A Fault-Tolerant Routing Protocol for Mobile Ad Hoc Networks

Rana E. Ahmed
American University of Sharjah, Sharjah, United Arab Emirates
Email: rahmed@aus.edu

Abstract— **Multi-hop mobile ad hoc networks (MANETs) consist of nodes and links that are vulnerable to frequent failures. In order to provide fault-tolerance in the network, it is important that the routing protocols take into consideration the redundancy in terms of multiple paths (ideally, disjoint) from the source to the destination nodes. Dynamic source routing (DSR) is a well-known protocol commonly applied to MANETs. In this paper, a new fault-tolerant routing protocol, that modifies the basic DSR protocol, is proposed. The proposed protocol tries to find two routing paths (if they exist) from the source to the destination node. During the route discovery process, the protocol identifies several new paths that are not able to be detected by the application of the basic DSR. The proposed protocol offers low overhead over the basic DSR, in terms of the number and sizes of control messages sizes. Simulation results show that the proposed protocol also achieves better packet delivery ratio as compared to DSR.**

Index Terms— **Routing Protocols, Mobile Ad hoc networks (MANETs), Fault-tolerant routing, Dynamic source routing**

I. INTRODUCTION

A Mobile Ad hoc Network (MANET) refers to a set of wireless mobile nodes that communicate over wireless links, without any kind of fixed infrastructure or centralized controlling authority. A MANET is a peer-to-peer, multi-hop network in which all mobile node cooperate in discovering routes and forwarding packets to other nodes in the network. Route discovery is one of the most challenging tasks in MANET mainly due to lack of centralized infrastructure, frequent changes in the topology of the network, unpredictable state of the nodes and/or wireless links. Several routing protocols for the MANETs have been proposed [1-6]. There are two general categories of routing protocols in MANETs: *proactive* and *reactive*. For the *proactive (or table-driven)* protocols, the source modes have routing information available to the destination node at the time of sending the data packets; i.e., routing tables at the nodes are set up, and the tables are periodically updated to track changes in the network topology. For the *reactive (or on-demand)* protocols, the source node initiates route discovery process only when needed; i.e., the routing information to the destination node may not be available at the time of sending data packets. One popular on-demand protocol that has been extensively studied in the literature is *Dynamic Source Routing (DSR)* protocol [1-7].

Most existing ad hoc network routing protocols assume that every node in the network functions well, and the related wireless links are up and working during the data packet delivery process. In other words, the routes are stable for long period of times. However, such assumptions usually do not hold true in realistic environments. Wireless nodes crash due to several reasons including, lack of battery power, hostile environment, etc. The wireless link are broken and re-established due to noise and mobility in of the nodes. This means that a route discovered by the routing protocol may become unusable in case of node or link failures on the routing path. In the absence of fault-tolerant routing protocol, we need to re-initiate the route discovery process in order to find new route to accommodate new network topology. It is to be noted that ad hoc networks are highly redundant networks. There are generally multiple paths between the source and the destination nodes. Such network redundancy allows ad hoc networks to tolerate faulty nodes or broken wireless links. In order to enable such capability, a fault-tolerant routing algorithm needs to explore the network redundancy through some sort of multipath routing.

This paper proposes a new fault-tolerant, low-overhead routing protocol that attempts to find two routing paths (if they exist) from the source node to the destination node. The protocol is an extension of the basic DSR protocol, and it adds low overhead (in terms of control messages and the message sizes) on the DSR, and it performs better than DSR by achieving better packet delivery ratio.

The rest of the paper is organized as follows. Section 2 presents a brief overview of the problem and describes briefly the working of the DSR. Section 3 describes the proposed fault-tolerant routing protocol and compares it with the basic DSR, and section 4 presents conclusions.

doi:10.4304/jait.2.2.128-132

II. FAULT-TOLERANT ROUTING

A. Problem Definition

The design of efficient and fault-tolerant routing protocol is more crucial in MANETs. The main objective in the design of a fault-tolerant routing protocol is that if a route breaks (due to node and/or link failures), there is always (at least) one more route available for a source-destination pair. The main challenge that a fault-tolerant routing protocol should address is coping with node and/or link failures without incurring high overhead. In this paper, an existing routing protocol is selected and fault-tolerance features are introduced in the basic DSR protocol without involving high overhead.

