Performance of throughput-based Q-routing

Nattapong Kitsuwan\textsuperscript{a)}, Takumi Neda, and Eiji Oki
Department of Computer and Network Engineering,
The University of Electro-Communications, Tokyo, Japan
\texttt{a) kitsuwan@uec.ac.jp}

Abstract: This paper proposes a scheme to reduce an initial learning period of Q-routing. Q-routing is a routing scheme to guide a packet on the fastest route. A neighbor node with the lowest $Q$ value, which is determined by a remaining time to the destination, is selected to send the packet. Before the routing is stable, the node learns the $Q$ value at the initial period. The Q-routing performs well in only a high traffic network, compared to a shortest path routing scheme. The proposed scheme reduces the initial learning period of Q-routing by considering a throughput of the node.

Keywords: Q-routing, traffic load, routing scheme
Classification: Network

References

[1] C. J. C. H. Watkins, “Learning from delayed rewards,” Ph.D. Dissertation, King’s College, Cambridge, 1989.
[2] S. Kumar and R. Miikkulainen, “Dual reinforcement Q-routing: An on-line adaptive routing algorithm,” Proc. Artificial Neural Networks in Engineering, 1997.
[3] S. P. M. Choi and D. Y. Yeung, “Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control,” 1996.
[4] S. Khianjoom and W. Usaha, “Anycast Q-routing in wireless sensor networks for healthcare monitoring,” Proc. International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014. DOI:10.1109/ECTICon.2014.6839897

1 Introduction

The information is transmitted in a network as data packets. Each packet travels from a source to a destination via transit nodes in a network. A traveling path of the packet is determined by a routing scheme.

A shortest path routing scheme considers the minimum total link cost along the path between the source and the destination. Either a transmission time on a link, number of hops, or link capacity is considered as the link cost. One of the possible paths that has the lowest cost is selected for the transmission. Some link is popular to be served for several source and destination pairs since its link cost is low. As a result, the communication via that link will be congested. Moreover, the processor
of the nodes that connect to the link may be overloaded due to the processing of the packet headers. Packets have to be queued in the nodes for the processing. As a result, the packets may take time to reach the destination. In this case, it would be better if the packets are transmitted to the other appropriate path to avoid the congestion.

Q-routing is a routing scheme to select a neighbor node with the shortest time to deliver a packet to the destination [1, 2, 3]. It considers the minimum $Q$ value, which is obtained by time to send a packet to the destination. The $Q$ value at each node is updated, using the information from replied message of a neighbor node on the path, every time a packet is sent. Q-routing has been adopted in wireless sensor networks for healthcare monitoring to lower the communication overhead [4].

The performance of both shortest path routing and Q-routing schemes were compared in [1]. In a network with low traffic load, an average delivery time of the shortest path routing scheme is low, while the Q-routing performs as well as the shortest path routing scheme after an initial learning period. In a network with high traffic load, the shortest path routing scheme no longer endures the packet load due to the congestion of the popular link. The Q-routing then outperforms the shortest path routing scheme. The traffic load in the network fluctuates. For example, the traffic load in a network in a resident area is high in the morning and evening, and it is low during a business hour. One of the routing schemes cannot achieve a good performance in every traffic situation. Sending the packet through a path with low congestion can reduce the delivery time since a queue for packet processing in each node is short.

In this paper, a routing scheme to reduce the initial learning period using throughput history is proposed, called a throughput-based Q-routing (TQ) scheme. The proposed TQ scheme modifies the updating process for $Q$ value of the Q-routing scheme. In the updating process, the TQ scheme considers throughput of a local node, instead of considering the time spending in a queue at the local node and time to transmit the packet to a neighbor node. Simulation result shows that the initial learning period in the TQ scheme is reduced, compared to the Q-routing scheme.

2 Q-routing scheme

The Q-routing scheme determines a path based on a remaining time to reach the destination. At the initial stage, a node randomly selects a path for a packet. Once the packet is transmitted, the node learns a travelling time of the selected path. The node makes an adjustment to select a path who has the shortest travelling time. After the learning period, a path with the shortest time to the destination is obtained.

A packet is sent from a source node $s$ to a destination node $d$. $x$ is an intermediate node along the path between $s$ and $d$. Let $y$ be one of the neighbor nodes of $x$. Let $Q_x(d, y)$ be a delivery time to send a packet from node $x$ to node $d$ via node $y$, including queuing time at node $x$, $q_x$, and a transmission time between nodes $x$ and $y$, $s_{xy}$. After the packet arrives at node $y$, node $y$ replies a remaining time on the path, $t$, to node $x$, where
t = \min_{z \in \text{neighbors of } y} Q_y(d, z). \quad (1)

Once node \( x \) receives the response from node \( y \), \( Q_x(d, y) \) is updated as

\[
Q_x(d, y) = Q_x(d, y) + \eta(q_x + s_{xy} + t - Q_x(d, y)),
\]

where \( \eta \) is a learning rate.

