A Real-time Aware Routing Strategy in Smart City Environments

YuSheng Xia*, JinShu Su† , RongMao Chen*, YaPing Liu‡, Shuhui Chen*
*College of Computer, National University of Defense Technology, Changsha, China
†National Key Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China,
‡Information Security and Big Data Research Institute, Central South University, Changsha, China,
Email:{xys,sjs,chromao,shchen}@nudt.edu.cn, yp.liu@csu.edu.cn

***Abstract*—In future smart cities, the proliferation and mobility of smart devices will bring many challenges to efficient communication for the city-wide networking, including frequent link fluctuation and disconnection. GSTAR (Generalized storage-aware routing) has been a robust routing protocol that utilizes in-network storage to address these challenges. Unfortunately, the cache-and-forward mechanism adopted by GSTAR usually results in extra delay for storing real-time data, and thus makes it not applicable for some particular real-time applications, which crucially require low latency. Motivated by this problem, we propose a real-time aware strategy to reduce the total delay of time-sensitive data for real-time applications. Also, we specifically analyze the threshold to store and forward via mathematical modeling and access the rationality of associated parameters. Through NS3-based simulation, we show that our designed real-time aware strategy can decrease the average delay of time-sensitive data and prevent the block of time-insensitive data compared with the other strategies.**

***Index Terms*—GSTAR; Wireless; Real-Time; Modeling;**

I. INTRODUCTION

In the near future, our cities will be highly intelligentized due to the proliferation of an incredibly higher number of smart devices. These devices generate, collect, exchange and process big data, provide distributed services, offer computational resources and cooperate to perform some tasks. The common mobility of these devices(e.g. smartphones) easily results in link fluctuation and disconnection, which demands for rubust routing protocol to efficiently guarantee communication. In this circumstance, GSTAR (Generalized Storage Aware Routing) [1] is an optional routing protocol originating from the *MobilityFirst* project [2], funded by NSF FIA program.

GSTAR is designed to deal with the challenges of the wireless domain including mobility, varying levels of disconnection, multi-homing and so on, by unifying techniques from MANET [3] and DTN protocols [4]. It can be applied in the access networks with both wired and wireless elements, and works as an intra-domain routing protocol when the network is mostly well-connected. The primary mechanism adopted by GSTAR is the utilization of in-network storage that enables routers to deal with network fluctuations. In particular, GSTAR uses the in-network storage [5] to store data when the network problems (e.g., link quality fluctuation, congestion, and disconnection) occur, and after that forwards the cached data once the link quality gets better.

Despite the fact that GSTAR could utilize storage to handle the link fluctuation, the cache-and-forward mechanism adopted by GSTAR usually results in extra delay for storing real-time data, and thus makes it not applicable for some particular real-time applications, which crucially require low latency. For example, for on-line games and applications supporting VoIP (Voice over IP), Storing the real-time data at the router would append additional storing time and result in unnecessary delay, which jeopardizes the user experience.

In addition, the network fluctuation in practice is usually of various forms and frequencies. It is therefore necessary to dynamically adjust the threshold of the underlying cache-and-forward mechanism [6]. However, GSTAR lacks in flexibility to accommodate the dynamic network environment. The plan of making GSTAR more robust to dynamic network environment by automatically varying its threshold has been unsolved since the proposal of GSTAR.

In this work, we investigate the study of addressing the aforementioned limitations of GSTAR when used for real-time applications. Precisely, we present a real-time aware strategy where the threshold is variable for time-sensitive and time-insensitive packets respectively. Furthermore, we establish a mathematical model to find the optimized range of the variation and analyze the potential problem caused by such a variation.

The main contributions of our work are three folds:

- We present the real-time aware strategy, which intelligently forwarding data according to the real-time requirements of them by varying the threshold to forward and store. To the best of our knowledge, we are the first to consider real-time requirements of GSTAR.
- we establish a mathematical model to calculate the optimized range of the varying threshold and analyze the impact of the varying threshold on the delay. Also, we utilize Matlab to help assess the rationality of the strategy.
- we implement GSTAR and the real-time aware strategy with NS-3 simulator and present comparative results of various strategies. The results show that our strategy can decrease the average delay of time-sensitive packets.

978-1-5386-5241-1/18/$31.00 ©2018 IEEE

The remainder of this paper is as follows. Section II introduces the details and limitations of GSTAR and present a simple solution. Section III gives the details of the real-time aware strategy and utilizes a model to assess the impact of the varying threshold on the delay of stored data. Section IV presents an NS3-based evaluation to compare the receiving time and average delay of various strategies. Section V makes some necessary discussion. Finally, Section VI concludes the paper.

