 we describe a novel metric for measuring nodes' significance. In Section 3 we present our proposed distributed broadcasting protocol. In Section 4 evaluation through simulation the proposed protocol, and finally in Section 5 we conclude the paper.

2 A new measure of node importance

Before proceeding in the presentation of the main paper ideas, we will give some necessary definitions. A wireless ad hoc network is abstracted as a graph $G(V, E)$. An edge $e = (u, v)$, $u, v \in E$ exists if and only if u is in the transmission range of v and vice versa. All links in the graph are bidirectional. The network is assumed to be in a connected state. The set of neighbors of a node v is represented by $N_1(v)$, i.e., $N_1(v) = \{u : (v, u) \in E\}$. The set of two-hop nodes of node v, i.e., the nodes which are the neighbors of node v's neighbors except for the nodes that are the neighbors of node v, is represented by $N_2(v)$, i.e., $N_2(v) = \{w : (u, w) \in E$, where $w \neq v$ and $w \notin N_1$ and $(v, u) \in E\}$. The combined set of one-hop and two-hop neighbors of v is denoted as $N_{12}(v)$.

Definition 1 (Local network view w.r.t. node v) *The local network view, denoted as LN_v, of a graph $G(V, E)$ w.r.t. a node $v \in V$ is the induced subgraph of G associated with the set of vertices in $N_{12}(v)$.*

We define a *path* from $u \in V$ to $w \in V$ as an alternating sequence of vertices and edges, beginning with u and ending with w, such that each edge connects its preceding with its succeeding vertex. The *length* of a path is the number of intervening edges. We denote by $d_G(u, w)$ the *distance* between u and w, i.e., the minimum length of any path connecting u and w in G, where by definition $d_G(v, v) = 0$, $\forall v \in V$ and $d_G(u, w) = d_G(w, u)$, $\forall u, w \in V$. Note that the distance is not related to network link costs (e.g., latency), but it is a purely abstract metric measuring the number of hops.

2.1 Measuring node significance

We mentioned in the introduction that all methods to-date use the node id or the node's degree in prioritizing the node for inclusion in the dominating set, e.g., [4, 5, 7]. Some methods first consider the node(s) which serves as the only neighbor of a node in $N_{12}(\cdot)$ and then examine the node(s) with the maximum degree w.r.t. nodes not covered yet, whereas other methods simply consider the node(s) with the highest degree. None of these approaches is appropriate because: a) the former methods treat nodes in a heterogeneous way, and b) latter methods, even though they are aware of the 2-hop neighborhood, do not make full usage of the available information. In the sequel, we will present a new definition of node's significance that avoids both drawbacks.

Let $\sigma_{uw} = \sigma_{wu}$ denote the number of shortest paths from $u \in V$ to $w \in V$ (by definition, $\sigma_{uu} = 0$). Let $\sigma_{uw}(v)$ denote the number of shortest paths from u to w that some vertex $v \in V$ lies on. Then, we define the *node importance* index $\mathcal{NI}(v)$ of a vertex v as:

Definition 2 *The node importance index $\mathcal{NI}(v)$ of a vertex v is equal to:*

$$\mathcal{NI}(v) = \sum_{u \neq v \neq w \in V} \frac{\sigma_{uw}(v)}{\sigma_{uw}}. \tag{1}$$

Large values for the $\mathcal{NI}$ index of a node v indicate that this node v can reach others on relatively short paths, or that the node v lies on considerable fractions of shortest paths connecting others. Let us see the $\mathcal{NI}$ indexes for the nodes of the graph presented in Figure 5 of [9]. The results are illustrated in our Figure 1.

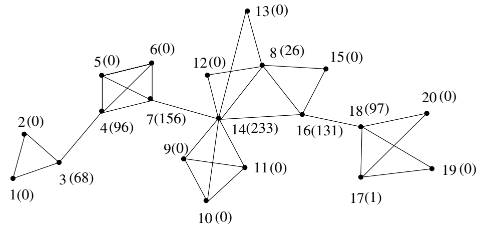

Figure 1. Calculation of $\mathcal{NI}$ for a sample graph. Each node is characterized by a pair of ID($\mathcal{NI}$).

