Computer Networks 279 (2026) 112121

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Age of information utility-based routing in networks with unreliable links

Abdalaziz Sawwan [a,*], Mingjun Xiao [b], Jie Wu [a]

[a] Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
[b] School of Computer Science and Technology, University of Science and Technology of China, China

ARTICLE INFO

Keywords:
Age of information (AoI)
Multi-hop wireless networks
Unreliable networks
Utility-based routing

ABSTRACT

The investigation of message freshness in multi-hop wireless networks with unreliable links and heterogeneous forwarding costs is crucial for real-time monitoring and control applications. Existing routing protocols for unreliable networks typically optimize hop count, expected transmissions, or simplified time-based utility models, which can overlook how information value decays with staleness at the destination. In this work, we introduce an AoI-sensitive *net-utility* formulation that explicitly couples an update's delivery benefit with an Age-of-Information freshness penalty and cumulative forwarding costs under retransmissions on unreliable links. We consider stochastically periodic message update generation, where inter-generation times are random with mean T, and each update carries an intrinsic benefit b while incurring node-dependent forwarding costs and random additional delay due to link failures. We propose an optimal routing algorithm that maximizes the long-run expected utility by dynamically selecting routes in real time and discarding updates whose anticipated utility contribution does not justify their remaining expected cost. By deriving a closed-form expression for the expected utility attrition of repeatedly attempting an unreliable link until first success, we reduce the routing decision to a shortest-path computation with analytically computed edge weights and an online forwarding/drop rule. When a link attempt fails (detected after a random failure-detection time), the transmitter incurs an additional forwarding cost and an additional freshness loss proportional to the elapsed waiting time. In our baseline analysis and algorithm, we adopt a standard link-layer Automatic Repeat reQuest (ARQ) rule that immediately retries the *same* next hop until first success. We also discuss how the same shortest-path reduction extends to non-exponential empirical failure-time distributions by replacing only a per-link conditional mean term. Finally, we conduct simulations on three representative graphs to evaluate the effectiveness of the proposed algorithm under varying reliability, cost, and AoI-sensitivity parameters.

1. Introduction

Timely information delivery is a first-order requirement in many multi-hop wireless systems, including cyber-physical monitoring and control, vehicular and aerial sensing, emergency and disaster response, and industrial IoT. In these settings, routing decisions are constrained not only by path length or energy expenditure, but also by (i) link unreliability, which induces retransmissions and random additional delay, and (ii) heterogeneous forwarding costs across intermediate nodes due to energy, congestion, or operational incentives. As a result, a routing strategy that is "good" for one update may be suboptimal for the next update, because the *value of delivering an update* depends on how stale the destination currently is and how soon a newer update is expected to arrive.

The *Age of Information (AoI)* has emerged as a principled metric for quantifying destination-side freshness; it measures how long it has been since the generation time of the most recently received update [1,2]. A large body of AoI research focuses on scheduling, sampling, and transmission policies that *minimize* average/peak AoI under queueing and interference constraints [3]. Complementary multi-hop studies examine how routing and link activation affect AoI and throughput tradeoffs [4]. However, in many operational networks, the objective is not AoI minimization *per se*: operators often care about an explicit trade-off between freshness *and* the cost paid to deliver updates (energy, payments, or resource consumption), and they may rationally prefer to *discard* an overly delayed update if a newer update is imminent.

Motivated by this gap, we study *utility-based routing* for stochastically generated status updates in *unreliable* multi-hop networks. Our key modeling choice is to optimize a *net utility* that explicitly couples (a) a message's intrinsic benefit upon delivery, (b) an AoI-sensitive freshness penalty that increases with the destination's current AoI at delivery time, and (c) the cumulative forwarding cost incurred along the route,

* Corresponding author.
E-mail address: sawwan@temple.edu (A. Sawwan).

https://doi.org/10.1016/j.comnet.2026.112121
Received 9 March 2025; Received in revised form 21 January 2026; Accepted 14 February 2026
Available online 16 February 2026
1389-1286/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

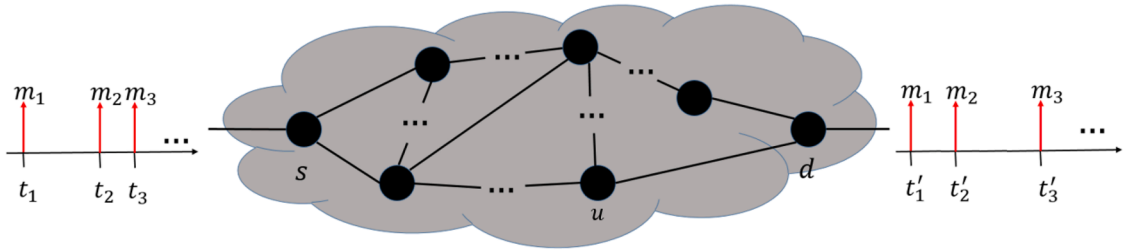


Fig. 1. The model of our problem. The routing network has unreliable links. The stochastically periodic message updates are generated at the source at t_i, where $(t_i - t_{i-1}) \sim X$, and received at t'_i, where X is the distribution that the temporal lag between consecutive messages follows, where $\mathbb{E}[X] = T$.

including the cost and delay of retransmissions under link failures. Critically, the utility of an update is evaluated in the *context of the update stream*: it depends on the time of the *previous successfully received* update and on the expected time scale of the *next* update arrival (via the stochastic update period). This differs from standard routing formulations that treat packets independently and from AoI-minimization formulations that do not attach explicit per-update benefits/costs.

From an algorithmic standpoint, we show that this AoI-sensitive net-utility maximization admits an *efficient* optimal solution under complete statistical knowledge of link reliability and delay. We derive a closed-form expression for the *expected utility attrition* of attempting an unreliable link until first success, under a failure-time model that captures the empirical phenomenon that failures are detected after a random waiting time. This converts the stochastic routing problem into a shortest-path problem on a graph with analytically computed edge weights, enabling the use of classical polynomial-time shortest-path routines. Importantly, the resulting routing rule is *online*: intermediate nodes can decide to continue forwarding or to drop the current update when its remaining expected net contribution becomes negative.

Feasibility in real networks is supported by a lightweight feedback mechanism: upon reception, the destination broadcasts a short acknowledgment carrying only the update sequence ID, allowing intermediate nodes to track the most recently delivered update without maintaining large state. Because this feedback is small and can be prioritized, it is reasonable in low-rate status-update regimes to assume it propagates faster than the update generation period. Under this standard "single-in-flight update" regime, each update can be routed with a fresh view of the destination's latest received sequence. Fig. 1 shows an illustration of the problem.

The novelty of this work is not merely using AoI as a performance metric, but *embedding AoI into a cost-benefit utility* and deriving an *optimal* routing rule for unreliable multi-hop networks under that objective. Concretely, we make the following contributions:

- AoI-sensitive net-utility model for unreliable multi-hop routing. We formalize a utility that rewards delivery benefit while penalizing (i) freshness loss via destination AoI and (ii) cumulative forwarding cost, including retransmission-induced increments under link failures. The model is tailored to stochastically periodic update generation and explicitly accounts for the update-stream context (previous delivered update and the expected time scale of the next update).
- Reduction to a shortest-path problem via expected utility attrition. We derive a closed-form expected utility reduction for an unreliable link when it is retried until first success, yielding analytically computable edge weights. This converts the stochastic utility-maximization problem into a dynamic shortest-path computation with polynomial complexity.
- Online optimal routing with principled dropping. Building on the optimal path structure, we provide an online forwarding rule that may drop an update mid-route when its remaining expected net utility becomes negative. This directly addresses scenarios where delivering an overly stale update is not worthwhile because a newer update is expected soon.
- Evaluation on representative topologies. We validate the proposed scheme via simulation on sparse, dense, and ARPANET-like graphs, demonstrating how unreliability, costs, and AoI-sensitivity jointly shape routing decisions and achieved utility.

The remainder of the paper is organized as follows. In Section 2, related works are reviewed. In Section 3, the general model of the problem is proposed. In Section 4, the optimal solution for the general case of unreliable networks is presented. Section 5 presents the simulation results to verify the efficiency of the algorithms. Finally, Section 6 gives the conclusion.

2. Related work

We position this paper relative to three main research threads: (i) reliability- and cost-aware routing metrics for multi-hop wireless and sensor networks, (ii) utility-based routing with time-sensitive objectives, and (iii) AoI theory and AoI-centric optimization (including multi-hop settings). Our contribution sits at their intersection: we embed AoI into a per-update cost–benefit utility in an *unreliable* multi-hop routing problem and derive an *optimal* (in expectation) shortest-path structure with an online forwarding/drop rule.

