논문 08-33-08-10 한국통신학회논문지 '08-08 Vol. 33 No. 8

Improving TCP Performance Over Mobile ad hoc Networks by Exploiting Cluster-Label-based Routing for Backbone Networks

Vitaly Li*, Jae Yeol Ha** ***Regular Members***,
Hoon Oh***, Hong Seong Park* ***Lifelong Members***

ABSTRACT

The performance of a TCP protocol on MANETs has been studied in a numerous researches. One of the significant reasons of TCP performance degradation on MANETs is inability to distinguish between packet losses due to congestion from those caused by nodes mobility and as consequence broken routes. This paper presents the Cluster-Label-based Routing (CLR) protocol that is an attempt to compensate source of TCP problems on MANETs – multi-hop mobile environment. By utilizing Cluster-Label-based mechanism for Backbone, the CLR is able to concentrate on detection and compensation of movement of a destination node. The proposed protocol provides better goodput and delay performance than standardized protocols especially in cases of large network size and/or high mobility rate.

Key Words : TCP performance, Routing, MANETs, Cluster-label, Backbone

Ⅰ. 서 론

Mobile ad hoc networks (MANETs) are formed by the wireless mobile nodes communicating with each other without presence of fixed infrastructure. The nodes communicate directly while in transmission range of each other and use routing algorithms for multi-hop communications. Every node could function as an end system, a router or both of them at a time where end system nodes send or receive data and router nodes forward data. The IETF MANET working group [1] has standardized OLSR (Optimized Link State Routing) [2], AODV (Ad hoc On-demand Distance Vector Routing Protocol) [3] and DSR (Dynamic Source Routing) [4], where first is proactive and last two are reactive routing protocols respectively. In proactive protocols like OLSR or DSDV (Distributed-Sequenced Distance Vector Routing Protocol) [5], all nodes should maintain its routing table to all possible destinations regardless of actual needs for the route between source and destination nodes. In reactive protocols such as AODV or DSR, the route is obtained by source node in on-demand manner only when there is a data to send.

In addition to the above mentioned network layer protocols, a transport layer protocol such as TCP (Transmission Control Protocol) is also needed for reliable data communications. As the TCP protocol is most widely used for current Internet applications it is well tuned for handling wired connections. The TCP attempts to determine the optimal available bandwidth using congestion control mechanisms such as slow-start and AIMD (additive increase and multiplicative decrease). The packets loss is used as congestion indication forcing to decrease the congestion window. However, in an environment presented by MANETs packet loss due to broken routes can result in the counter-productive invocation of TCP's congestion control mechanism thus leading to underutilization of bandwidth and reducing effectiveness of protocol.

※ 본 연구는 2007년도 강원대학교 학술연구조성비로 수행되었습니다.
* 강원대학교 제어계측과, 산업정보통신 연구실({vitaly, hspark}@control.kangwon.ac.kr),
** (주) 엠젠 개발 2팀(hjy8099@mzen.co.kr), *** 울산대학교 컴퓨터정보통신학과, 모바일 및 유비쿼터스 컴퓨팅 연구실
논문번호 : KICS2008-04-190, 접수일자 : 2008년 4월 28일, 최종논문접수일자 : 2008년 7월 30일

There have been a number of studies that address that problem in literature [6-11].

It has been shown that the mobility of nodes causes most degradation of TCP performance in MANETs [6-7, 9, 11].In addition to the problems inherited from wireless networks, the multi-hop environment in a presents of mobility poses new problems. This paper is focused on a routing stability problem. Since the route in a MANET is a set of links between nodes connecting source and destination at any given time any of the nodes participating in a data routing could move out causing the packet loss and consequently the performance degradation. The common approach for solving mobility problem is to use mechanisms to (a) detect and distinguish link failures, and (b) initiate the proper response. Examples of such schemes include Explicit Link Failure Notification (ELFN) [6] and TCP-F [12], where the intermediate node detects packet loss and sends explicit notification to the source node so that the sender can distinguish between route failures and congestion, and initiate a proper response. However, the approach has problem of scalability in case of large or dense network with heavy traffic.

This paper proposes a routing protocol called Cluster-Label-based Routing Protocol (CLR) for improving TCP performance over MANET.