We can easily observe the striking correspondence between the nodes with large $\mathcal{NI}$ value and those characterized as *intergateway* and *gateway* nodes, using the terminology of [5]. Thus, the $\mathcal{NI}$ index calculated over the whole graph captures structural features of the graph better than the node degree does. Moreover, it induces a *ranking* of the nodes according to their contribution in covering the whole network. Actually, the $\mathcal{NI}$ value identifies what we would call the **geodesic** nodes of the network, i.e., nodes that act as *articulation points*, or nodes with large degree relative to their neighbors.

The $\mathcal{NI}$ index would be useful in designing broadcast protocols in ad hoc networks only if it captures structural features of small graphs, e.g., of the 2-hop neighborhood of a node and only if it can be computed really fast. If these conditions hold, then it can be used in designing localized algorithms. Fortunately, they both hold. The reader can easily verify that, for any node v, the $\mathcal{NI}$ indexes of the nodes in $N_{12}(v)$ calculated only for the subgraph LN_v reveal the relative importance of the nodes in covering the subgraph N_{12} (from v's point of view). For instance, the $\mathcal{NI}$ index for the nodes belonging to LN_8 (see Figure 1) are indicated in Table 1. For a node u, which belongs to the 2-hop neighborhood of a node v (or if $u \equiv v$), the $\mathcal{NI}$ index of u (calculated over LN_v) will be denoted as $\mathcal{NI}_v(u)$.

$n(ode)$	$\mathcal{NI}_8(n)$	$n(ode)$	$\mathcal{NI}_8(n)$	$n(ode)$	$\mathcal{NI}_8(n)$
7	0	11	0	15	0
8	14	12	0	16	23
9	0	13	0	18	0
10	0	14	65		

Table 1. $\mathcal{NI}$ index of the nodes belonging to LN_8.

At a first glance, the computation of the $\mathcal{NI}$ seems expensive, i.e., $O(m*n^2)$ operations in total for a 2-hop neighborhood, which consists of n nodes and m links. Fortunately, we can do better than this by making some smart observations. For the interest of space, we will not present the details here, but direct the readers to the work [3] (which computes an index analogous to $\mathcal{NI}$ for the edges of a graph), and to the Appendix A, where we present the pseudo-code of the algorithm *CalculateNodeImportanceIndex* for the calculation of the $\mathcal{NI}$ index of a node. The algorithm is capable of handling multiple shortest paths between two nodes.

Theorem 1 *The algorithm CalculateNodeImportanceIndex is correct in the sense that, when it executes (on a whole network or a neighborhood), it correctly calculates the number of shortest paths passing through a node (of this network or the neighborhood).*

Theorem 2 *The complexity of the algorithm CalculateNodeImportanceIndex is $O(n * m)$ for a graph with n vertices and m edges.*

3 The NIB Broadcasting algorithm

In this section, we describe a localized broadcasting protocol, which exploits the $\mathcal{NI}$ index in creating the forward node set. We assume an ad hoc network in which the nodes periodically exchange with their neighbors "Hello" messages, which contain the list of their neighbors. Thus, each node is able to form a graph that corresponds to its 2-hop neighborhood. Also, each node, when it receives a packet, is able to figure out from which 1-hop neighbor this packet was sent.

Our proposed broadcasting protocol is *dynamic* or *source-dependent*, i.e., the constructed forward node set depends on the *location of the source and the progress of the broadcast process*, avoiding thus the effect of the "hot-spots". We will describe our scheme as a *neighbor-designating* one. The astute reader will understand that the $\mathcal{NI}$ ranking derived by each node regarding its neighbors, combined with a backoff procedure, allows the node to decide by its own whether to rebroadcast or not, but for simplicity we adopt the *neighbor-designating* procedure. We name the protocol as $\mathcal{NIBB}$, from the initials of the words *node importance-based broadcasting* protocol or *Geodesic* protocol.

We will state a couple of propositions without proof, which are essential for $\mathcal{NIBB}$.

Proposition 1 *The $\mathcal{NI}$ index of a node v computed over its 2-hop neighborhood is always larger than or equal to its $\mathcal{NI}$ index calculated over the 2-hop neighborhood of any other node u ($u \neq v$), where u is a 1-hop or 2-hop neighbor of v, i.e., $\mathcal{NI}_v(v) \geq \mathcal{NI}_u(v)$, $u \in N_{12}(v)$, $v \neq u$.*

Corollary 1 *If $\mathcal{NI}_v(v) = 0$, then $\mathcal{NI}_u(v) = 0$, $\forall u \in N_{12}(v)$.*

Proposition 2 *A node v is not needed to retransmit a message in achieving full network coverage, if $\mathcal{NI}_v(v) = 0$ holds.*

Proposition 2 is reminiscent of *Coverage Condition I (static)* presented in [7], but it is completely independent on any node priority, defined in an ad hoc manner, like node IDs.