2.1. Reliability- and cost-aware routing metrics

Classical multi-hop routing work optimizes structural proxies such as hop count or additive transmission costs, which are often insufficient under lossy links and heterogeneous forwarding costs. Early ad hoc and sensor-network routing protocols emphasize feasibility and distributed operation under mobility and limited state [5], while subsequent work proposed objective-driven routing designs such as maximum-lifetime routing and energy-aware forwarding [6]. Related families of schemes incorporate link reliability through path lifetime or stability notions [5,6], or through *bottleneck-link* criteria that maximize the weakest-link reliability on a path [7]. These approaches capture aspects of link quality, but do not directly model how retransmission-induced delay affects the *value* of delivered information, nor do they incorporate a destination-side freshness state.

A widely used reliability-aware metric is the Expected Transmission Count (ETX/ETC) and its variants, which estimate the expected number of (re)transmissions required for successful delivery [8,9]. MintRoute similarly incorporates measured link delivery ratios for robust data collection in sensor networks [10], and follow-on work studies its security and deployment considerations [11]. These metrics are closely related to our analysis in that they normalize link "cost" by success probability. However, ETX/ETC-style objectives typically target throughput or delivery efficiency and do not *couple* retransmission-induced delay with a destination freshness penalty or an explicit per-update benefit. In contrast, our formulation assigns each update an intrinsic benefit and subtracts an AoI-sensitive freshness loss and cumulative forwarding costs, yielding a route choice that can change from update to update and can trigger principled dropping when continued forwarding is not worthwhile.

2.2. Utility-based and time-sensitive routing

Utility-based routing introduces composite objectives that trade off delivery gain against resource expenditure. Opportunistic and utility-based ad hoc routing has been used to balance reliability and energy/cost, often prioritizing higher-reliability paths for higher-value packets [12,13]. In delay-/disruption-tolerant and intermittently connected regimes, utility-driven designs select forwarding opportunities based on estimated delivery benefit under contact uncertainty [14] and exploit social structure to improve delivery probability [15,16]. A complementary line considers *time sensitivity* directly via utility decay with time or deadlines: examples include time-sensitive opportunistic utility-based routing in DTNs [17], time-sensitive single-copy routing under low duty cycles [18], time-sensitive duty-cycled sensor routing under unreliable links [19], deadline-sensitive routing in cyclic mobile social networks [20], and other utility-based models with location- or context-dependent utility [21,22]. These works motivate the importance of time sensitivity, but they typically employ a packet-centric or linear time-decay objective and do not explicitly incorporate the destination's AoI state or the update-stream context (previous delivered update and expected time-scale of the next update) that is central in our net-utility definition.

2.3. AoI foundations and AoI-centric optimization

AoI has emerged as a principled freshness metric capturing how long the destination has remained uninformed since the generation time of its latest received update [23,24]. Foundational work formalized AoI-based performance measures and optimal update-frequency tradeoffs in status-update systems [2], and surveys provide a broad taxonomy of AoI models, metrics, and optimization approaches [1,25]. Much of the AoI literature seeks to *minimize* average/peak AoI through scheduling, sampling, or transmission policies under queueing and interference constraints [3], and it has expanded to diverse settings such as random access [26,27], opportunistic scheduling and channel access [28,29], multi-source queueing models [30], service preemptions and request delays [31], and energy-harvesting update systems [32]. AoI has also been extended beyond pure communication delay to incorporate processing/offloading decisions [33] and the effects of network coding [34].

Within multi-hop networks, AoI introduces additional structure because routing and link activation affect both delivery times and the destination's freshness evolution. Prior work studies AoI/throughput tradeoffs in routing-aware multi-hop wireless networks [4], and AoI-aware DTN routing strategies have also been explored [35]. These AoI-centric studies generally optimize AoI (possibly with throughput or access constraints) and typically forward all generated updates according to the chosen scheduling/routing policy. Our objective differs: we optimize *long-run net utility* that explicitly includes (i) a per-update delivery benefit, (ii) an AoI-sensitive freshness loss evaluated at delivery time, and (iii) cumulative forwarding cost including retransmissions on unreliable links. This change in objective produces qualitatively different behavior-most notably, the ability to *drop* an update mid-route when its remaining expected net contribution is negative.

2.4. AoI-aware utility objectives and closest comparisons

Hybrid objectives that mix freshness with other system costs/benefits appear implicitly across several recent AoI application domains (e.g., access control and scheduling under constraints [26–29], energy-limited updates [32], and queueing/service constraints [30,31]), but these formulations are usually not expressed as a per-update cost–benefit *routing* utility with node-dependent forwarding costs and explicit retransmission modeling. On the routing side, prior utility-based models combine link cost and reliability to maximize an expected utility metric [13], and time-sensitive utility-based routing under unreliable links has been studied with simpler time factors [17–20]. Our work differs in two key ways: (i) the time sensitivity is *AoI-grounded* and evaluated relative to the destination's freshness state [1,23,24], and (ii) the update utility is evaluated in the *update-stream context* (previous delivered update and expected next-update timescale), rather than treating packets independently. A restricted version of our AoI-sensitive routing perspective was introduced for deterministically periodic generation [36]; here we generalize to stochastically periodic generation and derive a closed-form *expected utility attrition* for unreliable links retried until success, enabling a shortest-path reduction with analytically computed edge weights and an online forwarding/drop rule. Related AoI-graph structuring ideas such as edge pruning under AoI considerations [37] further motivate graph-level approaches, but do not provide the same net-utility objective and shortest-path optimality structure we establish for unreliable multi-hop routing with retransmission costs.

2.5. Unreliability modeling and algorithmic perspective

A distinguishing feature of our model is that link failures are detected after a random time (rather than instantaneously), and retransmissions therefore incur both additional cost and additional delay that directly degrades freshness at delivery. Exponential-time failure models are standard in reliability and failure-time analysis [38,39], and they yield tractable closed forms that we exploit to compute expected utility attrition per link. Algorithmically, our reduction leverages classical shortest-path methods: Bellman–Ford-style routines (including improved variants) [40] and Floyd–Warshall-style all-pairs methods (with attention to pathological cases such as negative cycles) [41]. Finally, while our appendix uses standard probabilistic tools to compute the expected trials to first success (a geometric/Bernoulli-trial fact), generating-function techniques are a common and reusable analytic approach in reliability and optimization settings [42].

2.6. Summary

Taken together, existing work either (a) optimizes routing metrics that reflect cost/reliability but do not model destination freshness [5,7–11], (b) uses utility-based routing with time sensitivity not explicitly tied to AoI [12–22], or (c) optimizes AoI directly under scheduling/access/queueing constraints, including emerging multi-hop AoI studies [1,3,4,23–35]. Our work bridges these threads by embedding AoI into a per-update cost–benefit utility for unreliable multi-hop routing, deriving closed-form link-level expected utility attrition under failure-time uncertainty, and yielding an optimal shortest-path structure with an online forwarding/drop mechanism.

3. Proposed model

In this section, we show the model of the AoI-based utility that is attributed to the stochastically periodic message updates. Furthermore, we discuss the model of an unreliable network that is used to deliver the messages from the source node s to the destination node d.

3.1. Age of information

Consider a source-destination system. Let t_i be the times at which a message update is generated at the source node s. Let t_i' be the times at which this message is received by the destination node d. First, we need to define that at any time t, the function $N(t)$ would be the index of the most recently received message at the destination node. Thus the following equation defines $N(t)$:

$$N(t) = \max\{i | t_i' \leq t\} \tag{1}$$

Now, we simply define the AoI of the message at any time t. The AoI of a message generated at the source node s to be delivered to the destination node d is given as follows [23]:

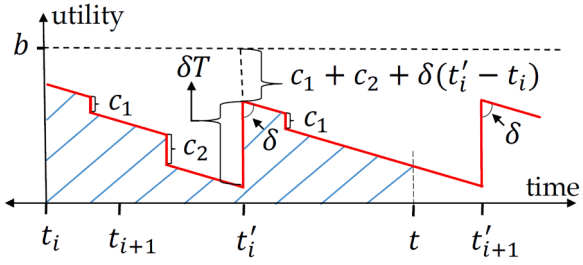


Fig. 2. An illustration showing the utility function over the transmission for two consecutive periodic messages. The red linear segments have slope $-\delta$ (utility decays at rate δ per unit time while AoI increases at unit rate); the δ callouts indicate this decay rate, not a geometric angle. The average height of the shaded area is the expected utility at time t. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\Delta(t) = t - t_{N(t)} \tag{2}$$

Unless otherwise stated, AoI refers to the instantaneous age defined in (2), while performance is evaluated through long-run time averages as in (5).