II. Principles and limitations of GSTAR

GSTAR utilizes in-network storage to handle network fluctuations and controls them in a limited part of the whole network. This section presents a brief overview of GSTAR. Following this, we put forward some limitations and a simple solution.

A. Principles of GSTAR

In generalized storage aware routing protocol, each node in the network sustains two types of graphs [6] for routing, containing intra-partition graph and DTN graph. Intra-partition graph is only used for routing in the partition while DTN graph allows a node to gain general connectivity information of the nodes outside of its current partition, and therefore it can be used for routing beyond the current partition.

In GSTAR, routing consists of path selecting and forwarding. During selecting the path, the node utilizes intra-partition graph and DTN graph to select the next-hops by using the shortest path algorithm.

During forwarding, if an intra-partition path exists to the destination, the node compares the short-term path quality ($STPath$) and the long-term path quality ($LTPath$) to determine whether forwarding the packets. $STPath$ and $LTPath$ are the sum of short-term and long-term Expected Transmission Time (ETT) along the path. If $STPath > k * LTPath$, the packets are stored, else, a forward decision is made. k is a parameter that influences the threshold.

B. Limitations of GSTAR

The limitations of GSTAR are two folds, due to requirements for the lower latency of real-time applications and requirements for dynamic k.

1) Requirements for low latency: The emergence and popularization of real-time applications improve the requirements for low latency for better user experience. However, according to the storage-based mechanism of GSTAR, Storing the real-time data at the router would append additional storing time and result in an unnecessary delay.

2) Requirement for dynamic k: It is essential to route with the appropriate value of k. However, the problem of finding appropriate k in GSTAR has been unresolved. Precisely, the network fluctuation in practice is usually of various forms and frequencies. It is therefore necessary to dynamically adjust the k of the underlying cache-and-forward mechanism. Besides, GSTAR lacks in flexibility to accommodate the dynamic network environment. Thus, it is necessary to utilize dynamic k to face the challenges.

We need to point out that this paper do not attempt to find the optimized k, but utilizes the requirements for dynamic k to help to realize real-time aware routing.

C. Sequential forwarding strategy

Sequential forwarding strategy [7] is the easiest method dealing with real-time issues. During forwarding period, the time-sensitive packets have higher priority. After sending all the time-sensitive packets, the node begins to handle the other packets.

However, this strategy has the following problems:

- If time-sensitive packets continuously arrive, the time-insensitive packets may be choked at the router.
- This strategy is unable to contribute to load balance because all packets are forwarded to the same path once the threshold is satisfied.

III. Real-time aware strategy

The real-time aware strategy aims to decrease the average delay of real-time packets, avoid the choked packets and contribute to the load balance. This section gives the details of the real-time aware strategy and establishes a mathematical model to analyze the dynamic k.

A. Real-time aware strategy

Because the threshold is essential for routing and can vary according to specific situations, the real-time aware strategy works by adjusting the threshold when caching and forwarding.

For time-sensitive packets, to decrease the transmission delay, we can forward them when the short-term path quality is slightly worse. We can achieve this goal by increasing k in $STPath > k * LTPath$, because $STPath$ can be worse for forwarding. Thus we set $(k + \delta) * LTPath$ as the threshold for time-sensitive packets where δ is a little variable that can be configured by the network administrator. The rationality of δ will be discussed in the following section.

For time-insensitive packets, it is acceptable to increase the transmission delay, so we can store them when the short-term path quality is slightly better. We can achieve this goal by decreasing k because $STPath$ needs to be better for forwarding. Thus we set $(k - \delta) * LTPath$ as the threshold for time-insensitive packets.

Based on the above discussion, Table I shows the decisions to forward and store for the real-time aware strategy.

TABLE I
Decisions to forward and store

	time-sensitive	**time-insensitive**
$STPath < (k - \delta) * LTPath$	forward	forward
$(k - \delta) * LTPath < STPath < (k + \delta) * LTPath$	forward	store
$(k + \delta) * LTPath < STPath$	store	store

B. The value of δ

In this section, we focus on two questions: First, *what's the optimized value of δ?* Second, *is δ rational?* This section aims to establish a mathematical model to analyze the value and the rationality of δ.

1) The establishment of the model: In this model, we suppose that dynamic k is influenced by the fluctuation and arrival rate of packets at the router.