The CLR is built on top of the backbone created by the topology control scheme called Cluster-Label-based mechanism for Backbone (CLaB) [14]. The CLaB constructs and constantly maintains the overlay infrastructure based on interconnected clusters. Each cluster is assigned with unique identifier called Cluster-Label and maintenance algorithm provides constant connectivity between Cluster-Labels. The CLR establish path from source to destination utilizing the cluster-labels rather than nodes IDs. Using such approach leaves the problem of intermediate nodes movement to the CLaB and remain problems of destination node movement and route recalculation. In order to cope with these problems the destination-initiated movement notification is used and cluster-labels routing tables is constructed using proactive messages exchanges.

The rest of the paper is organized as follows. Section Ⅱ gives an overview of Cluster-Label-based mechanism for Backbone. Section Ⅲ describes the Cluster-Label-based Routing protocol. Section Ⅳ shows performance evaluations and Section Ⅴ concludes the paper.

Ⅱ. An Overview of Cluster-Label-based mechanism for Backbone

The Cluster-label-based mechanism for Backbones (CLaB) forms and maintains backbone over an ad hoc network. The following definitions are used.

Definition 1: A cluster is a set of nodes with a central node called clusterhead. Any node in a cluster can communicate with a clusterhead directly.

Definition 2: A host cluster X for a node N is a cluster such that the N and the clusterhead of X can communicate directly.

Definition 3: Two clusters are overlapping if there is at least one node that can communicate directly with clusterheads of both clusters.

The goal of the CLaB is to create clusters and maintain connections between them in case of nodes movement. In addition unique identifier called Cluster-Label is assigned to each newly created cluster. Hence, a node can be found in a network by the Cluster-Label of a host cluster.

The creation and maintenance of a clustered network is based on a periodic exchange of local information between neighbor nodes. Nodes periodically exchange "Hello" packets with information about itself that defines node's unique ID, status and host cluster related information. The status can be a clusterhead, a member or orphan which corresponds to neither clusterhead nor member. Each orphan node should participate in an election of a clusterhead within transmission range. Once the clusterhead is selected every node in its transmission range silently becomes a member of a newly created cluster and thus is not eligible to participate in a further clusterheads election. If node does not hear from any clusterhead for a three "Hello"packet periods it changes its status to orphan and starts the clusterhead election process again. For the simplicity the lowest ID clustering algorithms [14]is used throughout the paper. However, it is worth noting that any clustering algorithm can be used as long as it does not avoid overlapping clusters and can be executed locally.

For each newly created cluster the unique Cluster-Label is generated and spread among cluster members by clusterhead. Once the Cluster-Label is propagated by clusterhead the formation part is complete and the continuous maintenance part takes place.

Each node n by receiving "Hello" message from neighbor nodes collects and keeps following information: l(n) is a set of neighbors of node n and c(n) is a set of cluster labels of host clusters for node n. Then

$$p(n) = \bigcup_{\forall i \in l(n)} c(i)$$

is a set of cluster labels whose clusterheads are residing within two hops range from node n. The two hops restriction is set in order to track overlapping clusters only. The "Hello"message from node n therefore contains ID, information about current state, c(n) and p(n). The clusterhead should indicate its cluster by setting additional field.

The following relationship is defined

$$\begin{cases} p(a) > p(b) \text{ if } p(b) \subset p(a) \\ p(a) = p(b) \text{ if } p(b) \subset p(a) \vee p(a) \subset p(b) \end{cases}$$

On receiving the "Hello"message from a clusterhead each node n should check whether p(n)>p(CH). Here notation of CH is used to substitute ID of particular clusterhead. Therefore the p(CH) is a p(X) where X is an ID of clusterhead. If node Y finds that p(Y)>p(CH), the node Y should automatically set its status to clusterhead and start propagating the "Hello" message. No confirmation is required from a previous clusterhead. The decision is based simply on comparison of local p(n) versus one received from a clusterhead.

Since the maintenance part cannot assure that clusterheads of neighbor clusters are two hops away from each other the newly selected clusterhead should discover and keep set of three hops away connections with neighbor clusters. Let's denote list of three hops away clusterheads as p3(CH). Then

$$p3(CH) = \left(\bigcup_{\forall i \in l(CH)} \right) - p(CH)$$

Both p(CH) and p3(CH) are used for communications with neighbor clusters.

From the procedure described above, the following definition of a cluster-label and a clusterhead can be stated:

Definition 4: A cluster-label is an unique identifier of a group of nodes with a single node carrying the role of a clusterhead. Once created the cluster-label is further defined by neighbor cluster-labels.