We will describe the actions taken by a node v running the $\mathcal{NIBB}$ protocol, which intends to broadcast a new message (i.e., v is the *source* of the message), or which has been designated to broadcast by one of its neighbors (i.e, v is a *forwarding node*).

STEP 1. Assuming that node v has just gathered the collection of its neighbors and their neighbors by "Hello" messages, it executes *CalculateNodeImportanceIndex* over its 2-hop neighborhood graph LN_v. $\hookleftarrow$

STEP 2. Then, it runs a sorting algorithm to obtain a list of its neighbors, sorted in descending value of their $\mathcal{NI}_v(\cdot)$ index. Note that the execution of any efficient sorting algorithm, e.g., *quicksort*, does not harm the computation complexity of the broadcasting scheme, since the sorting complexity is $O(n \log n)$, where n is the cardinality of the set $N_1(v)$. $\hookleftarrow$

STEP 3. Indicate which neighbors are covered by the retransmission of v itself. $\hookleftarrow$

If node v does not have links to all the other nodes of the ad hoc network, then there exists at least on node u, such that $u \in N_{12}(v)$, but $u \notin N_1(v)$. Therefore, broadcast by v does not cover its 2-hop neighborhood. If v is the message source, it executes STEP 4a, whereas if v was designated to broadcast, it executes STEP 4b.

STEP 4a. While its 2-hop heighborhood is not covered, examine one-by-one the members of the list obtained in STEP 2. If the currently examined 1-hop neighbor u covers at least one (not covered yet) 2-hop neighbor, then designate the 1-hop neighbor as a forwarding node. Keep examining the next 1-hop neighbor of the list, till the neighborhood is covered. $\hookleftarrow$

STEP 4b. If there are any 1-hop neighbors which have already broadcast the message, then find which part of the 2-hop neighborhood is not covered yet. While this part of the 2-hop heighborhood is not covered, examine one-by-one the members of the list obtained in STEP 2 (skipping any nodes that have already broadcast). If the currently examined 1-hop neighbor u covers at least one (not covered yet) 2-hop neighbor, then designate the 1-hop neighbor as a forwarding node. Keep examining the next 1-hop neighbor of the list, till the neighborhood is covered. $\hookleftarrow$

STEP 5. Retransmit the message, augmented by the list of neighbors designated as forwarding nodes. $\hookleftarrow$

Theorem 3 *The $\mathcal{NIBB}$ algorithm is reliable, in the sense that the broadcasting packet can be disseminated to every node in the network (if it is connected).*

4 Performance Evaluation

In general, for the performance evaluation of a broadcasting algorithm, we can adopt either a "traffic-dependent" or a "traffic-independent" approach. The former assumes a realistic MAC layer with contention and/or collision, whereas the latter assumes an ideal MAC layer. The former approach focuses on the "graph-theoretic" aspects of the algorithm, i.e., in its ability to determine small connected dominating sets.

Due to space restrictions, we employ the former approach and present a detailed performance evaluation of our algorithm by developing a custom simulator, along the lines followed also by [9, 7]. We are primarily interested in measuring the size of the connected dominating set obtained, which directly reflects the *Saved Rebroadcasts (SRB)* metric used in many evaluations.

The input to the simulator is a graph representing the topology of the ad hoc network. We depart from the methodology followed for instance in [5, 7], which randomly places nodes in the plane and then creates connections between them whenever two nodes' distance is less than a transmission radius r, because *it is not realistic*, since MANETs' graphs are usually clustered. We develop such clustered graphs.

The parameters of the network topology generator are: a) gn: the number of nodes of the ad hoc network (default value: 100), b) gc: the number of clusters existing in the graph (default value: 7), c) gs: a float number which controls the relative size (in terms of number of nodes) of the clusters (default value: 0.10), d) gd: a float number depicting the fraction of graph edges relative to the edges of a complete graph with gn nodes. gd is used to control the average degree of a node (default value for average degree is equal to 7), e) $ga \in [0.5 \ldots 0.99]$: a float number depicting the fraction of edges which exist inside the clusters, relative to the total number of edges present in the graph (default value: 70%); we refer to this variable as the *assortativity* of the network.