3.2. The AoI-based utility model of messages

In our model, message generation occurs in a stochastically periodic manner, with a temporal lag between consecutive messages sampled from some random distribution X that is characterized by its average $\mathbb{E}[X] = T$. Each message update carries an intrinsic beneficial reward b, which is the reward in case of the arrival of the message from the source node s to the destination node d. However, there is a cost charged at each intermediate node (all nodes but s and d) for forwarding the message. We will represent the total cost charged on the message utility by the network nodes using C. Furthermore, the total utility of the message is linearly proportional to the age of the message (i.e. the freshness of the message). This means that higher AoI values of the message ($\Delta(t)$) make the total utility decrease linearly. In general, we set a factor of δ to represent the proportion by which the AoI affects the total utility with respect to the total forwarding cost C and beneficial reward b. This factor of δ is set depending on the importance of the freshness of the message with respect to its intrinsic beneficial reward of arrival b and the total cost of forwarding C. Hence, the construction of the utility function has to follow the term:

$$b - \delta\Delta(t) - C \tag{3}$$

This utility needs to be lower-capped by $-C$. Applying that, we end up with a time-sensitive AoI-based utility model of messages $u(t)$ shown in Eq. (4).

$$u(t) = \max\{-C, b - \delta\Delta(t) - C\} \tag{4}$$

We notice that once the value of the proportion of the message age penalty $\delta\Delta(t)$ becomes high enough to reach the threshold of the beneficial reward b, it would be better to discard the message after incurring a cost of C rather than delivering the message. Fig. 2 illustrates the AoI-based utility model of messages. Between receptions, $\Delta(t)$ increases at unit rate, so on the unclipped linear segment the utility decreases linearly in time with constant slope $-\delta$ (i.e., δ is the magnitude of the utility-decay rate per unit time). In Fig. 2, the δ callouts annotate this decay rate rather than a geometric angle. The utility function is lower-capped by the value of $-C$.

Finally, the routing objective is to maximize the long-run time-average utility induced by (4), i.e., (5). This formulation captures steady-state performance rather than a finite-horizon objective.

$$\text{maximize} \lim_{t\to\infty} \frac{1}{t} \int_0^t u(\tau) d\tau \tag{5}$$

Remark (utility maximization vs. AoI minimization). Eq. (5) optimizes a *utility* that includes an AoI-sensitive freshness penalty and explicit forwarding costs, rather than optimizing AoI alone. This distinction is central: minimizing AoI treats all successful deliveries as beneficial and typically targets age metrics directly, while our formulation allows the network to trade freshness against cost and to drop updates whose expected net contribution is negative. Consequently, our "optimality" claims throughout the paper are with respect to maximizing the time-average expected utility in Eq. (5) under the stated network and information assumptions.

3.3. The model of the network

We consider an unreliable network represented by graph G that is composed of the nodes set V, which includes the source node s and the destination node d, with their associated links E. Those nodes are linked with probabilistically unreliable links. The probability that a message transmits successfully in the link between nodes i and j is denoted by $p_{i,j}$. N_i denotes the neighbor set of node i, and $j \in N_i$.

In case the link between nodes i and j succeeds in transmitting the message, the message would take exactly $\tau_{i,j}$ time units. However, in case the link between nodes i and j fails to transmit the message (which would occur with a probability of $(1 - p_{i,j})$), the failure at the link doesn't happen immediately once the node forwards the message. The failure happens after some time that follows some distribution. In our model, we assume a truncated exponential failure-detection time with mean parameter $\lambda_{i,j}$, which can be characterized in terms of $\tau_{i,j}$ and $p_{i,j}$. This choice is justified as exponential distributions arise naturally in the context of failure time [37–39]. In case of failure, the node may try the same link again immediately with the same previous success chance $p_{i,j}$.

3.3.1. Retransmissions and how delay/cost increments are quantified.

We model a single transmission *attempt* over link (i, j) as follows: with probability $p_{i,j}$ the attempt succeeds and consumes a deterministic time $\tau_{i,j}$; with probability $1 - p_{i,j}$ the attempt fails and the failure is detected after a random time $F_{i,j} \in [0, \tau_{i,j})$. Each attempt incurs the forwarding cost c_i regardless of success or failure. After a failure is detected, node i immediately retries the *same* link (i, j) (ARQ-style) until the first success; across attempts, success/failure outcomes and failure-detection times are assumed i.i.d. Therefore, each additional failed attempt contributes an *extra* utility reduction of $c_i + \delta F_{i,j}$ (the extra forwarding cost plus the freshness loss accrued over the extra waiting time), while the final successful attempt contributes $c_i + \delta\tau_{i,j}$. These are the "additional time delay and cost increments" incurred by retransmissions, and Lemma 1 aggregates them into the expected per-link utility attrition $\mathbb{E}[R_{i,j}]$ used as an edge weight by Algorithm 1.

3.3.2. Adaptability beyond exponential failure times.

While we use a truncated exponential model to obtain closed-form expressions, the shortest-path reduction does not rely on the exponential *shape* per se. For real-world links whose failure-detection times follow other distributions, the same analysis applies as long as we can compute (or estimate from measurements) the conditional mean $\mu_{i,j}^f \triangleq \mathbb{E}[F_{i,j} \mid \text{failure}]$. In that general case, the same geometric-trials argument yields $\mathbb{E}[R_{i,j}] = \frac{c_i}{p_{i,j}} + \delta\left(\tau_{i,j} + \frac{1-p_{i,j}}{p_{i,j}}\mu_{i,j}^f\right)$. Thus, Algorithm 1 remains unchanged: one simply plugs these empirically calibrated $\mathbb{E}[R_{i,j}]$ values as edge weights. The truncated exponential assumption is adopted here because it provides a closed-form $\mu_{i,j}^f$ in terms of $(\tau_{i,j}, p_{i,j})$, yielding the simplified expression stated in Lemma 1.

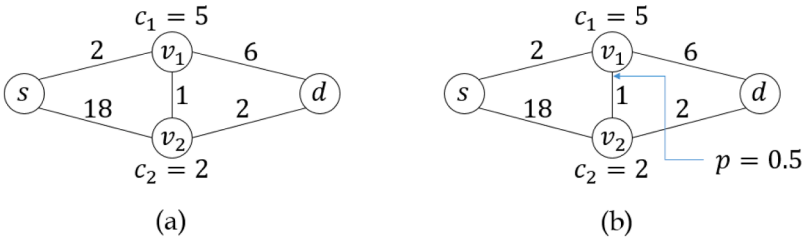
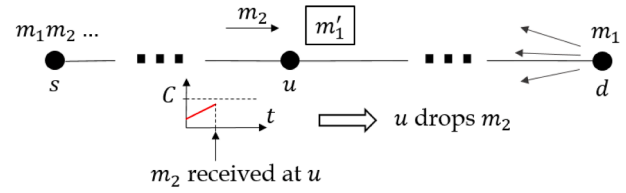
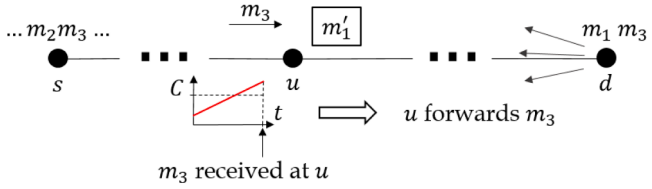


Fig. 3. An example of the problem, where each link between node i and node j has a specific time delay $\tau_{i,j}$. Each intermediate node v_i has its forwarding cost c_i. (a) has all links reliable. (b) has all links reliable except the middle one where $p_{1,2} = 0.5$..



(a) The case where the expected uptick of utility is lower than C when a message arrives at node u, hence the message is dropped.



(b) The case where the expected uptick of utility is higher than C so the message is forwarded.

Fig. 4. An illustration of the optimal routing protocol. Feedback messages m_1' and m_3' are flooded back into the network (only m_1' is illustrated).

Since the time taken by the link between nodes i and j in case of failure follows the exponential probability distribution, the cumulative distribution function (CDF) would be:

$$F(t) = 1 - e^{-t/\lambda_{i,j}}, \quad 0 \le t < \tau_{i,j} \tag{6}$$

In addition, since the probability that the link takes time less than $\tau_{i,j}$ equals the probability of failure, which is $(1 - p_{i,j})$, the value of the CDF at $\tau_{i,j}^-$ would be the same as this failure probability. Hence, $\lambda_{i,j}$ can be characterized as shown in Eqs. (7)-(8).

$$F(\tau_{i,j}^-) = 1 - e^{-\tau_{i,j}/\lambda_{i,j}} = 1 - p_{i,j} \tag{7}$$

$$\lambda_{i,j} = -\tau_{i,j} / \log p_{i,j} \tag{8}$$

where the log represents the natural logarithm with the base e. It is worth mentioning that in our model, we consider that the consecutive events of trials using a certain link to attempt delivery of a message are independent from each other.

Furthermore, we consider a forwarding cost at each node i to be denoted by c_i. This forwarding cost would be incurred each time the node tries to forward the message regardless of whether the message was successfully or unsuccessfully delivered. Lastly, in the model of the network that we have, whenever a message update m_i is received at the destination node d, it broadcasts back into the network a short timestamped acknowledgment which includes only an update sequence ID m_i'. We also assume that the message and acknowledgment transmission time is much shorter than the average message update period T. Hence, the network handles one message update at a time.

Algorithm 1: Determining the optimal path.

