Secure Route Discovery for QoS-Aware Routing in Ad Hoc Networks

Panagiotis Papadimitratos
Electrical and Computer Engineering
Virginia Tech
papadp@vt.edu

Zygmunt J. Haas
Electrical and Computer Engineering
Cornell University
haas@ece.cornell.edu

***Abstract*— We address the problem of securing the route discovery for Quality-of-Service (QoS)-aware routing in ad hoc networks. We provide a specification of secure route discovery for QoS-aware routing. We propose a reactive secure routing protocol, SRP-QoS, to defend against adversaries manipulating link and route metrics and, thus, prevent them from influencing the route selection. SRP-QoS ensures the accuracy of the discovered route(s) with respect to generalized link and route metrics. SRP-QoS is generally applicable, as it does not make restrictive assumptions on the network membership and trust, and it provides metrics for the constituent links of the discovered route(s), allowing the implementation of any route selection algorithm. As a result, SRP-QoS can enable QoS-aware routing in a wide range of ad hoc network instances.**

I. INTRODUCTION

The routing protocol discovers the network connectivity and provides communication paths to the network nodes. A route with few hops may be preferable than a longer one, but the hop count does not convey, in general, information on the 'quality' of the route. Attributes such as the route's reliability or resistance to failure, or the rate and the delay of data transmission achievable along the route, can be crucial in making a judicious route selection. For example, a set of reliable routes can alleviate frequent communication interruptions [1], or paths with sufficiently high bandwidth, low delay, or low delay variability can support resource-constrained applications [2]. *QoS-aware* routing, that is, routing of data according to the paths *Quality-of-Service* (*QoS*) *metrics*, can be highly beneficial in supporting a wide range of applications in ad hoc networks.

It is possible though that network nodes deviate from the protocol definition and exhibit malicious behavior. The challenge is to prevent such nodes, which we term *adversaries*, from misleading other nodes that a path is better than it actually is. If successful, an adversary can attract traffic and degrade or disable altogether the communication of other nodes. The recently proposed secure routing protocols, e.g., [3], [4], [5], provide only structural information, that is, route links or number of links (hops), on the network connectivity.

This is exactly the problem we address here. To discover accurate link and route metrics, we propose the *Secure Routing Protocol* for *QoS*-aware routing, *SRP-QoS*. We clarify that the role of *SRP-QoS* is not to secure the data transmission [6][1], or provide for route selection, metric estimation, and traffic handling [7], [8], [9]. Rather, *SRP-QoS* seeks to provide an accurate quantitative description of the discovered paths' attributes, and thus enable a variety of schemes that utilize such information to configure communication.

II. NETWORK AND ADVERSARY MODEL

A network *node* is a process with a unique identity *V*, a public/private key pair E_V, D_V, a module implementing the networking protocols, such as *SRP-QoS*, and a module providing communication across a wireless network interface, e.g., the widely adopted *IEEE 802.11* [10]. We are concerned with pair-wise communication across multiple wireless links between a *source S* and a *destination T*. We denote *S* and *T* as the *end nodes*, and the nodes that assist the *S, T* communication as *intermediate nodes.* We assume that each end-node knows the identity and the public key of its peer end-node, and that all nodes know the identities and the public keys of their neighbors. These, as well as the establishment of symmetric shared keys by the end nodes or two neighbors, can be achieved by protocols such as the *Neighbor Lookup Protocol* (*NLP*) or mechanisms that are part of the *Secure Routing Protocol* (*SRP*) [15], [11], [3].

We consider two models of *active* adversaries, *independent* adversaries and *arbitrary* adversaries [15]. *Independent adversaries* can modify, forge, or replay routing or data packets, but ignore received traffic that does not comply with the operation of the networking protocols, and thus do not generate any message due to the receipt of such traffic. Any message that does not follow the expected, protocol-specific format or fails one of the protocol checks is deemed as non-compliant.[2] If non-compliant traffic is attributed to misbehavior, independence implies that adversaries do not process and relay traffic that appears to originate from or have

[1] Adversaries can disrupt both phases of communication (route discovery, data transmission). In this work, we are concerned with attacks that target the route discovery.

[2] We emphasize that traffic is non-compliant if and only if the receiving node can detect that a message does not comply with the protocol; otherwise, messages that appear to be compliant, but actually are not, are processed as compliant.

been previously relayed by an adversary. In other words, independent adversaries do not attempt to assist other adversaries mounting an attack, either by ignoring the attack and further relaying traffic or by not responding to received non-compliant traffic.