The definition 4 is essential in a sense that it not only defines the cluster-label but the mechanism of preventing the change of it. Since the maintenance of a cluster-label depends on cluster-labels of neighbor clusters the group of adjacent clusters forms protection against nodes mobility.

In general, the maintenance part takes place between cluster members and the current clusterhead sharing same cluster label. The clusterhead replacement is based on ability to support existing connections with neighbor clusters and then the affect of a clusterhead replacement is minimal from the point of view of the backbone. Since both formation part and maintenance part relies on periodical exchange of "Hello" packets between one hop neighbors which is similar to most routing protocols, the energy consumption does not exceed energy consumptions presented by previous routing protocols.

Ⅲ. Cluster-Label-based Routing Protocol

The main purpose of routing is to provide stable connection between source and destination despite node movement. The Cluster-Label-based Routing Protocol (CLR) is to utilize the advantages given by CLaB mechanism.

Each node in network is defined by its unique identifier called node's ID and its location with regard to the backbone created by CLaB. Therefore the task of CLR is to find the initial location of a node and track its movement through clusters.

The discovery part uses reactive approach. Whenever source node needs to acquire the route to destination it issues the Route Request (RREQ) packet to its host cluster's clusterhead in the following form: {sourceID, destID}, in which sourceID and destID are IDs of source and destination nodes, respectively. The clusterhead then forwards the RREQ to clusterheads of neighbor clusters. The process is repeated for each receiver clusterhead, until the cluster of the destination node is found. The difference from traditional reactive routing algorithms is that each clusterhead attach its cluster label instead of node's ID to RREQ prior to forwarding. None of other forwarding nodes (gateways) can change the RREQ. Therefore, upon receiving the RREQ, the destination node sends RREP back to source node using reversed route in the following form: {sourceID, sourceCL, CL1, CL2, ⋯, destCL, destID}, in which sourceCL and destCL are the cluster labels of source's and destination's host clusters, respectively; and CL1, CL2, ⋯ are cluster labels of intermediate clusters. In essence the route discovery is a form of controlled flooding algorithm where the RREQ message is propagated to all network through forwarding from one cluster to another. Every clusterhead propagates only first received RREQ message for a particular source/destination pair therefore the problem of uncontrolled flooding often presented in reactive routing protocols is absent.

The route maintenance part divided into two separate algorithms: notification and route re-calculation. Each clusterhead is responsible for propagating its p(CH) to the network triggered by permanent addition or deletion of records from p(CH).On receiving such information from all clusters each node constructs the cluster-labels routing table and is able to calculate the route to any cluster in terms of cluster-labels.

At any time a node X is able to associate itself with a cluster-label A if it receives "Hello" messages from clusterhead of a cluster A. Therefore, node X also able to detect if current cluster-label is changed. Since such movements present a problem for validity of active routes, it is necessary to provide mechanism for detection of node movements by both sender and receiver. It is done using RCHG message. The RCHG message is issued by a node that detects change of associated cluster-label. The payload of the RCHG is {ID, CL}, where ID is unique identifier of sender and CL is current associated cluster-label. For every destination in routing table the new route is recalculated using last two entries in a routing sequence (destCL, destID) and cluster-labels routing table described in paragraph above. Then, RCHG is sent to every destination using updated routing entries. On receiving the RCHG a node should check whether the sender is in the active routing table. If not, the RCHG is ignored. Otherwise, the route is recalculated using cluster-labels routing table. The recalculation is required since the movement of a source or destination node can be quite chaotic and simple addition of a current host cluster's cluster-label could lead to longer route causing delay and throughput degradation.

In case of route failure due to simultaneous movement of source and destination node or packet loss the RREQ is issued again following standard routing protocols routine. However, any of suggested techniques for improvement of TCP performance such as ELFN[13] could be used in conjunction with CLR.