Require: G, δ, T.
Ensure: Minimum cost $C + \delta\Delta(t)$ from s to d. //i.e., maximum utility.
Initialization: $\forall(i,j) \in E$, set distance$(i,j) = \mathbb{E}[R_{i,j}]$ based on Eq. (9).
1: Invoke Bellman–Ford$_\delta(G)$.
2: **for** $i \in V$ **do**
3: Get shortest distance $D(s,i)$.
4: **return** both shortest distance $D(s,d)$. and the shortest path, which is $\pi[d], \pi[\pi[d]], \ldots$.

4. The optimal solution of the problem

In this section, we provide the optimal solution for the general unreliable problem and prove the optimality of our proposed solution. We assume that the information is always complete.

4.1. Evaluating the expected utility reduction

First, because of the probabilistic nature of our problem, we consider that the optimal solution is the solution that chooses the path of the message updates generated at the source node s to be delivered to the destination node d through the network (or chooses to drop the message), where this choice guarantees that the expected value of the utility $\mathbb{E}[u(t)]$ is maximized, where the notation $\mathbb{E}[.]$ represents the expected value.

Since the objective is to maximize the expected utility value, we can direct our focus on minimizing the expected reduction of utility instead. We denote the expected reduction of utility when the message update is forwarded from node v_i to node v_j by $\mathbb{E}[R_{i,j}]$. Lemma 1 gives the accurate characterization of the expected reduction of utility $\mathbb{E}[R_{i,j}]$.

Lemma 1.
Given two nodes v_i and v_j with success probability $p_{i,j}$ and transmission time $\tau_{i,j}$ for the unreliable link between them, and given the forwarding cost c_i, the utility reduction for transmitting a message update from node v_i to node v_j is given by:

$$\mathbb{E}[R_{i,j}] = (1 - p_{i,j})(-\tau_{i,j})\delta/(p_{i,j} \log p_{i,j}) + c_i/p_{i,j} \tag{9}$$

Proof. The proof of the lemma is in the appendix. □

4.2. Optimal path algorithm

Algorithm 1 shows the way to return the path that has the minimum expected total attrition in utility, which is the "shortest path". $D(s,i)$ in the algorithm represents the expected "distance" from the source node

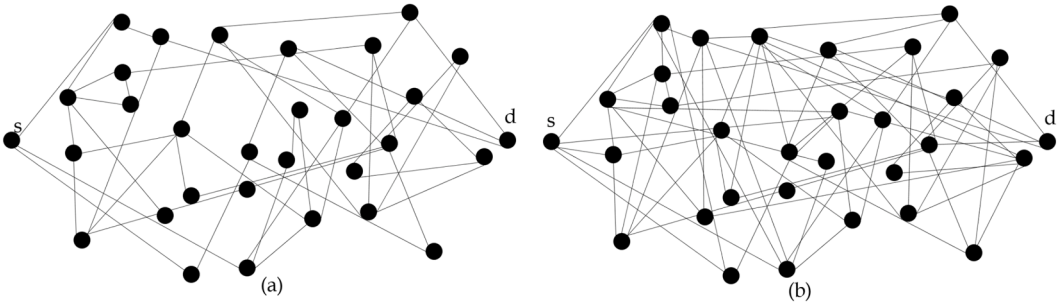


Fig. 5. The topologies tested in the simulation (a) is the sparse topology and (b) is the dense topology.

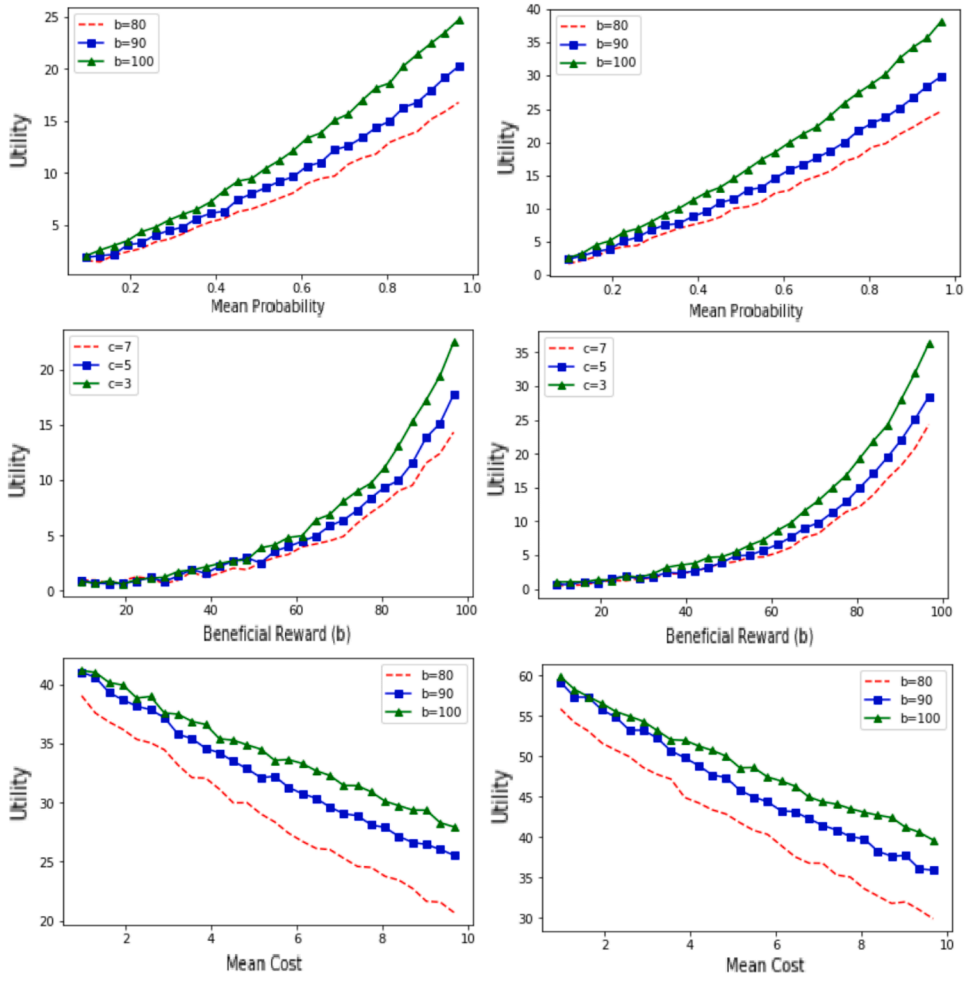


Fig. 6. The simulation results from running Algorithm 2 when $\delta = 1$. The left column is the result after applying the algorithm on the sparse topology, and the right column is result after applying the algorithm on the dense topology.

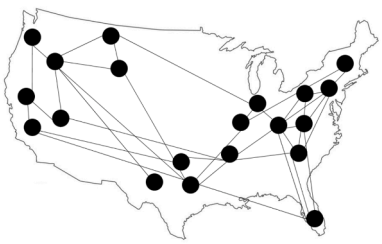


Fig. 7. Backbone ARPANET topology. The reliability and cost values are set to depend on the distances.

s to the node i. This distance is the expected reduction of the total utility for a certain path taken from s to i. The node $\pi[i]$ represents the predecessor of node i in the optimal path evaluated.

Theorem 1. *The path produced by Algorithm 1 is the optimal path that maximizes the expected AoI-sensitive utility. The time complexity is* $O(|V| \times |E|)$.

Proof.

Our efficient optimal solution is based on choosing the path that minimizes the total reduction of the utility $(\delta\Delta(t) + C)$. Furthermore, since the links of the network are unreliable and succeed only at a certain probability, the optimal solution will be based on the expected total reduction of the utility if a link is used over and over again until it