This model of failures allows for a range of malicious behaviors and it is more general than *crash*, *omission* failures, and *timing* failures [12], [13]. Even though the malicious behavior of independent adversaries is constrained, the model does not prevent adversaries from simultaneously launching their attacks, which may have a compound effect. As it will become clear, the model of independent adversaries serves as a necessary condition to achieve stronger protocol properties than those achieved without the model's constraint on the adversarial behavior.

In general, adversarial nodes are allowed to deviate from the protocol execution in an arbitrary manner [14]. *Arbitrary adversaries* can be more sophisticated and powerful than independent adversaries, having, for example, knowledge of the identities of other adversaries in the network, devoting resources (e.g., route discovery) to establish direct and possibly private communication with other adversaries, and exchanging traffic and information about their local execution of the protocol.

III. Specification of Secure Route Discovery for QoS-Aware Routing

Let N be the set of network *nodes*, and E the set of unordered pairs of distinct nodes we denote as *links*. A *route* is a sequence of nodes $V_i \in N$, and links $e_{i,i+1}=(V_i, V_{i+1}) \in E$, for $0 \leq i \leq n-1$, i.e., *route*$=\{V_0, e_{0,1}, V_1, e_{1,2}, V_2, \ldots, V_{n-1}, e_{n-1,n}, V_n\}$. Referring to a route as a sequence of nodes V_i implies that for any two consecutive nodes of the route $(V_i, V_{i+1}) \in E$. We call a route with $V_0 \equiv S$ and $V_n \equiv T$ an (S,T)*-route*. Let $f : E \rightarrow M \subseteq \Re$ be a function that assigns *labels*, that is, real values $f(e_{i,i+1})=m_{i,i+1} \in M$, to route edges $e_{i,i+1}$. Each label $m_{i,i+1}$, which we denote as a *link metric*, provides a quantitative description of the $e_{i,i+1}$ attribute(s).

The routing protocol input is a pair of nodes, S and T, and the output is an (S,T)-route; we term this as *basic* routing protocol. What we are after here is an *augmented* routing protocol, with S and T as input, and output an (S,T)-route and a sequence of link metrics, with one metric for each (S,T)-route link. Then, the attributes of the entire route can be 'summarized' by the aggregate value of the link metrics. The aggregate value is calculated by a function $g : M \rightarrow \Re$ we denote as the *route metric* $g(m_{0,1}, \ldots, m_{n-1,n})$. The form of g is dependent on the protocol, and we consider four different forms in Sec. IV. Moreover, we define $l_{i,i+1}$ to be the *actual* metric value for link $e_{i,i+1}$, and the aggregate $g(l_{0,1}, \ldots, l_{n-1,n})$ of the actual link metrics as the actual route metric.

Let t_1 and $t_2 > t_1$ be two points in time that define a time interval (t_1, t_2), with time t_2 the instance at which the routing protocol discovers a route. We are interested in routing protocols which ensure three properties for the discovered route(s): *loop-freedom*, *freshness*, and *accuracy*. An (S,T)-route is loop-free if it has no repetitions of nodes, and it is fresh with respect to the (t_1, t_2) interval if each of the route's constituent links is *up* at some point in time during the interval (t_1, t_2). We do not discuss further loop-freedom and freshness [15], as they are not specific only to the augmented or *QoS*-aware route discovery.

Accuracy**:** an (S,T)-route is accurate with respect to a route metric g and a constant $\Delta_{good} > 0$ if $| g(m_{0,1}, \ldots, m_{n-1,n}) - g(l_{0,1}, \ldots, l_{n-1,n}) | < \Delta_{good}$.

Accuracy provides the assurance that the quantitative description of a route reflects its actual attributes. It is necessary to prevent adversaries from manipulating the metric values, e.g., contributing arbitrary metric values and from altering metrics provided by other nodes along the route, and thus misleading end-nodes into believing that a discovered route is better than it actually is. Route accuracy requires that the route metric calculated by the protocol is within Δ_{good} from the actual value. Δ_{good} is a constant such that, despite malicious or benign faults that lead to inaccurate metric values, the route metric is still 'reasonably' close to the actual value and meaningful for the protocol. The definition allows for some protocol- and metric-specific $\Delta_{good} > 0$, because, even in a benign network, impairments can affect measurements and calculations for the metric values.