B. Basic Dynamic Source Routing (DSR)

DSR is a well-known routing protocol used in MANETs. The basic approach used in this protocol during the route discovery phase is to establish a route by flooding Route Request (RREQ) packets in the network. In the basic DSR, when a source node does not have the route information to a destination node, and it has data packets to be sent to the destination, then it initiates the route discovery process by flooding the RREQ packets to its neighbors. Each node, upon receiving the RREQ, rebroadcasts the packet to its neighbors if it has not forwarded already or if the node is not the destination node. Each RREQ packet carries a sequence number generated by the source node and every node appends its own node ID to the traversed path information in the RREQ. A node, upon receiving a RREQ packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate RREQ packet. The destination node, after receiving *the first* RREQ packet, replies to the source node using the route replay (RREP) packet through the reverse path the RREQ packet had traversed, if all the links on the path are bi-directional. A node can also learn about the neighboring routes *by any means* and caches this route information for future use and route optimization. The route cache is also used during the route discovery phase. If an intermediate node receiving a RREQ packet has a route to the destination node in its route cache, then it replies to the source node by sending a RREP packet with the entire route information from the source node to the destination node, thus limiting the broadcast process from that node. In case of broken link, the immediate affected nodes send Route Error (RRER) packet along the route and all intermediate nodes update their cache to reflect the broken link.

III. PROPOSED PROTOCOL

A. Protocol Description

One can add fault-tolerance features to the working of the basic DSR in several ways. One possibility could be adding some known "good" and "reliable" nodes and route in the path, even at the cost of higher hop count. The proposed protocol attempts to keep at least two routes for each source and destination pair in the network, if those routes exist. The proposed protocol introduces the following modifications to the working of the basic DSR protocol to support fault-tolerant routing:

In the basic DSR, the RREP packet is sent to the source when the destination receives the first RREQ packet. In the proposed protocol, the destination node, upon receiving the first RREQ packet waits a known, short period of time, in the hope that another RREQ packet, traversing on different route, will arrive at the destination node. If there is another alternate path present between the source and the destination node, then another RREQ packet will eventually arrive at the destination; however, we need to wait only for a pre-defined time interval. The waiting time interval can be chosen based upon the topology and taking into consideration the estimate of RREQ message delay coming from a node located farthest away.

On receiving one or more RREQ packets from a source and waiting for a pre-defined period of time, the destination node can select *at most* two best routes, based upon certain known criteria, such as the number of hop count, disjoint paths, or the inclusion of certain known "good" nodes or links. The *best* route will be called the *primary route* and it will be used, by default. The other alternate route will be called the *secondary route* and it will be used only if the primary route fails. If there is only one route available between a source, destination pair, then that route will become the *primary route*.

Once the routes are selected, the destination node sends route reply (RREP) message to both primary and secondary routes. In contrast to the basic DSR protocol, the RREP message in the proposed protocol contains the routing information of the *both* routes (primary and secondary, if the two routes exist). The reason for including information about (at most) two routes (primary and secondary) is that all the intermediate nodes on those routes may use this information to optimize further the routes to other nodes.

In case the primary route for a source-destination pair fails due to node or link failure, then the secondary route (if it exists) is immediately available for use, and there is no need to initiate the route discovery process as in the case of basic DSR. The route error (RERR) message sent by the neighboring nodes can be used to delete/ update the cache routing information about the affected primary route.

B. An Example Scenario

An example is now presented to clarify the working of the proposed protocol and to compare its working with the basic DSR. Assume the network topology, as shown in Figure 1, consists of six nodes A through F with connectivity among nodes shown by solid lines. The wireless links are assumed to be bi-directional. Suppose node A needs to send data packets to node D and node A currently has no routing information to reach D. Node A starts the route discovery process by flooding the RREQ packets to its neighbor. In Figure 1, only the RREQs going forward towards the destination are shown with arrows. Before broadcasting the RREQ, a node stamps its own ID to RREQ packet (indicated with square brackets; e.g., [A,B]). Assume the first RREQ to reach the destination node D is through the path [A,B,C,D], followed by the RREQ traversing through the path [A,E,F,D]. In the case of the basic DSR, the destination node, D, send the RREP message corresponding to the first RREQ received; i.e., D will send RREP[A,B,C,D] to the source node A through the reverse path [D,C,B,A]. All the intermediate nodes on the path will update their route caches with this new routing information. In this example of basic DSR, after the routing information has been cached, if node B needs to send data packets to node F, then node B has to initiate a route discovery process as it has no routing information present for destination F in its cache.