4.1 Evaluation

As competing methods, we implemented two baseline schemes with provably moderate performance which are reported in [9], namely the basic scheme without the two rules (Rule 1 and Rule 2) indicated as *WL* and a scheme incorporating these rules indicated as *WL_1+2*. We also implemented the MultiPoint Relaying method denoted as *MPR* [4] and a high performance broadcasting algorithm [5], which was selected as a Fast Breaking Paper for October 2003, denoted as *SSZ*. All these algorithms are *reliable*. Due to space limitations, in the sequel we present a very small representative selection of the results obtained.

Impact of the number of nodes. The first experiment evaluated the impact of the number of nodes of the ad hoc network (varying parameter gn) on the size of the CDS generated. The results are depicted in Figure 2. We can easily figure out the linear dependence of the CDS size on the network size and the efficiency of the $\mathcal{NIBB}$ protocol, which always performs from 4% to 10% better than the second best performing algorithm no matter what the scale of the network is (in terms of number of nodes).

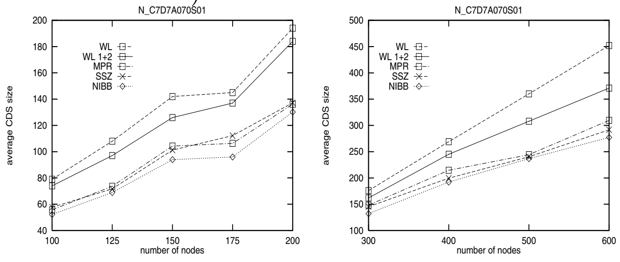

Figure 2. Impact of the nodes' number on retransmissions.

Impact of the average node degree. The second experiment evaluated the impact of the average node degree (varying parameter gd) on the size of the generated CDS. The results are depicted in Figure 3. Again, $\mathcal{NIBB}$ is the best performing algorithm, with gains up to 10% relative to the second best performing method in some cases. An interesting result obtained from this experiment is that the size of the CDS generated by *SSZ* is not always decreasing with increasing node degree, as it is implied by the results reported in [5]. The existence and number of clusters have direct impact on the performance of the algorithms. This observation is another important contribution of our work. We observe that (ignoring some statistical variation) the trend of *SSZ* to produce smaller CDS is apparent only in the case where the number of clusters is large compared to the number of nodes. In these cases, (depending on the assortativity of the network graph as well) a lot of links exist inside the clusters, thus the pruning rules (Rule 1 and Rule 2) and the large degree of the cluster nodes make the method efficient.

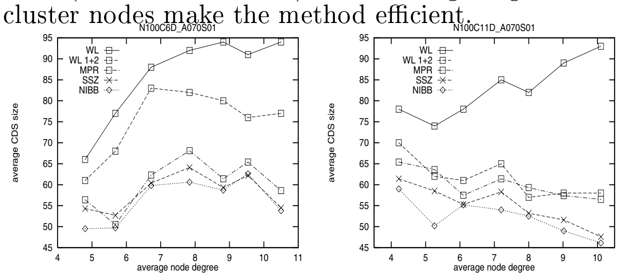

Figure 3. Impact of the average node degree on the number of retransmissions for 6 clusters (left) and 11 clusters (right).

Impact of the number of clusters. The third experiment evaluated the impact of the number of clusters (varying parameter gc) on the size of the CDS generated. The results are depicted in left part of Figure 4. The general trend is that the larger the number of clusters is (compared to the number of network nodes) the smaller the generated CDS is. This trend is followed by all methods (except from $\mathcal{NIBB}$) and it is explained by the fact that a large number of cluster implies smaller clusters with more dense linkage between the nodes of the clusters (since the density and the assortativity is the same). Therefore, it happens quite often that a single rebroadcast of a node of the cluster covers almost all the cluster nodes. $\mathcal{NIBB}$ is the most efficient algorithm and it is not affected significantly by the number of clusters, since it makes cleverer local decisions.

Impact of the strength of clusters. The fourth experiment evaluated the impact of the clusters' "strength" (ratio of intra-cluster links to total network links) on the size of the CDS generated (by varying the parameter ga). The results are depicted in the right part of Figure 4. $\mathcal{NIBB}$ exhibits an immunity on this parameter, which is a desirable feature for a broadcasting algorithm, since (ideally) we are interested in making locally optimal decisions, irrespectively of the existence or not (and number) of clusters. For the degenerate case where $ga = 0.90$, $\mathcal{NIBB}$ as well as *SSZ* and *MPR* take advantage of the well-clustered network in creating a very small forward-node set.

