Minimizing the Impact of Stale Link State Information on QoS Routing
Gang Cheng and Nirwan Ansari

Abstract—In this paper, we show that routing without considering the staleness of link state information introduced by update policies may generate significant percentage of false routing. Hence, we introduce and investigate the issue of minimizing the impact of stale link state information on the performance of QoS routing without stochastic link state knowledge. Under the assumption that trigger-based link state policies are adopted for updating link state information, we theoretically decouple the problem of finding the most probable feasible path (without link state stochastic knowledge) to the problems of finding the Multiple Additively Constrained Path (MACP) and finding the Least Cost Multiple Additively Constrained Path (LCMACP), respectively, and propose a framework for minimizing the impact of stale link state information on the performance of QoS routing. We show by theoretical analysis and extensive simulations that our proposed framework is effective in minimizing the undesirable effect of the staleness of link state information.

Index Terms—Link state update, QoS routing, NP-complete.

I. INTRODUCTION

State distribution and routing strategy are the two issues related to QoS routing [1]. State distribution addresses the issue of exchanging the state information throughout the network. Routing strategy is used to find a feasible path that meets the QoS requirements. Many proposed Quality-of-Service (QoS) routing solutions assume that accurate link state information is available to each node. However, this is practically impossible in real networks. To overcome this problem, many trigger-based link state update policies (threshold, equal class, and exponential class based update policies) [2] have been proposed. Given a predefined threshold value ($\tau$), an update is triggered in the threshold based update policy if ($b_c - b_0$) / $b_0 > \tau$, where $b_0$ is the last advertised value of the available bandwidth, and $b_c$ is the current available bandwidth. However, in equal class and exponential class based update policies, the bandwidth is partitioned into classes and an update is triggered whenever the available bandwidth crosses a class boundary. The only difference between them is that the bandwidth is partitioned into classes of equal size ((0, B), (B, 2B),...), in equal class based update policy, while it is partitioned into unequal classes, whose sizes ((0, B), (B, (f + 1)B), ((f + 1)B, (f^2 + f + 1)B),...) grow geometrically by a factor of $f$, in the exponential class based update policy, where B is a predefined constant. In [3], extensive simulations were made to uncover the effects of the stale link state information and random fluctuations in the traffic load on the routing and setup overheads. Instead of using the link capacities or instantaneous available bandwidth values, Li et al. [4] used a stochastic metric, Available Bandwidth Index (ABI), and extended BGP to perform the bandwidth advertising.

Multi-Constrained Path selection is an NP-complete problem and has been intensively investigated in the last decade. For the case that only inaccurate link state information is available to nodes, approximate solutions [6] have been proposed for the Most Probable Bandwidth Delay Constrained Path (MP-BDCP) selection problem by decomposing it into two sub-problems: the Most Probable Delay Constrained Path (MP-DCP) and the Most Probable Bandwidth Constrained Path (MP-BCP). In [7], a heuristic algorithm was proposed based on a linear cost function for two additive constraints; this is an MCP (Multiple Constrained Path Selection) problem with two additive constraints. A binary search strategy for finding the appropriate value of $\beta$ in the linear cost function $w_1(p) + \beta w_2(p)$ or $\beta w_1(p) + w_2(p)$, where $w_i(p)$ ($i = 1, 2$) are the two respective weights of the path $p$, was proposed, and a hierarchical Dijkstra algorithm was introduced to find the path. It was shown that the worst-case complexity of the algorithm is $O(\log \beta_{\max} (m + n \log n))$, where $\beta_{\max}$, $n$, and $m$ are the upper bound of the parameter $\beta$, and the number of links and nodes, respectively. Many researchers have posed the QoS routing problem as the $k$-shortest path problem. An algorithm, called TARCRA [8], was proposed for MCP by using a non-linear cost function and a $k$-shortest path algorithm. The computational complexity of TARCRA is $O(kn \log (kn) + k^3 m M)$, where $k$ is the number of shortest paths and $M$ is the number of constraints.