It is worth noting that the CLR uses both reactive and proactive approaches. Potentially this may lead to the high overhead comparing to other routing protocols. However, the route discovery uses RREQ broadcasting through clusterheads only, which is well known technique to reduce the control overhead. On the other hand the proactive approach uses event-triggered directed transmission. Since the CLaB takes care of connections between neighbor clusters such events occurs rarely comparing with that in proactive protocols such as DSDV or OLSR. Also, since the information about cluster-labels (group of nodes) is propagated rather than nodes, the amount of data is quite small. The same holds true in regard to the computation power consumption and amount of memory resources necessary for keeping the cluster-labels routing table. It can be argued that the destination-initiated notification could be harmful for the UDP connections since the destination often is not aware about route to the source node and therefore cannot send the RCHG. In such a case the route recovery could be done in a standard manner by re-issuing RREQ message from a source node. However, in this case the RREQ could be unicasted to the previous host cluster of a destination node to be broadcasted from there which can greatly increase the speed of route discovery. In previous work [14] it has been suggested to use previous cluster-head for packets forwarding and RCHG notifications in a way similar to one often used in the handoff procedure. However, such approach does not work well for slow data rate, where cluster-head could be changed before next packet arrives.

Ⅳ. Performance evaluation

4.1 Simulation Model

All simulations are implemented using ns2 network simulator [15]. The common simulation parameters are presented in Table 1

AODV, DSR and OLSR were used as targets for comparison. All three protocols are standardized by IETF MANET working group. First two protocols use reactive approach for routing discovery and maintenance. The route is obtained on-demand when needed and dropped if not used. The OLSR uses proactive approach meaning that regardless of actual needs for a route the routing table on each node contains routes to all possible destinations in a network. The information is updated through periodic data exchanges with neighbors.

Table 1. Simulation Parameters.

Item	Value
Simulation Time	1200s
Maximum Link Bandwidth	2Mbps
Number of Nodes	200
Mobility Model	Random-waypoint
Pause Time	30s
Maximum Number of TCP Connections	40
TCP Version	NewReno
TCP Window Size	32
Packet Size	1460bytes
Transmission Range	250m

4.2 Accuracy of simulations

Presented results had been obtained from the simulation on NS2 network simulator. In order to strengthen obtained results the average of 20 simulations is presented below. The parameters of a network for corresponding points in a graph are same while initial distribution of nodes is random across the network area. TCP connections between nodes are fixed, i.e. node X always establishesconnection with node Y throughout a simulation set. The CLaB algorithm starts immediately after the start of simulation while

first attempt to establish TCP connection issued 5 seconds after the simulation starts. Since the initial placement of nodes differs with each simulation run the connection parameters such as number of hops between connection participants differ as well.

Another issue is the choice and accuracy of a simulation tool and hence the accuracy of obtained results itself. There are three major simulation tools widely used by research community: NS2, Glomosim [17] (currently succeeded by QualNet [18]) and OPNET Modeler [16]. NS2 and Glomosim are free tools while QualNet and OPNET Modeler are available under commercial license. The wide support and a free license made NS2 network simulator a de facto

standard for network simulations within academia.

It has been argued [19] that network simulators often fail to represent realistic model of network and especially MANETs leading to inaccurate results. In particular it is often difficult or impossible to configure realistic simulation and environment parameters such as terrain relief, radio propagation model, climate, mobility model. From the other hand the network simulators could present fair general network model within certain boundaries. Since the real evaluations on a testbed are very expensive in a case of MANETs the NS2 remains primary research tool for initial discovery of a network performance.

4.3 Simulation Results

The first set of simulation results (Figures 1-4) presents result of average route lifetime in relation to the network size. Since movement of any intermediate node is the primary cause of route failure for reactive routing algorithms and does not particulary affect proactive routing algorithms the Figures 1-4 presents three routing protocols. Mobility rate affect the time that direct link between two adjacent nodes is kept and effect of a network size is a possible maximum number of intermediate nodes in a route. Figure 1 shows that route lifetime in case of CLR is more than 4 times longer in case of small network size than that of the AODV and DSR in case of small speed. With increase of a network size the gap between CLR and compared protocols is increase