IV. SRP-QoS Operation

Metrics are maintained for each link (V_i, V_{i+1}), with all nodes using the same algorithm to calculate or estimate $m_{i,i+1}$. To distinguish the values calculated by each node incident on (V_i, V_{i+1}), we denote the metric calculated by V_i as $m_{i,i+1}^{i}$ and the metric calculated by V_{i+1} as $m_{i,i+1}^{i+1}$. We require that $m_{i,i+1}^{i} = m_{i,i+1}^{i+1}$ or $\left| m_{i,i+1}^{i} - m_{i,i+1}^{i+1} \right| < \varepsilon$ for some $\varepsilon > 0$. If the metric in use is some fixed, 'administrative' cost agreed upon between the two neighbors, then $m_{i,i+1}^{i}$ must be equal to $m_{i,i+1}^{i+1}$. Otherwise, a protocol-selectable and metric-specific threshold ε determines the maximum allowable discrepancy between $m_{i,i+1}^{i}$ and $m_{i,i+1}^{i+1}$. Despite the assumed symmetry of the link, ε allows for some inaccuracy due to network impairments that may affect measurements necessary for the metric calculation. Metrics such as the willingness of the node to relay data, or its remaining battery power, can be determined only independently at each node and do not fit in the above definition.

The *source node* (S) determines the metric of interest in outgoing request packets, if more than one metrics are supported, and initiates a route discovery for a *destination node* (T). S transmits a route request packet (*RREQ*) that comprises S, T, a query identifier Q, an authenticator $A=f_K(S, T, Q)$ calculated as a function of the *RREQ* fields and a key K,[3] and two empty lists, *NodeList* and *MetricList*.

[3] The function f and the key depend on the cryptographic primitives. Similarly to *SRP*, K can be a symmetric key shared by S and T.

Each intermediate node V_i invokes the *PreviouslySeen(RREQ)* routine[4] to specify if *RREQ* should be processed. If so, V_i extracts the last entry of the *NodeList* and verifies it is the address of its *precursor* V_{i-1}. It checks the *NodeList* for duplicate entries and if the number of *MetricList* entries is equal to the number of the *NodeList* entries. If any check fails, *RREQ* is discarded. Otherwise, V_i appends its own address and $m^{i}_{i-1,i}$. For each *RREQ* it relays, V_i initializes a *ForwardList*, and adds to the *ForwardList* each neighbor V_{i+1} it overhears relaying *RREQ* with *NodeList*={*NodeList*, V_{i+1}} and *MetricList*={*MetricList*, $m^{i+1}_{i,i+1}$}; *S* updates its own *ForwardList* similarly.

T performs the same checks as intermediate nodes, and, additionally, calculates $f_K(S, T, Q)$. If this matches *A*, *T* returns a *route reply* (*RREP*), comprising *S*, *T*, *Q*, a *Route* list of the V_1, V_2, …, V_{n-1} nodes accumulated in the *RREQ NodeList* in *reverse order*, the corresponding *MetricList* entries with $m^{T}_{n-1,T}$ appended, and an authenticator $A' = f_K(S, T, Q, Route, MetricList)$.

V_i verifies that its *successor* V_{i+1} is indeed the node that now forwards the *RREP*, and that $V_{i+1} \in ForwardList$ unless the successor is *T*. If so, V_i checks whether $\left| m^{T}_{i,T} - m^{i}_{i,T} \right| < \varepsilon$. In general, V_i checks if $m_{S,i} = m'_{S,i}$, where $m'_{S,i}$ is the aggregate value calculated from the link metric values reported in the *RREP* for links (V_k, V_{k+1}), $k<i$. If any of these checks fails, *RREP* is discarded. Otherwise *RREP* is relayed to the next node along the *Route* list. Once *RREP* reaches the source, *S* calculates and compares $f_K(S, T, Q, Route, MetricList)$ to A' and extracts *Route* and *MetricList* entries.