We limit the scope of modeling to periodic functions. As shown in Figure 1, the link fluctuates in periodic form of $R = f(T)$. Normal traffic does not distinguish time sensitive and insensitive traffic. R represents the link transmission rate. t represents the retransmission delay during which the cached packets are forwarded completely. t_2 represents the time when the stored packets start retransmission.

$k' * R_{long}$ is the forwarding threshold in rate, compared with $k * LTPath$. We use Th to represent the forwarding threshold, including $k' * R_{long}$ and $k * LTPath$.

2) The relationship between transmission rate and delay: It is necessary to find the connection between the transmission rate and delay because, in our model, we utilize $k' * R_{long}$ to replace the threshold($k * LTPath$) of GSTAR.

Generally, the total delay of the packet includes the nodal processing delay(d_{proc}), the queuing delay(d_{queue}), the transmission delay(d_{trans}) and the propagation delay(d_{prop}) [8]:

$$d_{delay} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \tag{1}$$

The relationship between d_{trans} and transmission rate R is that

$$d_{trans} = \frac{L}{R} \tag{2}$$

Where L is the length of the packet. In our work, the relationship between $STPath(k * LTPath\ )$ and $R_{short}(k' * R_{long})$ is that:

$$k * LTPath - d_{static} = \frac{L_{probe}}{k' * R_{long}} \tag{3}$$

Where $d_{static} = d_{proc} + d_{queue} + d_{prop}$.

TABLE II
NOTATION

notation	meaning
R	the link transmission rate
V	the arrival rate
t	the transmission delay
t_2	the time stored packets start retransmission
R_{long}, R_{short}	the long-term and short-term transmission rate
t_4	the time stored real-time packets start retransmission
t_{rtt}	the transmission delay of real-time traffic
V_{rtt}	the arrival rate of real-time packets
Th_{rtt}	the forwarding threshold of real-time traffic
δ_{rtt}	δ for real-time packets
t_{nrtt}	time insensitive packets start retransmission
Th_v	the threshold of the arrival rate

3) The value of δ: We provide a dynamic solution to determine the value of δ and in which we follow three principles:

1) real-time traffic should not be forwarded when the link quality is far worse.
2) real-time traffic tends to be forwarded preferentially and individually.
3) real-time traffic should be forwarded at higher link quality to decrease retransmission and packet loss.

The priority of these three principles is from high to low.

To analyse δ's influence on the real-time traffic, we make some preparations at first, mainly about the relationship between variations. specifically, from the routing logic of the established model, the total size of the packets forwarded during retransmission(t) is equal to that of the arriving packets during the storing $(t_2 - t_1)$ and retransmission (t). Therefore, we can get that:

$$\int_{t_2}^{t_2+t} f(t)\Delta t = (t_2 - t_1 + t)V \tag{4}$$

Similarly, for real-time traffic shown in Figure 2:

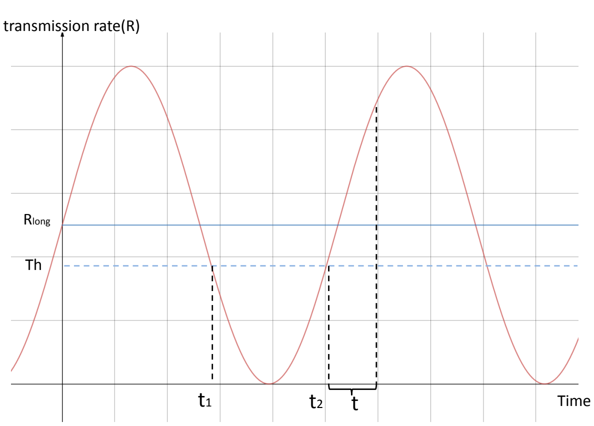

Fig. 1. Caching and forwarding model for normal traffic.

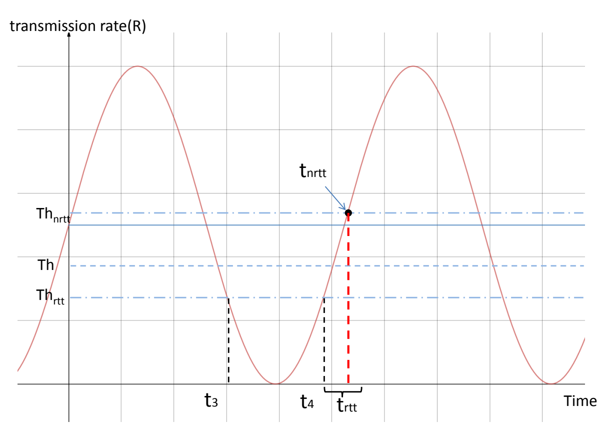

Fig. 2. Caching and forwarding model for real-time traffic.