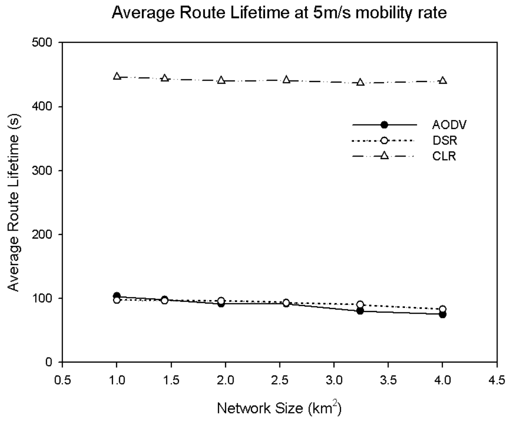

Fig. 1. Average Route Lifetime vs network size for mobility rate 5m/s

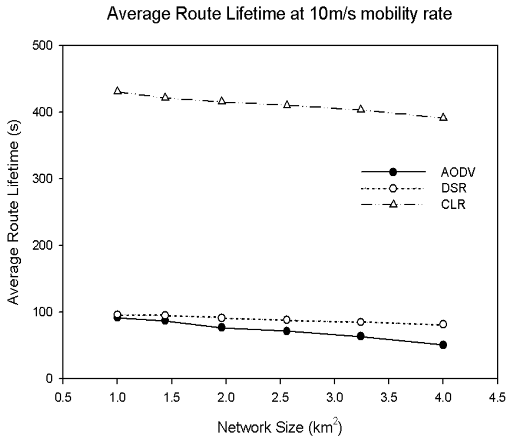

Fig. 2. Average Route Lifetime vs network size for mobility rate 10m/s

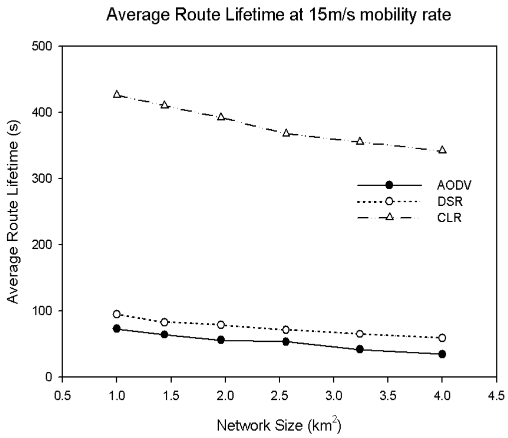

Fig. 3. Average Route Lifetime vs network size for mobility rate 15m/s

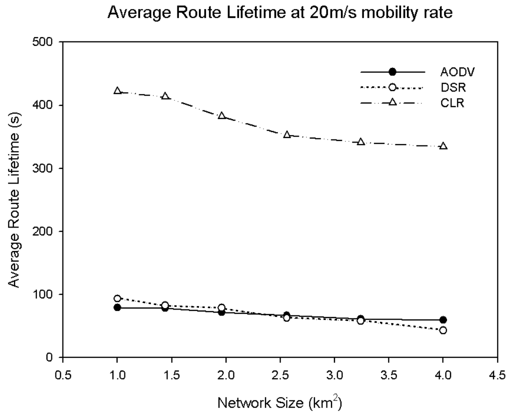

Fig. 4. Average Route Lifetime vs network size for mobility rate 20m/s

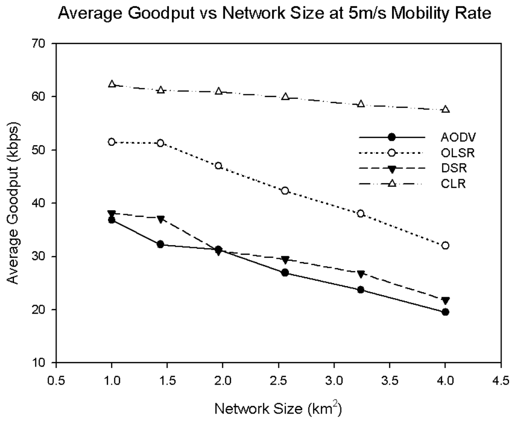

Fig. 5. Average Route Lifetime vs network size for mobility rate 5m/s

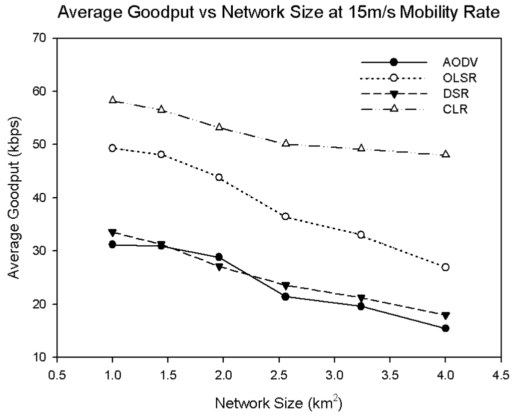

Fig. 7. Average Route Lifetime vs network size for mobility rate 15m/s

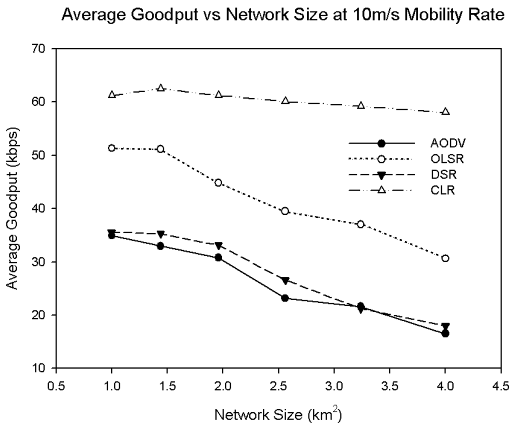

Fig. 6. Average Route Lifetime vs network size for mobility rate 10m/s

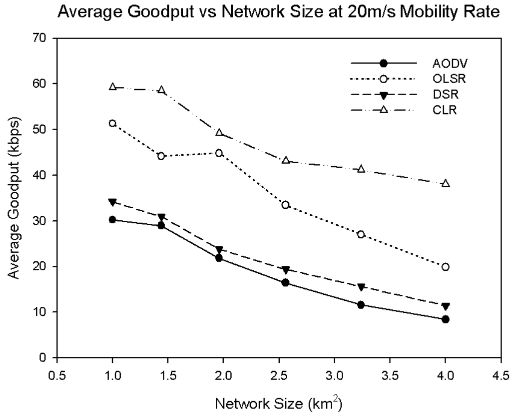

Fig. 8. Average Route Lifetime vs network size for mobility rate 20m/s

drastically. Note that route caching exploited by DSR helps to slow the decrease of route lifetime in case of slow mobility rate. Figures 2-3 shows similar pattern. The route lifetime of CLR slightly decreases with increase mobility. However it still much longer than route lifetimes of compared protocols. Figure 4 shows case of high mobility rate of 20 m/s. In this case CLR also performs much better than AODV or DSR. However it is worth noting that caching mechanism of DSR plays against it in case of large network size. This occurs because of invalidation of caching due to high mobility and network size.

The second set of simulation results (Figures 5-8) shows average goodput achieved using AODV, DSR, OLSR and CLR versus network size with varying mobility rate. The route length has indirect effect to the goodput. Once the number of intermediate nodes becomes bigger the risk of route failure increases. It does not affect proactive routing protocols as much as it affect reactive routing. However, it is shown that performance of all protocols drops as the network area increases. Among AODV, DSR and OLSR protocols the proactive algorithm works better. As could be expected from route lifetime simulations (Figures 1-4) CLR protocols outperforms both DSR and AODV with a big margin. However, while CLR could be considered reactive routing protocol it shows better results than OLSR. This