V. SRP-QoS Correctness

We consider three forms of the aggregate function *g* to calculate the route metric and show how route accuracy is achieved. If $g\left(m^{1}_{0,1},\ldots,m^{k}_{k-1,k}\right) = \sum_{i=0}^{k-1} m^{i+1}_{i,i+1}$, we denote the function *g* as g_{add} and the constant Δ_{good} as Δ^{add}_{good}, if $g\left(m^{1}_{0,1},\ldots,m^{k}_{k-1,k}\right) = \max_{0 \le i \le k-1}\left\{ m^{i+1}_{i,i+1} \right\}$ the function is denoted as $g_{\max}$ and the constant as $\Delta^{\max}_{good}$, and if $g\left(m^{1}_{0,1},\ldots,m^{k}_{k-1,k}\right) = \min_{0 \le i \le k-1}\left\{ m^{i+1}_{i,i+1} \right\}$ as $g_{\min}$ and $\Delta^{\min}_{good}$. For $m^{i+1}_{i,i+1} > 0$, $g\left(m^{1}_{0,1},\ldots,m^{k}_{k-1,k}\right) = \prod_{i=0}^{k-1} m^{i+1}_{i,i+1}$ can be written as $g_{add}\left(\overline{m}^{1}_{0,1},\ldots,\overline{m}^{k}_{k-1,k}\right)$, where $\overline{m}^{i+1}_{i,i+1} = \log(m^{i+1}_{i,i+1})$, for $0 \le i \le k-1$.

Lemma: *Routes discovered by SRP-QoS in the presence of independent adversaries are accurate, with respect to (i)* g_{add} *and* $\Delta^{add}_{good} = \varepsilon k^2 + k\delta^*$, *(ii)* $g_{\max}$ *and* $\Delta^{\max}_{good} = k\varepsilon + \delta^*$, *and (iii)* $g_{\min}$ *and* $\Delta^{\min}_{good} = k\varepsilon + \delta^*$, *with k the number of route links, ε>0 the maximum allowable difference between* $m^{i}_{i,i+1}$ and $m^{i+1}_{i,i+1}$, *and* $\delta^* > 0$ *the maximum error for a metric calculation by a correct node.*

Proof: *See Appendix.*

The lemma shows that the accuracy of a discovered route depends on the worst-case 'error' introduced by adversaries, and $\delta^* > 0$ due to the inaccuracy of the estimation of a link metric. The latter factor, captured by the second term of Δ_{good} in the three cases considered above, is not related to security but only the operation of the network. In fact, such inaccuracies cannot be eliminated in general. The former factor, captured by the first term of Δ_{good}, depends on the protocol parameter *ε* and the length of the discovered route *k*. The longer the route, the more adversaries may be along the route, and thus the higher the discrepancy between the actual and the discovered route metric due to the compound effect of the adversaries' misbehavior.

VI. Discussion

The assumption of independent adversaries is a necessary condition to achieve accuracy. Without it, *SRP-QoS* cannot ensure route accuracy, and, in fact, if at least two arbitrary adversaries M_1, M_2 are part of a discovered route, then at least one link of the route may have never been up during the (t_1, t_2) interval, where t_1 is the point in time S transmitted *RREQ* and t_2 the time of the route discovery. This is possible if M_1 and M_2 'tunnel' *RREQ*, *RREP* packets to each other, so that an (M_2, M_1) link is reported in the (S,T)-route, even though M_1 and M_2 are not neighbors. If M_1, M_2, …, M_k, $k \geq 2$, arbitrary adversaries form a path, it is possible that any M_i, for $i \neq 1,k$, can modify *RREQ*, e.g., adding or removing links and providing arbitrary metric values, while all such M_i will later relay the *RREP*. It suffices that any of the $M_2,\ldots, M_{k-1}$, (in the general case that $k>2$) does not perform the checks required by the protocol and simply relays the protocol packets. If M_i were independent, M_3 for example would have ignored any non-compliant *RREQ* it receives from M_2, and similarly for the tunneling attack, M_2 (M_1) would ignore traffic received from M_1 (M_2) as non compliant. As a result, there may be no actual link metric for a (M_i, M_{i+1}) link, and adversaries can provide any arbitrary value for the metrics of such links.

Finally, a note on related work: [16] proposes a symmetric-key mechanism for the discovery of an end-to-end resource metric, and [17] proposes the collection of link weights reflecting prior node (mis)behavior and validation of digital signatures at each intermediate node. In comparison, *SRP-QoS* is more general and more efficient, as it provides both general link and route metrics and relies on symmetric-key cryptographic primitives. Moreover, a generalized treatment of

[4] The *PreviouslySeen*( ) routine can be implemented in different ways, trading off robustness for lower routing overhead, ranging from relaying a single copy of each query identified by the *S*, *T*, *Q* triplet, up to relaying Q_{RED} distinct query packets, or even a constant number of *RREQ* copies received per neighbor. Note that the implementation can differ at intermediate nodes and the destination.

different metric types, and formal reasoning on the correctness of *QoS*-aware routing for *MANET*s was not presented before.