$$\int_{t_4}^{t_4+t_{rtt}} f(t)\Delta t = (t_4 - t_3 + t_{rtt})V \quad (5)$$

By combining Equation 3, the relationship between t_1, t_2, k' and k for normal traffic is that:

$$\begin{aligned} f(t_1) = f(t_2) &= k' \cdot R_{long} \\ &= \frac{R_{long}}{k * LTPath - d_{static}} \end{aligned} \quad (6)$$

Then, based on previous principles, we put forward the scheme to obtain the value of δ as follows (the router can obtain the link fluctuation curve by $STPath$):

Firstly, the router calculates Th_{rtt}^{min} and Th_v. According to *principle 1*, the router sets minimum forwarding threshold for real-time traffic to be Th_{rtt}^{min}(corresponding to δ_{max}=0.1). Also, the router calculates the threshold of V(i.e. Th_v) by setting $t_4 + t_{rtt} = t_{nrtt}$ in Equation 5, which reflects that real-time traffic has been completely forwarded before time insensitive traffic starts to be forwarded.

Secondly, the router sets $\delta = \delta_{rtt}$ when $V_{rtt} < Th_v$. The router calculates δ_{rtt} with Equation 5 and 6 in which $t_4 + t_{rtt} = t_{nrtt}$ and $V = V_{rtt}$. According to *principle 2*, $\delta \in [\delta_{rtt}, \delta_{max}]$, because real-time traffic tends to be completely forwarded alone and preferentially . According to *principle 3*, $\delta = \delta_{rtt}$, because δ ought to be smaller.

Thirdly, the router sets $\delta = \delta_{max}$ when $V_{rtt} < Th_v$. In this case, cached real-time traffic can not be completely forwarded before t_{nrtt}. According to *principle 2*, $\delta = \delta_{max}$, because the router tries forwarding more real-time traffic individually.

In conclusion, when the arrival rate of real-time traffic is less than Th_v, $\delta = \delta_{rtt}$. Otherwise, $\delta = \delta_{max}$.

4) the rationality of δ: Now we discuss the rationality of δ. Specifically, δ may cause the potential problem that the variation of k may seriously prolongs the retransmission delay of the stored data, due to the worse link quality which has not yet fully recovered.

As we all known, *Fourier Series* decomposes any periodic function or periodic signal into the sum of a set of simple oscillating functions, namely sines and cosines. In this circumstance, we only need to consider the commonest situation that the link fluctuates in the form of a sine function. Thus in this section, the link fluctuates in the form of $R = a \sin bt + c$.

In order to assess the impact of the variation of k(represented by t_2) on the retransmission delay t, we choose to assess $\frac{\partial t}{\partial t_2}$, which is the sensitivity of the change of t with respect to the change in t_2 with other variables treated as constant. We can get $\frac{\partial t}{\partial t_2}$ from Equation 4:

$$\frac{\partial t}{\partial t_2} = \frac{a \sin bt_2 - a \sin(bt_2 + bt) + 2V}{a \sin(bt_2 + bt) - V + c} \quad (7)$$

The further analysis of $\frac{\partial t}{\partial t_2}$ for the implicit function is conducted on the Matlab. We utilize Matlab to draw the implicit function to study the status of $\frac{\partial t}{\partial t_2}$. We set $R_{long} = 5Mbps$, $k_{min} = 0.8$, $k_{max} = 2$. For the rationality of δ, we analyze the maximum of $\frac{\partial t}{\partial t_2}$.