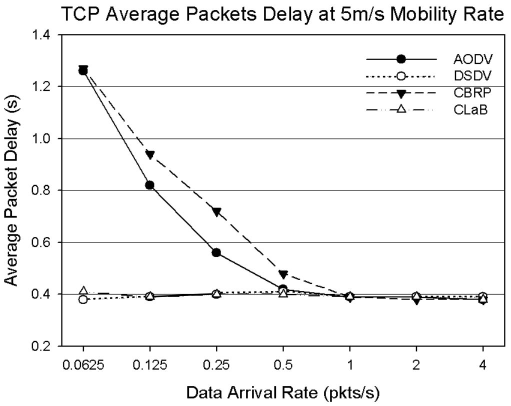

Fig. 9. Average Packets Delay vs Packets Arrival Rate for mobility rate 5m/s

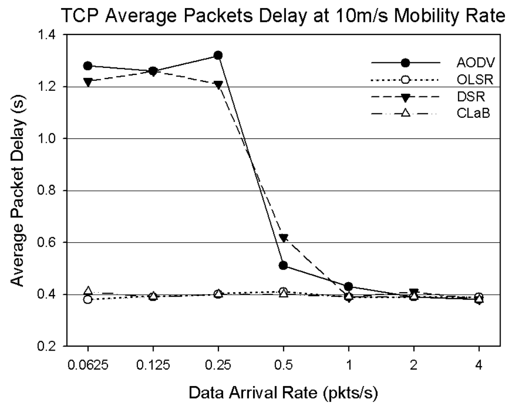

Fig. 10. Average Packets Delay vs Packets Arrival Rate for mobility rate 10m/s

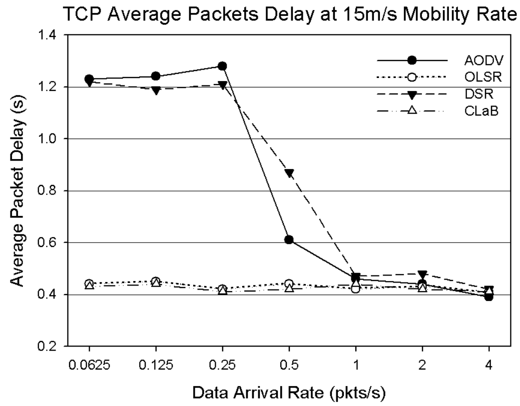

Fig. 11. Average Packets Delay vs Packets Arrival Rate for mobility rate 15m/s

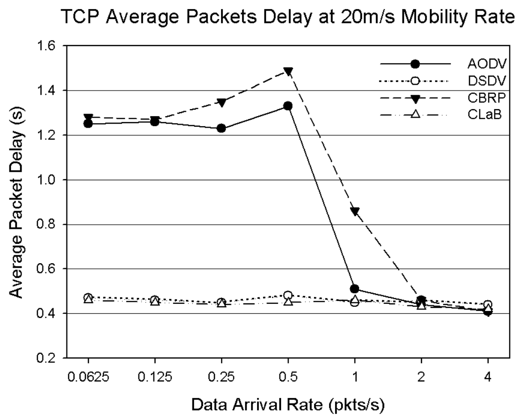

Fig. 12. Average Packets Delay vs Packets Arrival Rate for mobility rate 20m/s

is due to the high routing messages overhead caused by OLSR and the numerous layer 2 contentions and congestion periods that caused by this overhead. As oppose to that the CLR routing overhead consists only of relative information while allowing to avoid route failures.

The third set of simulations shows the delay performance of a 1km2 case with varying average packet arrival rate. The typical simulation scenarios consider high packet arrival rate. Therefore for such simulations a throughput or goodput is appropriate metric. However, some TCP applications such as interactive communications, chatting, telnet, etc. issue a data at much slower rate and require a low per-packet delay rather than high throughput. Figures 9-12 shows the average packet delay in a TCP flow. The average packet delay is calculated as an interval between packet arrival at the source node and receiving acknowledgement message from a destination node.

It is shown that for low speed corresponding to 5m/s case the average TCP packet delay is similar for all protocols while the data arrival rate is high enough so that reactive routing protocols can detect the broken route. In case the data arrival rate is low, the reactive protocols do not detect a broken route until after the route error notification. However in case of OLSR and CLR the route update is independent of data arrival rate and thus no delay is caused. The situation becomes even worse in case of increasing maximum nodes speed. The links between intermediate links becomes volatile more often and the delay is caused even in case of relatively high data arrival rate. Again delay is not increased in case of OLSR and CLR since both protocols use proactive approach and does not require sending the data in order to detect the broken route.

V. Conclusions

This paper presents the performance analysis of TCP over Cluster-Label-based Routing Protocol for Mobile Ad Hoc Networks. The routing protocol is presentedon the top of a backbone created by Cluster-Label-based mechanism for Backbone. The routing discovery algorithm is performed reactively and route maintenance algorithmis performed proactively. The presented protocol is compared with existing protocols: OLSR, AODV and DSR which are standardized protocols under IETF MANET working group.The performance evaluations are presented by simulations and show that in terms of goodput the CLR performs better than compared protocols, especially in cases of high nodes mobility and large network size. Average delay imposed by discovery of broken routes is comparable with presented by proactive routing protocols such as OLSR.Overall the obtained results combined with previous studies on Cluster-Label-based mechanism for Backbone makes the CLR suitable protocol for MANETs in case of large networks with low to very high mobility of nodes.

참 고 문 헌

〔1〕 Internet Engineering Task Force, “Manet working group charter” http://www.ietf.org/html.charters/manet-charter.html.

〔2〕 T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, ”Optimized Link State Routing Protocol(OLSR)”, *IETF RFC 3626.*

〔3〕 C. E. Perkins, E. M. Belding-Royer, and S. R. Das, ”Ad-hoc On-Demand Distance Vector (AODV) Routing”, *IETF RFC 3561.*

〔4〕 D. B. Johnson, D. A. Maltz, and Y. Hu, ”The Dynamic Source Routing Protocol for Mobile Ad-hoc Networks (DSR)”, *IETF Internet Draft*, draft-ietfmanet-dsr-08.txt.