Designing a QoS routing algorithm based on a specific update policy has been rarely considered. In this paper, we show that general QoS routing algorithms without considering the staleness of the link state information may introduce unignorable percentage of false routing. Hence, we introduce and investigate the problem of minimizing the impact of stale link state information. To this end, we propose a framework, with which the probability of false routing is minimized.

II. PROPOSED FRAMEWORK

We first lay out the assumptions made in this paper. Practically, protocol overhead will be intolerably high if link state is updated whenever a minor change occurs. Hence, for the sake of reducing the protocol overhead, the staleness of link state information is inevitably introduced. In this paper, we focus on
minimizing the impact of the staleness of link state information (introduced by the link state update policy) on QoS routing. Hence, we ignore the effect of the dissemination delay on the link state update and simply assume that the dissemination delay is zero, i.e., all nodes can instantly receive the link state information upon an update. In this section, for simplicity, we only consider the class based link state update policies. We will show that our proposed framework can be easily extended to cases where other trigger-based update policies are adopted. We will not focus only on a single QoS metric (e.g., bandwidth) as in [2] because the framework proposed in this paper is applicable to the case of multiple constraints. Assume that $c^1_j(t), c^2_j(t), \ldots, c^M_j(t)$ are $M$ QoS metrics associated with link $j$ at moment $t$, and we partition the $i$th metric of link $j$ into $k_i^j$ classes $((0, B^i_{t,j}), (B^i_{t,j}, B^{i+1}_{t,j}), \ldots, (B^{k_i^j-1}_{t,j}, B^{k_i^j}_{t,j}))$. Many studies have been done to characterize the Internet traffic, revealing interesting facts such as Long-Range Dependence or multi-fractal behaviors [9]-[11]. Therefore, we cannot assume that the link state is memoryless. Hence, each QoS metric of a link $(c^1_j(t), c^2_j(t), \ldots, c^M_j(t))$ is viewed as a random process with memory. Without loss of generality, we assume that for any $i$ and $j$, $k_i^j = k_i$ and $B^{i+1}_{t,j} = B^i_{t,j}$, i.e., $\forall i \in \{1, 2, \ldots, M\}$, the $i$th link metrics of all links are partitioned into the same set of classes, where $k_i$ represents the number of classes for the $i$th metric and $B^i_{t,j}$ the lower bound of the $(j+1)$th class of the $i$th metric. As mentioned above, an update is triggered only when the available bandwidth crosses a class boundary. Denote $C^j_{t,u}(t), C^j_{t,v}(t), \ldots, C^j_{t,M}(t)$ as the latest updated link state of link $j$ from the point of the view of node $u$ at moment $t$, i.e., from the perspective of node $u$, the $M$ QoS metrics of link $j$ at moment $t$ are $C^j_{t,u}(t), C^j_{t,v}(t), \ldots, C^j_{t,M}(t)$. Since we ignore the staleness of link state information introduced by the dissemination delay, for any two nodes $u$ and $v$,

$$C^j_{t,u}(t) = C^j_{t,v}(t) = C^j_{t}(t). \tag{1}$$

Moreover, $\exists \alpha^j_i(t)$ ($0 \leq \alpha^j_i(t) \leq k_i$) such that

$$B^{\alpha^j_i(t)-1}_{t,i} \leq C^j_{t,i}(t) \leq B^{\alpha^j_i(t)}_{t,i}. \tag{2}$$

Since we assume that the partitions (classes) for the metrics of all links are the same, as long as $C^j_{t,i}(t)$ is available to a node, $B^{\alpha^j_i(t)-1}_{t,i}$ and $B^{\alpha^j_i(t)}_{t,i}$ are also known. In fact, if the assumption does not hold for some networks, we can encode $B^{\alpha^j_i(t)-1}_{t,i}$ and $B^{\alpha^j_i(t)}_{t,i}$ in the update packets by which link state information is disseminated. At any given moment, when a connection request arrives at a node, we assume that the node tries to compute a path meeting the QoS requirement of the connection according to its available link state information (source routing). If, from the perspective of the node, there is enough network resource (bandwidth) to accommodate this connection, it starts a setup process for the connection. Otherwise, it rejects the connection request immediately. Ideally, the connection is accepted if there is actually enough network resource, and rejected otherwise. However, due to inaccurate link state information ($C^j_{t,i}(t) \neq C^j_{t}(t)$) and the adopted routing algorithm, the following two discrepancies may arise:

- False positive: There is actually not enough network resource to accommodate a connection, but is indicated otherwise by its link state information. Since a setup process will be initialized by the node, network resource is wasted.
- False negative: A connection can actually be accepted by the network, but is rejected by the node because of inaccurate link state information or failure of the adopted routing algorithm in finding a feasible path.

Collectively, both failures are referred to as false routing in this paper. Designing a link state update policy [12]-[13] is beyond the scope of this paper. Our objective here is to find a solution that minimizes false routing. Our problem is thus formulated as follows:

**Definition 1:** Given a network $G(N, E)$, where $N$ is the set of nodes, and $E$ the set of links. Assume $C^1_j, C^2_j, \ldots, C^M_j$ are the real $M$ QoS metrics associated with link $j$, and $C^1_j, C^2_j, \ldots, C^M_j$ the latest updates available to the nodes, where $j = 1, 2, \ldots, m$, and $m$ is the number of links. Find a (real) feasible path $p$ from node $s$ to a destination subject to $\forall i \in \{1, 2, \ldots, M\}$, $\sum_{j \in p} c^i_j \leq \rho_i$ if $c^i_j$ is an additive metric, or $\min_{j \in p} c^i_j > \rho_i$ if $c^i_j$ is a concave metric, where $\rho_1, \rho_2, \ldots, \rho_M$ are the given QoS constraints.

In general, QoS constraints can be classified into three categories: concave, additive, and multiplicative. Since multiplicative constraints can be converted into additive constraints by using the logarithm function, we only consider concave and additive constraints in this paper.

Ideally, we hope that the routing algorithms can achieve 100% success ratio in finding a real feasible path without false negatives and positives. It is, however, practically impossible owing to the stale link state information. On the other hand, we can reduce the probability of false routing by taking the advantage of the relationship between the current and updated link state. For instance, given a network as shown in Fig. 1, in which we denote $B^{\alpha^i-1}_{t,i}, C^j_{t}$, and $B^{\alpha^i}$ of link $j$ in the form of $(B^{\alpha^i-1}, C^j_{t}, B^{\alpha^i})$ (since there is only one constraint, subscript $i$ is omitted). Assume the metric is concave and a connection request from node $a$ to node $c$ with constraint $\rho = 0.45$ arrives. Both paths $a \rightarrow b \rightarrow c$ and $a \rightarrow d \rightarrow c$ seem to be feasible according to the updated link state information because $\min_{p \in \{a \rightarrow b \rightarrow c, a \rightarrow d \rightarrow c\}} \rho_i > \rho$. However, since the lower bounds of current classes of links $e(a, d)$ and $e(d, c)$ are 0.4 and 0.3, respectively, it is possible
that \( \min_{j \in \{a, b, c\}} c^j < \rho \), i.e., the path \( a \rightarrow d \rightarrow c \) can be an infeasible path. Meanwhile, the current link states of links \( e(a, b) \) and \( e(b, c) \) are both \((0.5, 0.6, 0.7, 0.8)\), implying that

\[
\min_{j \in \{a, b, c\}} c^j > \min_{j \in \{a, b, c\}} B^{a\rightarrow l} = 0.5 > \rho.
\]

(3)

Hence, we can guarantee that the path \( a \rightarrow b \rightarrow c \) is a real feasible path. Denote \( \alpha_i \) as the integer such that

\[
B_i^{a\rightarrow l} \leq C_i^j \leq B_i^{a\rightarrow l+1} \Rightarrow B_i^{a\rightarrow l} \leq C_i^j \leq B_i^{a\rightarrow l+1}.
\]

(4)

We call tightening the \( i \)-th metric of link \( j \) as setting the \( i \)-th metric of link \( j \) as \( B_i^{a\rightarrow l+1} \) if the \( i \)-th metric is concave, or as \( B_i^{a\rightarrow l} \) if the \( i \)-th metric is additive.