References

[1] P. Papadimitratos, Z.J. Haas, and E.G. Sirer, "Path Set Selection in Mobile Ad Hoc Networks," in proceedings of the *Third ACM Symposium on Mobile Ad Hoc Networking & Computing* (*MobiHoc 2002*), Lausanne, Switzerland, Jun. 2002

[2] E. Crawly, R. Nair, B. Rajagopalan, and H. Sandik, "A Framework for QoS-based Routing in the Internet," *IETF RFC 2386*, Aug. 1998

[3] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in proceedings of the *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference* (*CNDS 2002*), San Antonio, TX, Jan. 27-31, 2002

[4] Y-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in proceedings of the *8th ACM International Conference on Mobile Computing and Networking*, Sept. 2002

[5] M. G. Zapata and N. Asokan, "Securing Ad hoc Routing Protocols," in proceedings of the *ACM WiSe 2002*, Atlanta GA, Sept. 2002

[6] P. Papadimitratos and Z.J. Haas, "Secure Message Transmission in Mobile Ad Hoc Networks," in proceedings of the *ACM WiSe 2003*, San Diego CA, Sept. 2003

[7] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for Multi-Hop Wireless Routing," in proceedings of the *9th MobiCom*, San Diego, CA, Sept. 2003.

[8] G-S. Ahn, A. T. Campbell, A. Veres, and L. Sun, "Supporting Service Differentiation for Real-Time and Best Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN)," *IEEE Transactions on Mobile Computing*, Sept. 2002

[9] S-B. Lee, G-S. Ahn, X. Zhang and A. T. Campbell, "INSIGNIA," *Internet Draft*, draft-ietf-manet-insignia-00.txt, MANET WG, Nov. 1999

[10] IEEE Std. 802.11, "Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications," 1999

[11] P. Papadimitratos and Z.J. Haas, "Secure Link State Routing for Mobile Ad Hoc Networks," in proceedings of the *IEEE CS Workshop on Security and Assurance in Ad hoc Networks*, in conjunction with the *2003 International Symposium on Applications and the Internet*, Orlando, FL, Jan. 2003

[12] V. Hadzilacos and J. Y. Halpern, "Message-optimal protocols for Byzantine agreement," *Mathematical Systems Theory*, 26, pp. 41-102, 1993

[13] F. Cristian, H. Aghili, R. Strong, and D. Dolev, "Atomic broadcast: From simple message diffusion to Byzantine agreement," in proceedings of the *Fifteenth International Symposium on Fault-Tolerant Computing*, p. 200-206, June 1985. Also, IBM Research Laboratory Technical Report RJ5244, Apr. 1989

[14] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM*, 27(2):228-234, Apr. 1980

[15] P. Papadimitratos, "Secure and Fault-tolerant Communication in Mobile Ad Hoc Networks," PhD Dissertation, Cornell University, 2004

[16] Y-C. Hu and D. Johnson, "Securing QoS Route Discovery in On-Demand Routing for Ad Hoc Networks," in proceedings of *SASN' 04*, Washington, DC, Oct. 2004

[17] B. Awerbuch, D. Holmer, C. Nita-Rotaru and H. Rubens, "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures," in proceedings of the *ACM WiSe 2002*, Atlanta GA, Sept. 2002

Appendix

Proof of Lemma: Let an adversary $V_i \in (S,T)$-route modify one or more of the *MetricList* entries, relaying a *RREP* with a tampered $MetricList'$. S will discard such a *RREP*, because f_K(*S*, *T*, *Q*, *Route*, $MetricList'$) $\neq A'$. Next, consider an adversary V_i that tampers with one of the $m_{j,j+1}^{j+1}$ values in the *MetricList*, for $j<i-1$, and relays a *RREQ* with the tampered metric list. The *RREQ* appears as protocol-compliant to all nodes that further relay it, as well as *T* that generates a *RREP*. When *RREP* arrives back at V_j, $m_{S,i} - m'_{S,i} \neq 0$, and V_j rejects *RREP* as non-compliant. Moreover, let V_i tamper with one of the $m_{j,j+1}^{j+1}$ in the *MetricList*, for $j \geq i$, and relay *RREQ* with a tampered metric list. Then, V_{i+1} will reject *RREQ* as non-compliant, because all $m_{j,j+1}^{j+1}$ for $j \geq i$ must be void, as they correspond to links not yet discovered. If V_i appended one or more additional entries to *NodeList*, *RREQ* would be discarded by all neighbors of V_i that $receive_L$(*RREQ*), because the last node in *NodeList* would not be the neighbor relaying *RREQ*.