〔5〕 C. E. Perkins and P. Bhagwat, ”Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for Mobile Computers”, *SIGCOMM Symposium on Communications Architectures and Protocols.*

〔6〕 G. Holland and N. Vaidya, “Analysis of TCP Performance over Mobile Ad Hoc Networks”, *5th ACM/IEEE International Conference on Mobile Computing and Networking, ICMCN’99*

〔7〕 Zhenghua Fu, Xiaoqiao Meng, Songwu Lu, “How Bad TCP Can Perform In Mobile Ad Hoc Networks”, *7th International Symposium on Computer and Communications, ISCC’02.*

〔8〕 Ping Shung Ng and Soung Chang Liew, “Re-routing Instability in IEEE 802.11 Multi-hop Ad Hoc Networks”, *29th Annual IEEE International Conference on Local Computer Networks, LCN’04.*

〔9〕 Md. Abdullah-al-Mamun, M. Mahbubur Rahman and Hwee-Pink Tan, “Performance Evaluation of TCP Over Routing Protocols for Mobile Ad Hoc Networks”, *1st Chinacom*, October 2006.

〔10〕 Dongkyun Kim, Hanseok Bae, Jeomki Song, “Analysis of the Interaction between TCP Variants and Routing Protocols in MANETs”, *International Conference on Parallel Processing Workshops, ICPPW’05.*

〔11 A. Seddik-Ghaleb, Y. Ghamri-Doudane, S.-M. Senouci, “Effect of Ad Hoc Routing Protocols on TCP Performance within MANETs”, *International Workshop on Wireless Ad Hoc & Sensor Networks, IWWAN’06.*

〔12〕 K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. “A feedback-based scheme for improving TCP performance in ad hoc wireless networks.” *IEEE Personal communications*, 8 (1):34-39, Feb. 2001.

〔13〕 G. Holland and N.H. Vaidya, “Impact of Routing and Link Layers on TCP Performance in Mobile Ad Hoc Networks.", *in Proc. WCNC, New Orleans*, Sept. 1999.

〔14〕 Vitaly Li, Hong Seong Park, Hoon Oh, "Cluster label based Mechanism for Backbone on Mobile Ad Hoc Networks", *WWIC, 2006*

〔15〕 The network simulator – NS-2, *http://www.i-su.edu/nsnam/ns.*

〔16〕 C. Zhu, O. W. W. Yang, J. Aweya, M. Oullette, and D. Y. Montuno. "A comparison of active queue management algorithms using the OPNET Modeler.", *IEEE Communications Magazine*, 40(6): 158-167, 2002.

〔17〕 Ashwini K. Pandey and Hiroshi Fujinoki, "Study of MANET routing protocols by GloMoSim simulator", *Int. Journal of Network Management*, 15(6), pp.393-410, 2005.

〔18〕 Julian Hsu et al., "Performance of Mobile Ad Hoc Networking Routing Protocols in Realistic Scenarios.", *U.S. Army CECOM RDEC MILCOM '03.*

〔19〕 David Cavin and Yoav Sasson and André Schiper, "On the accuracy of MANET simulators", *Proceedings of the Workshop on Principles of Mobile Computing (POMC'02)*, pp.38-43, ACM, 2002.

Vitaly Li 정회원

2001년 8월 Khabarovsk State University of Technology (Russia), Applied Mathematics (학사)

2008년 8월 강원대학교 제어계측공학과 (박사)

<관심분야> WPAN, WLAN

하 재 열 (Jae Yeol Ha) 정회원

2003년 2월 서울대학교 전기공학부 (학사)

2008년 2월 서울대학교 전기컴퓨터공학부(박사)

2008년 3월~현재 (주)엠젠 선임연구원

<관심분야> WPAN, WLAN, Bio Mechatronics

오 훈 (Hoon Oh) 종신회원

1977년 성균관대학교 공학사

1992년 (미)텍사스A&M대학교 전산학(석사)

1995년 (미)텍사스A&M대학교 전산학(박사)

2000~2005 강원대 BK21 계약조교수

2005년~현재 울산대학교 교수

박 홍 성 (Hong Seong Park) 종신회원

1983년 서울대학교 제어계측공학과(학사)

1986년 서울대학교 제어계측공학과(석사)

1992년 서울대학교 제어계측공학과(박사)

1992년~현재 강원대학교 교수

<관심분야> Multimedia Communications, Wireless Networks, Bio Mechatronics