**Theorem 1:** Assume \( G(N, E, i) \) is the network constructed from \( G(N, E) \) by tightening the \( i \)-th metric of all links. If \( p \) is a feasible path of \( G(N, E, i) \), \( p \) must also be a feasible path of \( G(N, E) \).

**Proof:** Consider the case that the \( i \)-th metric is concave. By Eq. 4,

\[
\min_{j \in p} B_i^{a\rightarrow l-1} \leq \alpha_i \Rightarrow \min_{j \in p} c_i^j \geq \rho_i.
\]

(5)

Therefore, \( p \) satisfies the \( i \)-th constraint when it is concave. Consider the other case that the \( i \)-th metric is additive. By Eq. 4,

\[
\sum_{j \in p} B_i^{a\rightarrow l} \leq \alpha_i \Rightarrow \sum_{j \in p} c_i^j \leq \rho_i
\]

(6)

i.e., the sum of the \( i \)-th metric of \( p \) is less than the constraint. Therefore, \( p \) is a feasible path of \( G(N, E) \).

Denote \( G(N, E) \) as the network constructed from \( G(N, E) \) by tightening all metrics of links. We can eliminate false positives by the above theorem because the feasible path computed in \( G(N, E) \) must also be a feasible path of \( G(N, E) \), i.e., without knowing the exact link state information, false positives may be avoided by Theorem 1. However, a feasible path of \( G(N, E) \) may not be a feasible path of \( G(N, E) \). It is still possible that we fail to find a feasible path in \( G(N, E) \) although a real one exists in \( G(N, E) \). \( \blacksquare \)

**Definition 2:** For a concave constraint \( \rho_i \), \( i = 1, 2, ..., M \), link \( j \) is referred to as an ensured feasible link if \( B_i^{a\rightarrow l-1} \geq \rho_i \); an unsure feasible link if \( C_i^j \geq \rho_i \) and \( B_i^{a\rightarrow l-1} = \rho_i \); an unsure infeasible link if \( C_i^j < \rho_i \) but \( B_i^{a\rightarrow l} > \rho_i \); and an ensured infeasible link if \( B_i^{a\rightarrow l} < \rho_i \).

If no feasible path can be found in \( G(N, E) \), we hope to find a path with the largest probability to be a real feasible path in \( G(N, E) \). In this paper, we only assume that a node knows the updated link state information, i.e., no stochastic link state knowledge is available to the nodes. Hence, we cannot adopt a probabilistic approach as in [6]. Instead, we propose an alternative way to minimize the probability of false routing. Assume \( p \) is a path consisting of links \( 1, 2, ..., h \), satisfying \( \min_{j \in \{1, 2, ..., h\}} \{C^j\} > \rho \) and \( \min_{j \in \{1, 2, ..., h\}} \{B^{a\rightarrow l-1}\} < \rho \), where \( \rho \) is the concave constraint. Assume \( \forall j \in \{1, 2, ..., h\} \), \( p(c^j) > \rho | C^j > \rho, \, B^{a\rightarrow l-1} < \rho \) = \( \delta \). Since \( \min_{j \in \{1, 2, ..., h\}} \{C^j\} < \rho \) and \( \min_{j \in \{1, 2, ..., h\}} \{B^{a\rightarrow l-1}\} < \rho \), there must exist \( j \in \{1, 2, ..., h\} \) such that \( C^j > \rho \) and \( B^{a\rightarrow l-1} < \rho \). Moreover, assume the link metrics on any two links are independent from each other, and there are totally \( \tilde{h} \) links, \( i_1, i_2, ..., i_{\tilde{h}} \), in \( p \) satisfying that \( \forall j \in \{i_1, i_2, ..., i_{\tilde{h}}\} \), \( C^j > \rho \) and \( B^{a\rightarrow l-1} < \rho \) (\( \tilde{h} \) ensured feasible links). The probability for \( p \) to be a real feasible path becomes \( \delta^\tilde{h} \), i.e., the probability for \( p \) to be a real feasible path decreases exponentially with the number of ensured feasible links. Hence, for a concave metric, to find a path with the largest probability to be a real feasible path is to minimize the number of its ensured feasible links. Theoretically, if we fail to find a feasible path consisting only of ensured feasible and unsure feasible links, it is still possible to find a real feasible path that includes some unsure infeasible links. However, since the probability for the path (with unsure infeasible links) to be a real feasible path is relative small and false positive routing wastes network resources, such paths are not preferable.