In the proposed protocol, as shown in Figure 2, when the first RREQ packet through the path [A,B,C,D] is received by node D. The path [A,B,C,D] will be the primary path. Node D will wait for a pre-defined time interval for another RREQ to arrive via a different path. Suppose another RREQ message traversing the path [A, E, F, D] arrives shortly after at node D. The path [A,E,F,D] will be the secondary path. Now node will send the RREP message on both primary and secondary paths and it will include both paths with RREP message; i.e., node D will broadcast RREP[A,B,C,D][A,E,F,D] message towards node A using primary and secondary paths. All the intermediate nodes on both primary and the secondary paths will update their route caches with this new information. Note that using the two-path information, the nodes can discover several routes to other nodes that were not possible to detect in the case of basic DSR. For example, now node B know how to reach node F (i.e., either via path [B,C,D,F] or via [B,A,E,F]). This can be done by joining the endpoints of the primary and secondary paths; i.e., [A,B,C,D,F,E,A] in the example. All the intermediate nodes will also have both primary and secondary paths (if they exist) to other nodes, and making routing fault-tolerant.

C. Simulation Results

The proposed protocol was simulated using the Global Mobile Information System Simulator (GloMoSim) version 2.02 package [8]. The simulation parameters used were the same as used by Misra et al. in [7]. The parameters used are:

- Terrain size: 2000 m x 2000 m
- Simulation time: 15 minutes
- Number of nodes: 50
- Mobility model: Random Waypoint (RWP)
- Speed of mobile nodes: 0 to 10 m/sec.
- MAC protocol: IEEE 802.11

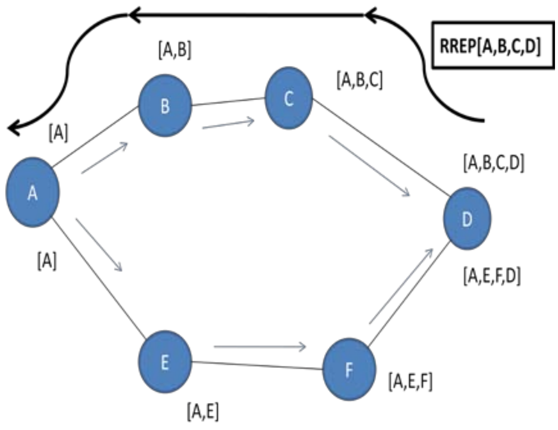

Figure 1: The basic Dynamic Source Routing (DSR) protocol operations in the example MANET.

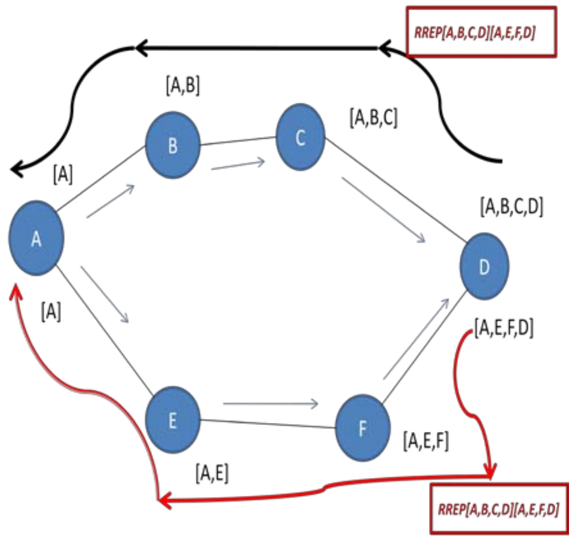

Figure 2: The proposed protocol operations in the example MANET.

Constant Bit Rate (CBR) traffic type was used in the simulation, and 1000 data items, each of size 512 bits, were simulated to be sent from a source node to the destination node, at a regular interval of 5 sec. The following performance metrics were studied through simulation: Packet Delivery Ratio (PDR); Overhead in terms of control messages sent in a session; and the network throughput.

The performance metric, Packet Delivery Ratio (PDR), is a ratio of the number of packets received by the destination to the number of packets sent by the source.

The control message overhead for the duration of an entire session for both classical DSR and the proposed protocol was compared. The control messages studied were: Route Request message (RREQ), Route Reply message (RREP), and the Route Error message (RRER).