We denote by $\delta_i > 0$ the error of the metric calculation with respect to the actual link metric, so that $m_{i-1,i}^{i} = l_{i-1,i} \pm \delta_i$ and $m_{i,i+1}^{i} = l_{i,i+1} \pm \delta_i$. From the protocol definition, V_i does not ignore a *RREQ* as non-compliant only if V_{i+1} appends $m_{i,i+1}^{i+1}$ such that $\left| m_{i,i+1}^{i} - m_{i,i+1}^{i+1} \right| < \varepsilon$. For the discovery of an (*S*,*T*)-route with *k* nodes, the above inequality must be true for all $0 \leq i \leq k-1$. We consider the worst case, with *S* and *T* correct, i.e., $\delta_0 < \delta^*$ and $\delta_k < \delta^*$, and all intermediate nodes adversaries and arbitrary δ_i. Then, $\left| m_{0,1}^{0} - m_{0,1}^{1} \right| = \left| l_{0,1}^{0} \pm \delta_0 - (l_{0,1}^{1} \pm \delta_1) \right| < \varepsilon \Rightarrow \delta_1 < \varepsilon + \delta^*$ for the first hop; then, for the 2nd hop, $\left| m_{1,2}^{1} - m_{1,2}^{2} \right| = \left| l_{1,2}^{1} \pm \delta_1 - (l_{1,2}^{2} \pm \delta_2) \right| = \left| \delta_2 - \delta_1 \right| < \varepsilon \Rightarrow \delta_2 < \varepsilon + \delta_1 < 2\varepsilon + \delta^*$, and, in general, $\delta_i < i\varepsilon + \delta^*$. Also, $\left| m_{k-1,k}^{k-1} - m_{k-1,k}^{k} \right| < \varepsilon \Rightarrow \delta_{k-1} < \varepsilon + \delta^*$ and $\delta_i < (k-i)\varepsilon + \delta^*$. Thus, $\delta_i < \min\left\{ i\varepsilon + \delta^*, (k-i)\varepsilon + \delta^* \right\}$, and since δ^* does not depend on *k*, *i*, and ε, $\delta_i < \min_{1 \leq i \leq k-1} \left\{ i\varepsilon, (k-i)\varepsilon \right\} + \delta^*$, $1 \leq i \leq k-1$.

First, if the route metric is calculated by g_{add}, $g_{add}\left(m_{0,1}, \ldots, m_{k-1,k}\right) = g(l_{0,1}, \ldots, l_{k-1,k}) \pm \sum_{i=1}^{k-1} \delta_i \pm \delta^*$. The sum of δ_i is bounded since each of its terms is bounded:

$$\sum_{i=1}^{k-1} \delta_i < \sum_{i=1}^{k-1} \left(\min\left\{ i\varepsilon, (k-i)\varepsilon \right\} + \delta^* \right) = \begin{cases} \frac{\varepsilon}{4}\left(k^2 - 5\right) + (k-1)\delta^*, & k \text{ odd} \\ \varepsilon\left(\frac{k^2}{4} - 1\right) + (k-1)\delta^*, & k \text{ even} \end{cases}.$$

Then, we select $\Delta_{good}^{add} = \varepsilon k^2 + k\delta^*$. Second, if the route metric is g_{max}, $g_{\max}\left(m_{0,1}, \ldots, m_{k-1,k}\right) = \begin{cases} \max\limits_{0 \leq i \leq k-1} \left\{ l_{i,i+1} \right\} + \max\limits_{1 \leq i \leq k} \left\{ \delta_i \right\} \\ \max\limits_{0 \leq i \leq k-1} \left\{ l_{i,i+1} \right\} - \min\limits_{1 \leq i \leq k} \left\{ \delta_i \right\} \end{cases}$ and we select $\Delta_{good}^{\max} = k\varepsilon + \delta^*$. Finally, if the route metric is g_{min}, $g_{\min}\left(m_{0,1}, \ldots, m_{k-1,k}\right) = \begin{cases} \min\limits_{0 \leq i \leq k-1} \left\{ l_{i,i+1} \right\} - \max\limits_{1 \leq i \leq k} \left\{ \delta_i \right\} \\ \min\limits_{0 \leq i \leq k-1} \left\{ l_{i,i+1} \right\} + \min\limits_{1 \leq i \leq k} \left\{ \delta_i \right\} \end{cases}$, and $\Delta_{good}^{\min} = k\varepsilon + \delta^*$. This completes the proof of the lemma.