Next, we show how to compute the path with the largest probability to be a feasible path for the case of an additive metric. Similar to [6], let \( \xi = \sum_{j=1}^h c^j \), and \( \xi \) can be approximated by Central Limit Theorem (CRT). Let the mean and variance of \( \xi \) be \( u \) and \( \sigma^2 \), respectively. Hence,

\[
\sigma^2 = \sum_{j=1}^h \sigma_j^2,
\]

(7)

and

\[
u = \sum_{j=1}^h u_j \approx \sum_{j=1}^h C^j = \beta,
\]

(8)

where \( u_j \) and \( \sigma_j \) are the mean and variance of \( c_i^j \) given \( C^j \), respectively. The probability for path \( p \) to be a real feasible path is approximately

\[
p(\sum_{j=1}^h x_j < \rho) \approx \int_{-\infty}^{\rho-\beta} \frac{1}{2\pi\sigma} e^{-\frac{x^2}{2\sigma^2}} dx.
\]

(9)

Many works [6], [16]- [17] have been reported in the literature to address the problem of finding the Most Probable Delay Constraint Path (MP-DCP). Here, since no stochastic knowledge is available, we cannot adopt such approach. By Eq. 9, the probability for \( p \) to be a real feasible path is only related to \( \rho - \beta \) and \( \sigma \). Note that we cannot compute \( \sigma \) (no stochastic link state knowledge). We can maximize the probability by simply maximizing \( \rho - \beta \) or minimizing \( \beta \) (the given constraint).

Based on the above analysis, our proposed framework consists of two parts:

- In the first part, we try to find a real feasible path in \( G(N, E) \). By Theorem 1, the path computed in the first part must be a feasible path.
- In the second part, we compute the heuristic most probable feasible path.

In the first part, since the links not satisfying the concave constraints can be pruned, our problem is converted to finding a path satisfying the additive constraint(s). If there is only
Given a link, which is either an ensured feasible link, an unsure feasible link, or an unsure feasible link, and the cost of a path is the sum of the cost of the path consisting of the three links is the sum of the link costs.

As shown in Fig. 2, two concave metrics, (1, 2, 3) and (1.2, 2.1, 3.5), (1, 2, 3) and (0.5, 0.8, 1.2), and (0.6, 0.8, 1) and (0.5, 0.8, 1.2), are respectively associated with links $e(a, b)$, $e(b, c)$, and $e(c, d)$. Assume the concave constraints are both 0.7. By the above definition, the costs of $e(a, b)$, $e(b, c)$, and $e(c, d)$ are 0, 1, and 2, respectively, and the cost of the path consisting of the three links is the sum of link costs, 3. As a result, we can convert the problem of finding the most probable concave constrained path into finding the least cost path with the above definition. Combining with the additive constraints, our problem becomes finding the least cost path subject to multiple additive constraints. In [5], an efficient algorithm (H_MCOP) for the Multiple Constrained Optimal Path (MCOP) selection has been proposed. However, H_MCOP cannot guarantee 100% success ratio in finding the feasible path.