Node mobility was simulated by varying the node pause time. Pause time is defined as the time a node remains static at a certain point. The pause time is inversely proportional to the mobility of nodes; i.e., lower the pause time, the higher the mobility of the node. The proposed protocol was compared with the classical DSR protocol, and the results are shown in Figures 3,4, and 5.

D. Comparisons and Discussions

The proposed protocol guarantees primary and secondary paths (if they exist) for a source-destination pair in the network. This will allow fault-tolerant routing in the network in case of node and/or link failures. One can argue that more than two paths could be available for a source-destination pair, but only two best routes are selected. The main reason for such selection is not to pollute route cache with data that probably will never be used.

The proposed protocol is an extension of the basic DSR for most of its operation. All the control messages, except RREP, have the same format and similar semantics. The destination node can select two *best* routes according to a pre-defined criteria. The criteria could include minimizing the hop length, disjoint paths, inclusion of a certain set of "good' and trust-worthy nodes and/ or links in the paths, etc. The criteria could address cost, fault-tolerance and security issues. The proposed protocol supports the discovery several additional routes during the route discovery process. These new routes are not possible to discover in a single route discovery phase in the basic DSR. The availability of primary and secondary routes (if they exists) for a source-destination pair increases network throughput, as the source node will not have to initiate the route discovery process on a route failure.

The simulation results, as shown in Figure 3, indicate that the proposed protocol offer better Packet Delivery Ratio as compared to DSR, in addition to providing fault-tolerance to the system. Figure 4 shows the overhead in terms of the number of control messages sent in a session between a source-destination pair for the classical DSR and the proposed protocol. The figure clearly indicates that the proposed protocol offer less overhead in terms of the number of control messages sent. In case of classical DSR, when a route fails due to a faulty node in the node, RRER message is sent to the source and the route discovery process is re-initiated by sending another RREQ and RREP messages. In the proposed protocol, this overhead is avoided at most occasions, as an alternate route is most likely to be available. The network throughput is better in case of the proposed protocol as, again, alternate route are available without the need to rediscover new routes, as is done in the classical DSR.

The proposed protocol has one disadvantage as compared to the basic DSR; i.e., the RREP message has more information (for both primary and secondary paths) to carry, while basic DSR carries information about primary path only.

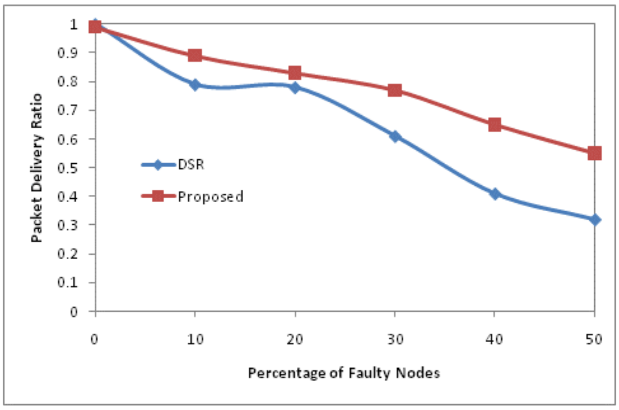

Figure 3: Simulation results for Packet Delivery Ratio vs. Percentage of Faulty Nodes for the proposed protocol and the classical DSR. Pause time= 50 sec.

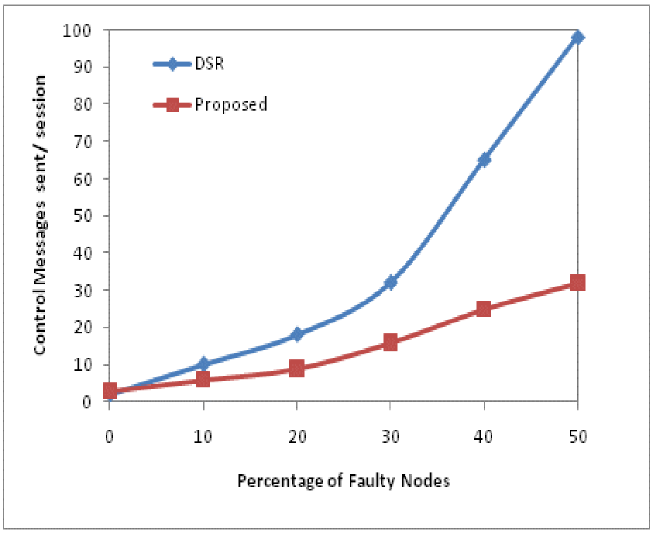

Figure 4: Simulation results showing the overhead in terms of number of control messages sent per session for the proposed protocol and the classical DSR. Pause time= 50 sec.