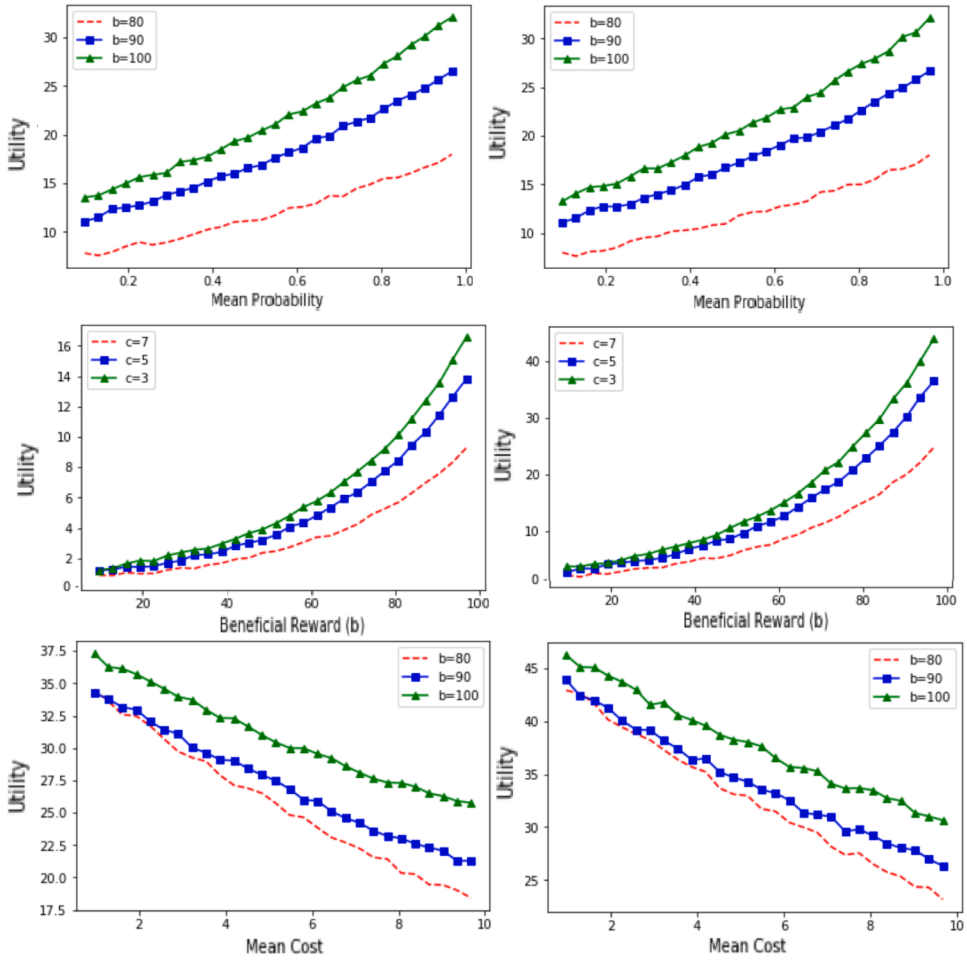


Fig. 8. The simulation results from running Algorithm 2 when $\delta = 2$. The left column is the result after applying the algorithm on the sparse topology, and the right column is the result after applying the algorithm on the dense topology.

succeeds in delivering the message, considering the exponential probability distribution of the time taken in case of failure.

Considering the utility reduction derived in Lemma 1 as the "distance" between nodes v_i and v_j, our optimal path would be the "shortest path" between the source node s and destination node d considering the normalizing factor δ. Algorithm 1 simply uses the Bellman-Ford algorithm to produce the optimal path for the message to be routed through in the network, where this path minimizes the expected attrition in utility (i.e. maximizes the expected utility).

Regarding the time complexity of Algorithm 1, it directly comes from the time complexity of the Bellman-Ford algorithm which has a total time complexity to determine the optimal path to be $O(|V| \times |E|)$. □

It is worth mentioning that any shortest path algorithm, that evaluates the shortest distance between the source node and all other nodes, would work instead of the Bellman-Ford algorithm that is used in line 1. For example, a centralized Floyd-Warshall algorithm [40,41] would suffice to evaluate all distances needed. However, the time complexity needed in case this algorithm is used will be $O(|V|^3)$.

To illustrate the dynamic selection of paths, we consider the example shown in Fig. 3 (a) in which the links are reliable to simplify the example. The optimal path would depend on the δ for each update, and hence, will change for each one of the message updates depending on its δ value. Let's consider the cases when the value of δ is $0.1, 0.5$, and 1.0. For the case where $\delta = 0.1$, the optimal path that guarantees the least reduction possible in utility is $s \to v_2 \to d$. The reason is that for smaller values of δ, the utility becomes more sensitive to forwarding cost and less sensitive to delay-induced freshness loss. Hence, the algorithm would choose the path with the lowest total cost. Secondly, to consider the case where $\delta = 0.5$, we find that the optimal path that has the least utility reduction would be the path $s \to v_1 \to d$. The last case of $\delta = 1.0$ yields the path $s \to v_1 \to v_2 \to d$. The optimal path is that because when the value of δ is high, the total time delay gets more weight compared to the total cost to calculate the reduction in utility. Therefore, the optimal path would be the one with the least time delay.

A more interesting case is in Fig. 3 (b) where we consider the link between v_1 and v_2 to be unreliable with probability $p_{1,2} = 0.5$. In this case, in order to evaluate the expected reduction of utility going from v_1 to v_2, it's needed to evaluate the expected number of trials until success if we choose to use that path. As derived in the proof of Lemma 1, we can plug the values we have to get the expected utility reduction, which would be $(1.44\delta + 10)$. Considering this value as the distance from v_1 to v_2, it will be used to evaluate the optimal path for different δ values in this case with the unreliable link.

In Algorithm 2, each iteration of the **while** loop corresponds to a *single transmission attempt* from *current* to its successor on the selected path P. If the attempt fails after waiting time t (a realization of $F_{current,\text{succ}}$), then *current* remains unchanged and the incurred increment is $x \leftarrow x + c_{current} + \delta t$, so the next loop iteration retries the same successor link. If the attempt succeeds, then the increment is $x \leftarrow x + c_{current} + \delta\tau_{current,\text{succ}}$, and *current* advances to the successor. This makes the "additional time delay and cost increment" under retransmissions explicit in the online drop decision in line 6.

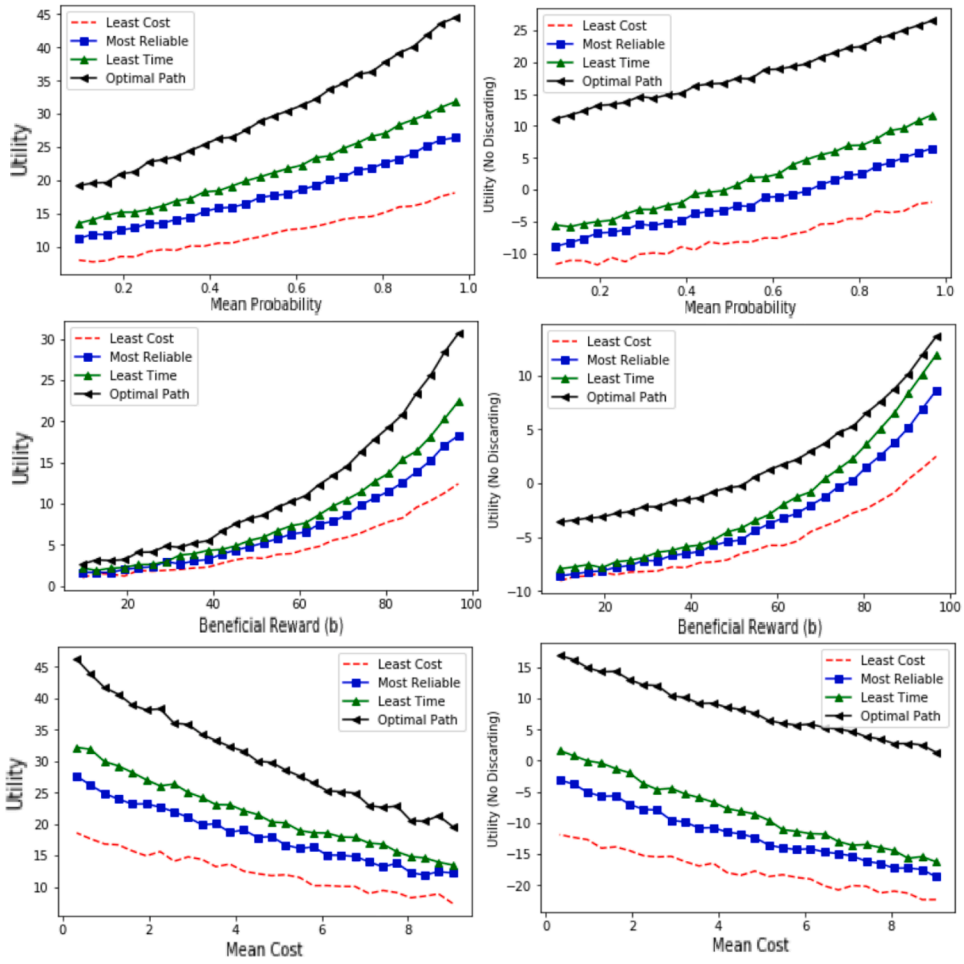


Fig. 9. The simulation results from running the different algorithms when $\delta = 1$. The left column is result after running the algorithms with discarding, the right column forces the algorithms to deliver without discarding. The backbone ARPANET topology is applied.

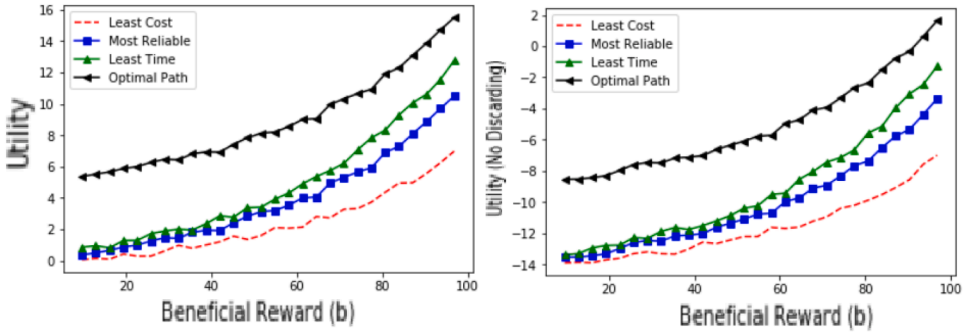


Fig. 10. The simulation results from running the different algorithms when $\delta = 1$. The left column is result after running the algorithms with discarding, the right column forces the algorithms to deliver without discarding. The sparse topology is applied.