Remark 1: We have proposed a framework that can be applied to cases where the class-based link state update policy is adopted for updating link state information. Note that we can also apply the framework for cases of other trigger-based link state update policies. For example, assume the threshold update policy is adopted for updating link state information. Given a link $j$ with the current link state metric $C_j$ (assume one metric). By the definition of the threshold ($\tau$) update policy, an update will be triggered at the moment of $t$ if

$$\frac{|c_i(t) - C_j|}{C_j} > \tau \Rightarrow c_i(t) = (1 + \tau)C_j \quad \text{or} \quad c_i(t) = (1 - \tau)C_j. \quad (10)$$

Hence, we can simply set $B^{\alpha - 1}_j = (1 - \tau)C_j$ and $B^{\alpha}_j = (1 + \tau)C_j$, and our proposed framework can be applied. Specifically, for any given trigger based update policy, assume the current link state metrics of link $j$ are $C_1^j, C_2^j, \ldots, C_M^j$, and an update will be triggered when

$$c_i^j(t) = B^{\alpha - 1}_i < C_i^j, \quad i = 1, 2, \ldots, M, \quad (11)$$

or

$$c_i^j(t) = B^{\alpha}_i > C_i^j, \quad i = 1, 2, \ldots, M. \quad (12)$$

Hence, our proposed framework can be applied.

III. SIMULATIONS

For comparison purpose, we develop a routing algorithm, named Heuristic Most Possible Routing (HMPR), based on our proposed framework. We adopt SAMCRA ($\Gamma$) for finding the path subject to multiple additive constraints and flooding as the MCOP solution. The equal-class based update policy is assumed, and all link metrics are partitioned into the same number of classes ($k_1 = k_2 = \ldots = k_M = n_k$). We adopt only two constraints in this simulation: bandwidth and delay. For comparison purposes, we present a Simple Dijkstra Algorithm (SDA) to show that the generic QoS routing algorithm without
considering the staleness of link state information may introduce significant false routing. Similar to most source routing algorithms, in which the links not having enough bandwidth are pruned, SDA simply removes all the unused infeasible and ensured infeasible links, and treat the other links as the ones satisfying the concave constraints. Specifically, assuming the first QoS metric is bandwidth, link $j$ is pruned from the network if $C_1^j < \rho_1$. Then, SDA executes the shortest path search algorithm, the Dijkstra algorithm, to find the least delay path. The network topology is the 32-node network. We assume the traffic load of the network is 0.8 and the available bandwidth is uniformly distributed from 0 to 0.4 (average available bandwidth is 0.2). The delay of every link is uniformly distributed from 0 to 1. In the simulation, the delay constraint is increased from 0.5 to 5.5 by a step of 0.2, and $n_c = 10$.

We first demonstrate, as shown in Fig. 3, the relationship between the probability for a path to satisfy a concave constraint and the number of its unsured feasible links. The probability roughly decreases exponentially with the number of the unsured feasible links; this is consistent with our analysis. Fig. 4 illustrates the false routing probability of SDA with the delay and bandwidth constraints. It can be observed that the impact of the stale link state information on the performance of routing algorithm is very serious, i.e., without considering the staleness of link state information, SDA introduces a significant percentage of false routing, which can be up to 35% in the 32-node network. For comparison purposes, we focus on the case when the bandwidth constraint is less than 0.05, and assume it is uniformly distributed from 0 to 0.05. The simulation results are illustrated in Fig. 5, in which the performance of SDA and HMPR is compared. Note that the false routing probability of HMPR is much lower than that of SDA, i.e., the impact of the stale link state information is minimized by HMPR. The probability of false routing of SDA increases obviously with the network size. HMPR is more scalable than SDA in terms of the false routing probability, i.e., the false routing probability of HMPR almost does not increase with the network size. Note that the false routing probability can be reduced by increasing the accuracy of link state information; this is achieved by increasing the number of classes ($n_c$) in our simulation. However, this approach increases the protocol overhead and burden on the network resource, and is thus not preferable. On the other hand, we can reduce the false routing probability without increasing protocol overhead by adopting our proposed framework in real networks.

IV. CONCLUSIONS

In this paper, we have introduced and investigated the issue of minimizing the impact of stale link state information on the performance of routing algorithms without any stochastic link state knowledge, and have proposed a framework under the assumption that trigger-based link state update policies are adopted. With extensive simulations, we have shown that our framework can effectively minimize the effect of the staleness of link state information. As a result, the false routing probability is greatly reduced.

REFERENCES