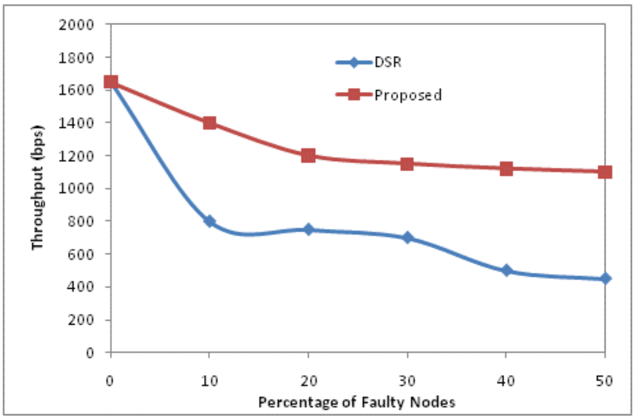

Figure 5: Simulation results showing the throughput changes as a result of increasing the percentage of faulty nodes for the proposed protocol and the classical DSR. Pause time= 50 sec.

IV. CONCLUSIONS

Design of a fault-tolerant routing protocol for mobile ad hoc network is a complex process. If a source node does not have an alternate route to the destination node, then the source has to initiate a route discovery process on the failure of a node and/or link in the original path. The route discovery process has its own latency and it limits the network throughput. This paper proposes a fault-tolerant routing protocol that identifies two routing paths (if they exist) by making some minor modification to the basic DSR protocol. The protocol supports the discovery several additional routes during the route discovery process. These new routes are not possible to discover in a single route discovery phase in the basic DSR. The working of the protocol is compared with that of basic DSR and it is found that protocol offers low overhead, in terms of number of control messages, as compared to the basic DSR. Simulation result also indicate higher packet delivery ratio, higher network throughput, and less control message overhead as compared to the classical DSR protocol.

REFERENCES

[1] D.B. Johnson, D.A. Maltz, and J. Broch, ``DSR: The dynamic source routing protocol for multihop wireless ad hoc networks," *Ad Hoc Networking*, pp. 139–172, 2001.
[2] A. Nasipuri, R.Castaneda, and S.R. Das, ``Performance of multipath routing for on-demand protocols in mobile ad hoc networks," ACM/Kluwer Mobile Networks and Applications (MONET), vol. 6, no. 4, pp. 339–349, 2001
[3] C.E. Perkins and E.M. Royer, ``Ad hoc on-demand distance vector routing," Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), pp. 90–100, 1999.
[4] D. Johnson, D. Maltz and Y. Hu. The dynamic source routing protocol for mobile ad hoc networks. IETF MANET Working Group, Internet Draft 2003.
[5] E. Royer and C. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," IEEE Personal Communications Magazine, pp. 46–55, 1999.
[6] Y. Xue and K. Nahrstedt, "Providing Fault-Tolerant Ad hoc Routing Service in Adversarial Environments ", Wireless Personal Communications 29: pages 367–388, 2004.
[7] S. Misra, S. Dhurandher, M. Obaidat, K. Verma, and P. Gupta, "A low-overhead fault-tolerant routing algorithm for mobile ad hoc networks: A scheme and its simulation analysis", Simulation Modelling: Practice and Theory 18: pages 637-649, 2010.
[8] GloMoSim Simulator: Available at http://www.pcl.cs.ucla.edu/projects/glomosim.

Call for Papers and Special Issues

Aims and Scope

JAIT is intended to reflect new directions of research and report latest advances. It is a platform for rapid dissemination of high quality research / application / work-in-progress articles on IT solutions for managing challenges and problems within the highlighted scope. JAIT encourages a multidisciplinary approach towards solving problems by harnessing the power of IT in the following areas:

- **Healthcare and Biomedicine** - advances in healthcare and biomedicine e.g. for fighting impending dangerous diseases - using IT to model transmission patterns and effective management of patients' records; expert systems to help diagnosis, etc.
- **Environmental Management** - climate change management, environmental impacts of events such as rapid urbanization and mass migration, air and water pollution (e.g. flow patterns of water or airborne pollutants), deforestation (e.g. processing and management of satellite imagery), depletion of natural resources, exploration of resources (e.g. using geographic information system analysis).
- **Popularization of Ubiquitous Computing** - foraging for computing / communication resources on the move (e.g. vehicular technology), smart / 'aware' environments, security and privacy in these contexts; human-centric computing; possible legal and social implications.
- **Commercial, Industrial and Governmental Applications** - how to use knowledge discovery to help improve productivity, resource management, day-to-day operations, decision support, deployment of human expertise, etc. Best practices in e-commerce, e-commerce, e-government, IT in construction/large project management, IT in agriculture (to improve crop yields and supply chain management), IT in business administration and enterprise computing, etc. with potential for cross-fertilization.
- **Social and Demographic Changes** - provide IT solutions that can help policy makers plan and manage issues such as rapid urbanization, mass internal migration (from rural to urban environments), graying populations, etc.
- **IT in Education and Entertainment** - complete end-to-end IT solutions for students of different abilities to learn better; best practices in e-learning; personalized tutoring systems. IT solutions for storage, indexing, retrieval and distribution of multimedia data for the film and music industry; virtual / augmented reality for entertainment purposes; restoration and management of old film/music archives.
- **Law and Order** - using IT to coordinate different law enforcement agencies' efforts so as to give them an edge over criminals and terrorists; effective and secure sharing of intelligence across national and international agencies; using IT to combat corrupt practices and commercial crimes such as frauds, rogue/unauthorized trading activities and accounting irregularities; traffic flow management and crowd control.

The main focus of the journal is on technical aspects (e.g. data mining, parallel computing, artificial intelligence, image processing (e.g. satellite imagery), video sequence analysis (e.g. surveillance video), predictive models, etc.), although a small element of social implications/issues could be allowed to put the technical aspects into perspective. In particular, we encourage a multidisciplinary / convergent approach based on the following broadly based branches of computer science for the application areas highlighted above:

Special Issue Guidelines

Special issues feature specifically aimed and targeted topics of interest contributed by authors responding to a particular Call for Papers or by invitation, edited by guest editor(s). We encourage you to submit proposals for creating special issues in areas that are of interest to the Journal. Preference will be given to proposals that cover some unique aspect of the technology and ones that include subjects that are timely and useful to the readers of the Journal. A Special Issue is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

The following information should be included as part of the proposal:

- Proposed title for the Special Issue
- Description of the topic area to be focused upon and justification
- Review process for the selection and rejection of papers.
- Name, contact, position, affiliation, and biography of the Guest Editor(s)
- List of potential reviewers
- Potential authors to the issue
- Tentative time-table for the call for papers and reviews

If a proposal is accepted, the guest editor will be responsible for:

- Preparing the "Call for Papers" to be included on the Journal's Web site.
- Distribution of the Call for Papers broadly to various mailing lists and sites.
- Getting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Instructions for Authors.
- Providing us the completed and approved final versions of the papers formatted in the Journal's style, together with all authors' contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

Special Issue for a Conference/Workshop

A special issue for a Conference/Workshop is usually released in association with the committee members of the Conference/Workshop like general chairs and/or program chairs who are appointed as the Guest Editors of the Special Issue. Special Issue for a Conference/Workshop is typically made of 10 to 15 papers, with each paper 8 to 12 pages of length.

Guest Editors are involved in the following steps in guest-editing a Special Issue based on a Conference/Workshop:

- Selecting a Title for the Special Issue, e.g. "Special Issue: Selected Best Papers of XYZ Conference".
- Sending us a formal "Letter of Intent" for the Special Issue.
- Creating a "Call for Papers" for the Special Issue, posting it on the conference web site, and publicizing it to the conference attendees. Information about the Journal and Academy Publisher can be included in the Call for Papers.
- Establishing criteria for paper selection/rejections. The papers can be nominated based on multiple criteria, e.g. rank in review process plus the evaluation from the Session Chairs and the feedback from the Conference attendees.
- Selecting and inviting submissions, arranging review process, making decisions, and carrying out all correspondence with the authors. Authors should be informed the Author Instructions. Usually, the Proceedings manuscripts should be expanded and enhanced.
- Providing us the completed and approved final versions of the papers formatted in the Journal's style, together with all authors' contact information.
- Writing a one- or two-page introductory editorial to be published in the Special Issue.

More information is available on the web site at http://www.academypublisher.com/jait/.