Lastly, we need to consider the anticipated immediate decrease of the AoI once the message arrives at the destination node d. Since we are considering a generation of messages at the source node with an average delay of T between consecutive messages, the average of this reduction in the AoI of the message will be exactly T as well. We utilize that fact by imitating this immediate decrease of the AoI value by using a negative cost incurred by the destination node. This will result in setting the cost incurred by the destination node d to equal $c_d = -\delta T$. This will apply in the realistic case in which the total time the message takes to be delivered in the network is less than the average time difference by which the messages are generated; this is the case that we address in our algorithm.

However, if the value of T is large enough, we will have the general expression of c_d shown in the following equation:

$$c_d = \max\{-\delta T, -\delta \times ((1-p_{s,1})/(-p_{s,1}\log p_{s,1}))\tau_{s,1} + \dots + ((1-p_{k,d})/(-p_{k,d}\log p_{k,d}))\tau_{k,d})\} \tag{10}$$

where the path taken would be $s \to v_1 \to \cdots \to v_k \to d$. Practically, T is much smaller than the threshold at which $c_d = -\delta T$.

Algorithm 2: The optimal routing.

Require: G, δ, b, T.
Ensure: Minimum cost $C + \delta\Delta(t)$ from s to d. //i.e., maximum utility.
1: Set P = the optimal path calculated from Algorithm 1.
2: Get $\mathbb{E}[D_{(|V|-1)}[i]] \quad \forall i \in P$ from Algorithm 1.
3: Set $x = 0$ //stores the utility decrease incurred so far.
4: Set $current = s$ //stores the location of the message.
5: **while** $current \neq d$ **do**
6: **if** $x + D(s,d) - D(s, current) < b$ **then**
7: Drop the message.
8: Try sending the message to $current$'s successor in P.
9: **if** success **then**
10: Update $current$ to $current$'s successor in P.
11: Update x depending on the utility reduction incurred.
12: **terminate**.

4.3. Optimal routing algorithm

Now, we present the algorithm that utilizes the optimal path produced by Algorithm 1 in order to optimally route the message throughout the network. Algorithm 2 shows the optimal routing algorithm. Having a stream of messages generated at the source node s, each one of the messages should be routed using the routing scheme in Algorithm 2. The algorithm ensures that the optimal path that guarantees minimized utility decrease is the path that the message would follow in the case that it would not be dropped.

In our model, we introduce a streamlined feedback mechanism to enhance the efficiency of message routing, as depicted in Fig. 4. The source node initiates the process by generating messages at an expected interval of T. Upon the successful delivery of a message to the destination node d, a feedback message is rapidly disseminated back through the network. This feedback, containing only the essential data of message ID and status, is designed to traverse the network with increased speed and reliability compared to the original message.

The assumption that feedback travels significantly faster than regular messages is grounded in several key factors. Firstly, the reduced size and simplicity of the feedback message-limited to mere identification and status information-greatly diminishes its transmission time and resource requirements. Unlike full-sized messages, these feedback packets do not suffer from the same level of delays due to processing or queuing in intermediate nodes. Secondly, we assume that the network infrastructure is optimized to prioritize these feedback messages, ensuring their swift propagation. This could be implemented through dedicated control channels or routing protocols that recognize and expediently handle such feedback.

An important aspect to consider is the network's approach to handling messages. Given that the network processes messages sequentially, one at a time, it becomes a mild assumption to presume that the feedback, which is lighter and simpler in nature, will be propagated back to all network nodes before the arrival of the next message update. This assumption is based on the operational characteristic of the network that prioritizes immediate message handling without concurrent transmissions, thereby reducing the likelihood of congestion and delay.

The sequential processing ensures that each message and its corresponding feedback are treated as discrete events within the network's operation. Since feedback messages are notably smaller and less complex than the original messages, they require significantly less time for transmission and processing. This operational feature of the network, combined with the expedited nature of feedback propagation, supports the assumption that the feedback will reach all relevant nodes in the network well before the next message update is generated.

This assumption considers the operational mechanics of the network and is also necessary for the effectiveness of AoI-based routing algorithms. It allows for a timely update of the routing decisions based on the most recent feedback to ensure that each subsequent message is routed in the most efficient manner possible. This takes into account the latest network conditions and message statuses. This dynamic adaptability is key to maintaining the high efficiency and reliability of message delivery in an unpredictable and variable network environment. By incorporating this assumption into our model, we enable a more responsive and adaptable routing strategy. That is vital for maintaining the timeliness and relevance of information in dynamic and unreliable network environments, such as those encountered in emergency response scenarios.

Furthermore as seen in Fig. 4, nodes in between store the last feedback they got. Every time an intermediate node gets a new message to forward, the node performs the calculations of the expected uptick of utility and how much, on average, it'll cost to forward the message toward the end (for simplicity we call this cost C as well). Depending on this calculation, the node chooses to either send the message further or just discard it. Fig. 4 (a) gives an example where the expected cost C is more than the expected utility uptick, so the message is discarded. But in Fig. 4 (b), the expected cost is less than the benefit, so the message is forwarded. Storing the last feedback is important to calculate $\mathbb{E}[u(t)]$ given that the message reaches the destination.

Algorithm 2 represents a utility-based routing scheme. i.e, the routing depends on the held utility value of the message, which includes the beneficial reward b. The concept of the algorithm is simple; it checks whether the total expected decrease of utility is larger than the beneficial reward b, this total expected decrease of utility is calculated more accurately as the algorithm proceeds by having an updated value of x, which is the cost already incurred by the path from the source node s to the node $current$.

Thus, the total expected decrease of utility will be the addition of both the expected decrease of utility by the path from the node $current$ to the destination node d, which is represented by $D(s,d) - D(s, current)$, and the cost incurred so far, which is x. This means that as long as the total value of $x + D(s,d) - D(s, current)$ is below the beneficial reward threshold, it would be, on average, better to keep forwarding the message.

Theorem 2. *The routing scheme in Algorithm 2 is the optimal routing scheme that ensures a maximized expected AoI-sensitive utility.*

Proof.

From the proof of Theorem 1, the path that Algorithm 2 uses, in the case it is beneficial to forward the message, is the optimal path that would minimize the total expected decrease in the utility. However, the Algorithm dynamically adjusts its strategy: it either keeps trying to forward the message through the optimal path, or it drops the message, depending on the result of each time a link is tried.

Since, on average, no other path would beat the path produced by Algorithm 1 as proven before, and the routing scheme of the message always considers the most accurate expectation of the decrease of utility, no other routing scheme would be better, on average, than Algorithm 2. □

5. Simulation

5.1. Experimental settings

We evaluate Algorithm 2 under multiple parameter settings on three representative topologies. The topologies are shown in Figs. 5 and 7: Fig. 5(a) represents the sparse topology (results shown in the left column of Fig. 6), Fig. 5(b) represents the dense topology (results shown in the right column of Fig. 6), and Fig. 7 represents the backbone ARPANET topology (results shown in Fig. 9).

We study two representative AoI-weight values: $\delta = 1$, which assigns the AoI penalty the same weight as the forwarding cost, and $\delta = 2$, which emphasizes freshness more strongly.

For Figs. 6 and 8, we draw link reliabilities and node costs from normal distributions: $p_{i,j} \sim N(p_{\text{mean}}, \sigma^2)$ with $\sigma = 0.2$ for all $(i, j) \in E$, and $c_i \sim N(c_{\text{mean}}, \sigma^2)$ with $\sigma = 1$ and $c_{\text{mean}} = 5$ for all $i \in V$. For the ARPANET topology, we assign link costs and reliabilities in proportion to link lengths. We compare our optimal algorithm against three baselines: (i) least total forwarding cost, (ii) highest end-to-end reliability (maximum product of link reliabilities), and (iii) least total link transmission time.

5.2. Experimental results

We observe that the delivered utility at the destination node d is higher in denser graphs than in their sparse counterparts. This is because denser graphs are more strongly connected and provide more candidate routes, increasing the likelihood of a path with smaller expected utility attrition.

5.2.1. $\delta = 1$

Now, we discuss the results shown in Fig. 6 where δ equals the unity. First, the first row shows the behavior of the total utility of the message when it arrives at the destination node d as the reliability values mean of the links changes. Although the cost values mean that the nodes would incur is constant, the distribution of the cost values is done probabilistically in a way that follows the normal distribution. The first row shows this change of behavior under different values of the beneficial reward of the message b. We can see that as the links become more reliable, the utility increases in a way that is nearly linearly proportional to the mean probability.