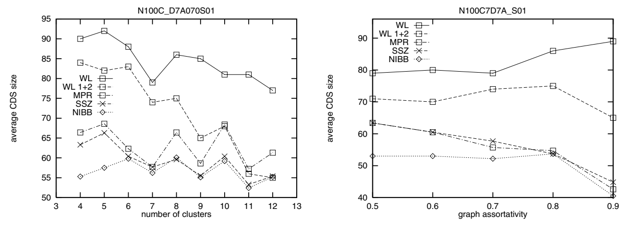

Figure 4. (Left) Impact of the clusters' number on retransmissions. (Right) Impact of the "strength" of clusters on retransmissions.

5 Conclusions and Future Work

We introduced a new reliable distributed broadcasting scheme in wireless ad hoc networks, the $\mathcal{NIBB}$ protocol. The proposed protocol is based on a novel localized metric for measuring the value of a node in "covering" the neighborhood with its rebroadcast. The calculation of this metric is linear in the number of nodes and linear in the number of radio links, thus appropriate for moderately/highly changing network topologies. This metric is itself of independent importance, and it can be used for wireless (ad hoc or sensor) network clustering and topology control. We described the broadcasting protocol as a neighbor designating method, although a self-pruning version of it is also possible. With a custom simulator, we tested the protocol's performance and the results obtained attest that the proposed protocol is very efficient and is able to reap significance performance gains in terms of saved rebroadcasts.

References

[1] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE TPDS*, 15(10):908–920, 2004.

[2] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE TMC*, 1(2):111–122, 2002.

[3] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(026113), 2004.

[4] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. Technical Report 3898, INRIA, March 2000.

[5] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE TPDS*, 13(1):14–25, 2002.

[6] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *ACM/Kluwer WINET*, 8(2–3):153–167, 2002.

[7] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. *International Journal of Foundations of Computer Science*, 14(2):201–221, 2003.

[8] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE TC*, 53(10):1343–1354, 2004.

[9] J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 18(1–3):13–36, 2001.

[10] J. Wu and W. Lou. Extended multipoint relays to determine connected dominating sets in MANETs. In *Proceedings of SECON*, pages 621–630, 2004.

A CalculateNodeImportanceIndex

```
**Algorithm** $CalculateNodeImportanceIndex$(graph $G(N, L)$)
// $N$: set of graph nodes, $L$: set of links between these nodes
// **Output**: Array $\mathcal{NI}[\cdot]$: $\mathcal{NI}$ index value for each node of $N$
begin
$\mathcal{NI}[t] = 0,\ \forall t \in N$;
foreach( $n \in N$ ) do {
    $S$: an empty stack;
    $P[\cdot]$: array of empty lists (one list $\forall$ node $w \in N$);
    $\sigma[\cdot]$: an array, where $\sigma[t] = 0, \forall t \in N$;     $\sigma[n]$=1;
    $d[\cdot]$: an array, where $d[t] = -1, \forall t \in N$;     $d[n]$=0;
    $Q$: an empty queue;
    $Q.enqueue(n)$;
    while( $Q.isNotEmpty()$ ){
        $v = Q.dequeue()$;
        $S.push(v)$;
        foreach $1$-hop neighbor $w$ of $v$ do
            // $w$ found for the first time?
                if( $d[w] < 0$ ) then
                    $Q.enqueue(w)$;
                    $d[w] = d[v] + 1$;
            // shortest path to $w$ via $v$?
                if( $d[w] == (d[v] + 1)$ ) then
                    $\sigma[w] = \sigma[w] + \sigma[v]$;
                    $P[w].append(v)$;
    }
    $\delta[\cdot]$: an array, where $\delta[t] = 0, \forall t \in N$;
    // S returns nodes in order of non-increasing hop distance from $n$
    while( $S.isNotEmpty()$ ) do
        $w = S.pop()$;
        foreach( $v \in P[w]$ ) do
            $\delta[v] = \delta[v] + \frac{\sigma[v]}{\sigma[w]} * (1 + \delta[w])$;
        if( $w \neq s$ )
            $\mathcal{NI}[w] = \mathcal{NI}[w] + \delta[w]$;
}
return $\mathcal{NI}$;
end
```