Fig. 1 shows an example of the Q-routing scheme. Node \( x \) receives an updated information of \( Q \) value from its neighbor nodes \( y_1 \) and \( y_2 \). It is assumed that the current \( Q_x(d, y_1) = 2 \), \( Q_x(d, y_2) = 3 \), and \( \eta = 0.5 \). A queuing time at node \( x \) is assumed to be 5 ms. The transmission time between nodes \( x \) and each neighbor node is 1 ms, \( s_{xy_1} = s_{xy_2} = 1 \) ms. The updated \( Q_x(d, y_1) \) and \( Q_x(d, y_2) \) become 5 and 6, respectively. Since \( Q \) value of \( Q_x(d, y_1) \) is less than \( Q_x(d, y_2) \), the packet is sent from node \( x \) to node \( y_1 \).

When a packet travels via a node with low throughput, it takes time to get an update from its neighbor nodes. As a result, a time for the learning period is long.

3 Proposed TQ scheme

The TQ scheme enhances the Q-routing scheme by considering throughput, instead of remaining time to a destination. The value of \( Q \) is modified as,

\[
Q_x(d, y) = Q_x(d, y) + \eta(a + t - Q_x(d, y)),
\]

where \( a \) is the number of packets that were forwarded from the local node to neighbor nodes within a given period of time. There are two policies to define \( a \), non-indicated direction (ND) and indicated direction (ID).

3.1 Non-indicated direction policy

In the ND policy, a throughput of node \( x \) is defined by the number of forwarded packets considering the aggregated number of packets sent out from node \( x \) to all the neighbor nodes. Fig. 2(a) shows an example of \( Q \) value in the TQ scheme with ND policy. Node \( x \) receives an updated information of throughput from its neighbor nodes \( y_1 \) and \( y_2 \). The values of \( Q_x(d, y_1) \), \( Q_x(d, y_2) \), and \( \eta \) are the same as in Fig. 1.

A period of time to count the number of sent packets is set to five. This example shows the calculation of \( Q \) values from 17th to 21st units of time. Five packets are sent out from node \( x \) during the counting period. Updated \( Q_x(d, y_1) \) and \( Q_x(d, y_2) \) are

\[
Q_x(d, y_1) = 2 + 0.5(5 + 1 + 2 - 2) = 5
\]

\[
Q_x(d, y_2) = 3 + 0.5(5 + 1 + 3 - 3) = 6
\]
4.5 and 5.5 respectively. Since $Q_x(d, y_1) < Q_x(d, y_2)$, the packet is sent to node $y_1$.

### 3.2 Indicated direction policy

In the ID policy, a throughput of node $x$ is defined by the number of forwarded packets considering a direction of neighbor node. Fig. 2(b) shows an example of $Q$ value in the TQ scheme with ID policy. Node $x$ considers a throughput to each neighbor node. There are two and three packets sent out from node $x$ to nodes $y_1$ and $y_2$, respectively, during the counting period. Updated $Q_x(d, y_1)$ and $Q_x(d, y_2)$ are 3 and 4.5, respectively. Since $Q_x(d, y_1) < Q_x(d, y_2)$, a packet is sent to node $y_1$ at $t=22$.

### 4 Performance and evaluation

The performance of the proposed scheme is evaluated via simulation using a topology in Fig. 3(a). The pairs of source and destination are randomly generated. The TQ scheme is compared to the conventional Q-routing scheme. In the simulation, we define a step as a packet forwarding stage for both queuing and transmitting. We take ten steps of the throughput from the history. The average number of generated packets per step at each node is set to two. Percentage difference of each scheme/policies is defined by a percentage difference value, which is calculated by $|V_i - V_{i+1}|/((V_i + V_{i+1})/2) \times 100$, where $V_i$ is the average.
number of steps for delivery at $i$-th simulation step. The last simulation step that has the percentage difference value less than the given value is considered as a saturated point. In the evaluation, we set the percentage difference value to 2%.

Considering the number of simulation steps, Fig. 3(b) shows that the TQ scheme with both ND and ID policies have shorter simulation steps to achieve the saturated average number of steps for delivery than that of the Q-routing scheme. The Q-routing scheme takes 14,400 simulation steps. The TQ scheme with both ND and ID policies achieve the same result of 7,250 simulation steps, which is 50% reduction compared to the Q-routing scheme.

5 Conclusion

A scheme to reduce the initial learning period of the Q-routing scheme, called a throughput-based Q-routing (TQ) scheme, was proposed. The TQ scheme considers a throughput of neighbor nodes instead of remaining time to the destination. Packets are sent a neighbor node with low throughput to avoid congestion. Two policies, non-indicated direction (ND) and indicated direction (ID) policies, were introduced. The ND policy considers the aggregated number of packets that were sent out. The ID policy considers the number of packets that was sent to each neighbor node. Simulation result confirmed the reduction of the initial learning period of the proposed TQ scheme with both ND and ID policies.