Second, the second row of Fig. 6 shows the behavior of the total utility of the message when it arrives at the destination node d as the beneficial reward that the message starts with for different forwarding cost values mean. Although the cost values mean that the links would incur is constant for each graph, the distribution of the cost values is done probabilistically in a way that follows the normal distribution. The same applies to the reliability values of the links. The second row shows this change of behavior under different values of the cost values mean c_{mean}. We can see that as the message carries a more beneficial reward b, the utility increases in a way that is stronger than the linear proportion.

5.2.2. $\delta = 2$

We now study the results in Fig. 8. In the first row of the figure, we may see the behavior of the utility of the message when it arrives at the destination node d as the mean reliability value of the links p_{mean} changes. Despite the fact that the forwarding cost values mean does not change, the distribution of the cost values is done in a probabilistic manner in a way that follows the normal distribution. The first row shows this change of behavior under different values of the beneficial reward of the message b. We can see that as the links become more reliable, the utility increases in a way that is nearly linearly proportional to the mean probability.

For the contrast between the settings where the AoI has different weights in the utility, we may observe that for a varying mean of reliability values, the total utility becomes more sensitive to the change of the initial beneficial reward b; a small change of the value of b would make the utility of the message changes more drastically when the value of δ is higher, i.e. when the AoI weighs more.

On the other hand, in the case when we have varying cost mean, we may observe that the utility becomes more sensitive to the change in the beneficial reward when the value of δ is lower, i.e. when the AoI has less weight with respect to the forwarding cost in the utility function.

Lastly, the plots in Fig. 10 consider the case in which the messages are not discarded when the expected utility value becomes negative. We observe that the effect of discarding the messages makes the performance of the optimal algorithm deteriorates the least, while we see that the performances of the other three algorithms deteriorate more drastically.

5.3. Simulation summary

We conducted simulations across multiple parameter settings and concluded that messages generally arrive at the destination node d with higher utility in denser graphs than in their relatively sparse counterparts. This is because dense graphs are more strongly connected and offer more routing options that can reduce expected utility attrition under the same settings.

We also observed an approximately linear relationship between the mean forwarding cost and the mean link reliability in their effects on delivered utility. In contrast, the relationship between the initial benefit b and delivered utility is more strongly nonlinear.

Moreover, the sensitivity of delivered utility to changes in b depends on the AoI weight δ and on which parameter is varied. When varying the mean cost, the sensitivity is higher for smaller δ; when varying the mean link reliability, the sensitivity is higher for larger δ.

Finally, our simulations show that enabling message discarding makes the proposed algorithm degrade the least among all compared methods, while the baselines degrade more sharply. Across the representative sparse/dense pair, the dense graph yields about 55% higher delivered utility than the sparse graph under the same parameterization.

6. Conclusion

In this work, we introduced an Age-of-Information-sensitive utility model for unreliable multi-hop networks, in which each stochastically generated update has a time-sensitive utility that decays with AoI. The model captures an explicit trade-off between forwarding cost and delay-induced freshness loss under an AoI-grounded utility. Under this model, we proposed an optimal routing algorithm that maximizes the expected utility of delivered updates. The algorithm leverages an optimal path that minimizes expected utility attrition while accounting for link unreliability, where each failed attempt consumes a random amount of time governed by an exponential failure-time model. By forwarding updates along this optimal path when the intrinsic benefit b covers the expected remaining utility attrition, and dropping updates otherwise, the algorithm achieves optimality in expectation. We also discussed a fully reliable special case, which reduces to a standard shortest-path formulation. Finally, we conducted simulations on three representative topologies to evaluate performance. Across a representative sparse/dense pair, the dense graph achieves about 55% higher delivered utility than the sparse graph under the same settings, and the proposed algorithm substantially outperforms the baselines both with and without discarding.

CRediT authorship contribution statement

Abdalaziz Sawwan: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Mingjun Xiao:** Writing – review & editing, Validation; **Jie Wu:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Data availability

Instructions of synthesizing the data are given in the paper

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported in part by NSF grants CNS 2214940, CPS 2128378, CNS 2107014, and CNS 2150152.

Appendix A. Proof of Lemma 1

In order to evaluate the expected total reduction of the utility if a link is used over and over again until it succeeds, we need first to evaluate the expected total reduction for each failure, and since the attempts to use the link are independent, we can apply the standard Bernoulli trial results to evaluate the total expected reduction of utility until success. Now, we consider the single case in which failure occurs. From the characterization in Eq. (8), we end up with the following probability density function (PDF):

$$f(t) = (1/\lambda_{i,j})e^{-t/\lambda_{i,j}} \tag{A.1}$$

on its range. Furthermore, the utility reduction given that a failure happened where that failure took time t after the transmission was attempted at time t_0 and node i is given by:

$$u(t_0) - u(t_0 + t) = c_i + \delta t, \quad 0 \le t < \tau_{i,j} \tag{A.2}$$

Now, we have everything set up to calculate the expected reduction of utility after using the link between node i and j, given that the delivery will fail. We do that by evaluating the integral of the multiplication of the functions in Eqs. (A.1) and (A.2) after normalizing the PDF shown in Eq. (A.1). This means we will normalize the PDF by multiplying it with a factor of $\frac{1}{1-p_{i,j}}$. Hence, the expected reduction of utility given that the delivery will fail will be given by:

$$\begin{aligned}&\int_0^{\tau_{i,j}} (c_i + \delta t) \times (1/(1-p_{i,j})\lambda_{i,j}) \times e^{-t/\lambda_{i,j}}\, dt \\ &= (p_{i,j}\log p_{i,j} + 1 - p_{i,j})\lambda_{i,j}\delta/(1-p_{i,j}) + c_i = R^f_{i,j}\end{aligned} \tag{A.3}$$

where $R^f_{i,j}$ represents the expected reduction of utility given that the delivery will fail. We know that the utility reduction after using the link between node i and j given that the delivery will succeed is given by:

$$\delta\tau_{i,j} + c_i = R^s_{i,j} \tag{A.4}$$

where $R^s_{i,j}$ represents the reduction of utility given that the delivery will succeed.

Now, we need to measure the expected number of times we try to use a link until we succeed. This includes the times that result in failure while using the link in addition to the first time that result in success. Since the probability that it will take exactly one trial until success is $p_{i,j}$, the probability that it will take exactly two trials until success is $(1-p_{i,j})p_{i,j}$, and the probability that it will take n trials until success is $(1-p_{i,j})^{n-1}p_{i,j}$, then we will get the following series for the expected total number of times until success:

$$p_{i,j} + 2(1-p_{i,j})p_{i,j} + \cdots = \Sigma_{n=0}^{\infty}(n+1)(1-p_{i,j})^n p_{i,j} \tag{A.5}$$

To evaluate this series, we denote its value by S.

$$S = p_{i,j} + 2(1-p_{i,j})p_{i,j} + \cdots = \Sigma_{n=0}^{\infty}(n+1)(1-p_{i,j})^n p_{i,j} \tag{A.6}$$

In order to evaluate S, we utilize the generating functions method [42] as follows:

$$\begin{aligned}g(x) &= p_{i,j} + 2(1-p_{i,j})p_{i,j}x + \ldots \\ &= \Sigma_{n=0}^{\infty}(n+1)(1-p_{i,j})^n p_{i,j}x^n\end{aligned} \tag{A.7}$$

Now, we know that, by definition, $S = g(1)$. So we need to find the closed form of $g(x)$. In order to do that, we integrate the function, get the closed form of the resulting series, then differentiate this closed form of the integration in order to get $g(x)$ back. We proceed as follows:

$$\begin{aligned}G(x) &= \int g(x)dx = \int \sum_{n=0}^{\infty}(n+1)(1-p_{i,j})^n p_{i,j}x^n dx \\ &= (p_{i,j}/(1-p_{i,j}))\sum_{n=0}^{\infty}(1-p_{i,j})^{n+1}x^{n+1} + C\end{aligned} \tag{A.8}$$

But, we know that

$$\sum_{n=0}^{\infty} r^{n+1} = \sum_{n=1}^{\infty} r^n = r/(1-r), \quad |r| < 1 \tag{A.9}$$

Hence, from Eq. (A.8), $G(x)$ reduces to the following:

$$G(x) = (p_{i,j}x/(1-(1-p_{i,j})x)) + C \tag{A.10}$$

Now, we differentiate $G(x)$ to get $g(x)$ as following:

$$\frac{\partial G(x)}{\partial x} = g(x) = p_{i,j}/(1-x+p_{i,j}x)^2 \tag{A.11}$$

which yields directly to $S = g(1) = 1/p_{i,j}$. This means that the expected number of trials needed to have the first success, given that the events are independent with success probability of $p_{i,j}$, is equal to $\frac{1}{p_{i,j}}$; one trial is the last successful trial, and $\frac{1}{p_{i,j}} - 1$ trials are expected to result in failure.

Now, we have everything set up to evaluate the total expected reduction of the utility after using a link continually until its first success. We denote this expected reduction by $\mathbb{E}[R_{i,j}]$ and show it in the following equation:

$$\begin{aligned}\mathbb{E}[R_{i,j}] &= 1 \times R^s_{i,j} + (1/p_{i,j} - 1)R^f_{i,j} \\ &= (1-p_{i,j})\lambda_{i,j}\delta/p_{i,j} + c_i/p_{i,j} \\ &= (1-p_{i,j})(-\tau_{i,j})\delta/(p_{i,j}\log p_{i,j}) + c_i/p_{i,j}\end{aligned} \tag{A.12}$$

By that, we have utilized the independence between the events that would occur at the links to reduce the intricate probabilistic model into the one that is concerned with the expected value of the reduction of the message's total utility.

General failure-time distributions. If the failure-detection time on (i,j) is not exponential but follows an arbitrary empirical distribution supported on $[0, \tau_{i,j})$, the same decomposition into a final successful attempt plus a geometric number of failed attempts still holds. Let $\mu^f_{i,j} = \mathbb{E}[F_{i,j} \mid \text{failure}]$. Then the expected link-level utility attrition under retry-until-success becomes $\mathbb{E}[R_{i,j}] = \frac{c_i}{p_{i,j}} + \delta\left(\tau_{i,j} + \frac{1-p_{i,j}}{p_{i,j}}\mu^f_{i,j}\right)$. The truncated exponential model in the main text is chosen because it yields a closed-form $\mu^f_{i,j}$ in terms of $(\tau_{i,j}, p_{i,j})$, which simplifies to (9).

References

[1] R.D. Yates, Y. Sun, D.R. Brown, S.K. Kaul, E. Modiano, S. Ulukus, Age of information: an introduction and survey, 2021.
[2] S. Kaul, R. Yates, M. Gruteser, Real-time status: how often should one update, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 2731–2735.
[3] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, E. Modiano, Scheduling policies for minimizing age of information in broadcast wireless networks, IEEE/ACM Trans. Netw. 26 (6) (2018) 2637–2650.
[4] J. Lou, X. Yuan, S. Kompella, N.F. Tzeng, AoI and throughput tradeoffs in routing-aware multi-hop wireless networks, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 476–485.
[5] C.K. Toh, A novel distributed routing protocol to support ad-hoc mobile computing, in: Conference Proceedings of the IEEE International Phoenix Conference on Computers and Communications, 1996, pp. 480–486.
[6] Y. Xue, Y. Cui, K. Nahrstedt, A utility-based distributed maximum lifetime routing algorithm for wireless networks, in: Qshine, 2005, p. 10.
[7] Z. Ye, S.V. Krishnamurthy, S.K. Tripathi, A framework for reliable routing in mobile ad hoc networks, IEEE Infocom 2003 1 (2003) 270–280.

[8] D.S.D. Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proceedings of the International Conference on Mobile Computing and Networking, the international conference on Mobile computing and networking, 2003, pp. 134–146.
[9] M. Jasim, A new definition of expected transmission count as an ad-hoc network routing information, 2017. Doctoral dissertation.
[10] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: Proceedings of the Conference on Embedded Networked Sensor Systems, the conference on Embedded networked sensor systems, 2003, pp. 14–27.
[11] M.A. Rassam, A. Zainal, M.A. Maarof, M. Al-Shaboti, A sinkhole attack detection scheme in mintroute wireless sensor networks, in: 2012 International Symposium on Telecommunication Technologies, IEEE, 2012, pp. 71–75.
[12] J. Wu, M. Lu, F. Li, Utility-based opportunistic routing in multi-hop wireless networks, in: The International Conference on Distributed Computing Systems, 2008, pp. 470–477.
[13] M. Lu, F. Li, J. Wu, Efficient opportunistic routing in utility-based ad hoc networks,IEEE Transactions on Reliability 2009, pp. 485–495.
[14] Z. Li, H. Shen, Utility-based distributed routing in intermittently connected networks, in: International Conference on Parallel Processing, 2008, pp. 604–611.
[15] Z. Li, H. Shen, SEDUM: Exploiting social networks in utility-based distributed routing for, DTNs IEEE Trans. Comput. 62 (1) (2011) 83–97.
[16] N. Niknami, A. Sawwan, J. Wu, A failover mechanism for sfc requests in next-generation networks,IEEE Trans. Reliab. 2024.
[17] M. Xiao, J. Wu, C. Liu, L. Huang, Tour: time-sensitive opportunistic utility-based routing in delay tolerant networks, in: Proceedings IEEE INFOCOM, IEEE INFOCOM, 2013, pp. 2085–2091.
[18] M. Xiao, J. Wu, L. Huang, Time-sensitive utility-based single-copy routing in low-duty-cycle wireless sensor networks, in: IEEE Transactions on Parallel and Distributed Systems, 2014.
[19] M. Xiao, J. Wu, L. Huang, Time-sensitive utility-based routing in duty-cycle wireless sensor networks with unreliable links, in: IEEE S, 2012, pp. 311–320.
[20] M. Xiao, J. Wu, H. Huang, L. Huang, W. Yang, Deadline-sensitive opportunistic utility-based routing in cyclic mobile social networks, in: IEEE SECON, 2015, pp. 301–309.
[21] T. Kimura, T. Matsuura, M. Sasabe, T. Matsuda, T. Takine, Location-aware utility-based routing for store-carry-forward message delivery, in: ICOIN, 2015, pp. 194–199.
[22] C. Zhou, D. Qian, H. Lee, Utility-based routing in wireless ad hoc networks, in: IEEE MASS, 2004, pp. 588–593.
[23] A. Kosta, N. Pappas, V. Angelakis, Age of information: a new concept, metric, and tool, Found. Trends Netw. 12 (3) (2017) 162–259.
[24] R.D. Yates, S.K. Kaul, The age of information: real-time status updating by multiple sources, IEEE Trans. Inf. Theory 65 (3) (2018) 1807–1827.
[25] H. Wang, Q. Sun, S. Wang, A survey on the optimisation of age of information in wireless networks, Int. J. Web Grid Serv. 19 (1) (2023) 1–33.
[26] X. Chen, K. Gatsis, H. Hassani, S.S. Bidokhti, Age of information in random access channels, IEEE Trans. Inf. Theory, 2022, pp. 6548–6568.
[27] Y. Liu, L.X. Cai, Q. Chen, H. Zhang, F. Hou, T.H. Luan, Minimizing age of information in non-orthogonal random access networks, IEEE Internet Things J., 2024, pp. 24886–24902.
[28] H. Tang, J. Wang, L. Song, J. Song, Minimizing age of information with power constraints: multi-user opportunistic scheduling in multi-state time-varying channels, IEEE J. Sel. Areas Commun. 38 (5) (2020) 854–868.
[29] L. Wang, R. Fan, H. Hu, G. Wang, J. Cheng, Age of information minimization for opportunistic channel access, IEEE Trans. Commun. 2024, pp. 7449–7465.
[30] N. Akar, O. Dogan, Discrete-time queueing model of age of information with multiple information sources, IEEE Internet Things J. 8 (19) (2021) 14531–14542.
[31] J.P. Champati, R.R. Avula, T.J. Oechtering, J. Gross, Minimum achievable peak age of information under service preemptions and request delay, IEEE J. Sel. Areas Commun. 39 (5) (2021) 1365–1379.
[32] M.A. Abd-Elmagid, H.S. Dhillon, Age of information in multi-source updating systems powered by energy harvesting,IEEE J. Sel. Areas Inf. Theory 2022, pp. 98–112.
[33] A. Ndikumana, K.K. Nguyen, M. Cheriet, Age of processing-aware offloading decision for autonomous vehicles in 5G open RAN environment,IEEE Trans. Mob. Comput.. 2023, pp. 7865–7877.
[34] A. Köse, M. Koca, E. Anarim, Impact of network coding on age of information, IEEE Internet Things J., 2023, pp. 9725–9739.
[35] F. Isayama, T. Shigeyasu, A new DTN routing strategy considering age of information, in: International Conference on Broadband and Wireless Computing, Communication and Applications, Cham; Nature Switzerland, Springer, 2023, pp. 263–272.
[36] A. Sawwan, J. Wu, Unreliable multi-hop networks routing protocol for age of information-sensitive communication, in: IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops, 2022, pp. 1–2.
[37] A. Sawwan, J. Wu, On edge pruning of communication networks under an age-of-Information framework, Algorithms 15 (7) (2022) 228.
[38] J.D. Kalbfleisch, R.L. Prentice, The Statistical Analysis of Failure Time Data, John Wiley & Sons, 2011.
[39] T.K. Sarkar, An exact lower confidence bound for the reliability of a series system where each component has an exponential time to failure distribution, Technical Report, Technometrics, 1971.
[40] A. Goldberg, T. Radzik, A heuristic improvement of the Bellman-Ford algorithm, Technical Report, Stanford University CA Department of Computer Science, 1993.
[41] S. Hougardy, The floyd-Warshall algorithm on graphs with negative cycles, Inf. Process. Lett. 110 (8–9) (2010) 279–281.
[42] G. Levitin, The Universal Generating Function in Reliability Analysis and Optimization, 6, Springer, London, London, 2005.