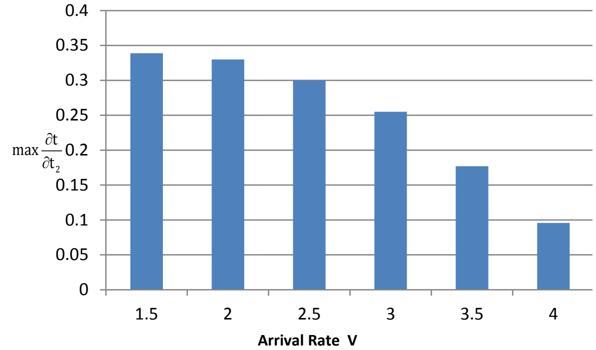

Fig. 3. Maximum $\frac{\partial t}{\partial t_2}$ for various arrival rate

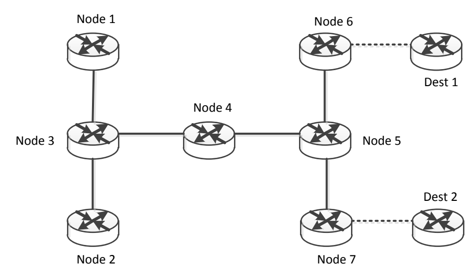

Fig. 4. Network topology

Figure 3 shows the maximum value of $\frac{\partial t}{\partial t_2}$ for various arrival rate V. As can be seen from the figure, when V decreases, $max\frac{\partial t}{\partial t_2}$ gets larger while the growth slows down and approaches 0. In addition, when $\Delta k(\delta)$ approaches δ_{max} of 0.1, Δt_2 is less than 0.1, according to Equation 6. Thus, Δt is less than $max\frac{\partial t}{\partial t_2} * \Delta t_2 = 0.035$, which is less than 1% of the retransmission delay. Therefore, the tiny variation of k (δ) will not have great impact on the retransmission delay t.

In conclusion, when the link fluctuates in a periodic function: δ which is the little variation of k, is rational because it hardly has a huge impact on the retransmission delay of the stored packets.

IV. Performance Evaluation

The goal of our evaluation is to verify the rationality of the real-time aware strategy. This section evaluates receiving time of various packets and average delay of time-sensitive packets.

The experiment is conducted with NS-3 simulator [9]. We implement GSTAR with the simulator and enhance it with the real-time aware strategy.

Figure 4 shows the network topology [10]. The links connecting the destination nodes (Dest 6 and 7) are wireless, and the other links are wired. The bit rate of the wired link is $6Mbps$, and the delay is $2ms$. The parameter δ is set 0.1, and k is set 1.11. The experiment has two flows:

- Flow 1 : Node1 $\rightarrow$ Dest1 of 5 hops, this flow sends time-insensitive packets
- Flow 2 : Node2 $\rightarrow$ Dest2 of 5 hops, this flow sends time-sensitive packets

Link fluctuation:

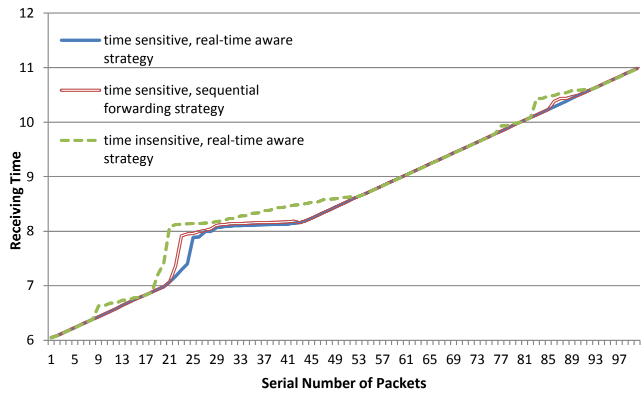

Fig. 5. Receiving time of various strategies

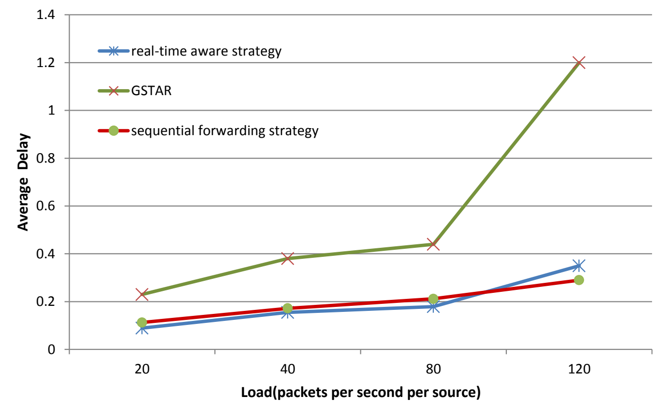

Fig. 6. Average delay of time-sensitive packets

- Fluctuation 1 : Node6 $\rightarrow$ Dest1, the link fluctuates every three seconds from 7s

A. Receiving Time

Two flows in the experiment both send 100 packets in 5s. The results are shown in Figure 5.

Figure 5 illustrates that, for the real-time aware strategy, time-sensitive packets are less susceptible to link fluctuation than time-insensitive packets. Though the time-insensitive packets' delay increases, it is in acceptable range. In addition, for time-sensitive packets, the real-time aware strategy stores packets later than the sequential forwarding strategy. As the link quality increases rapidly, the time-sensitive packets are forwarded preferentially.

B. Average delay of time-sensitive packets

The average delay is the time that each packet spends in the network. The average delay $\overline{T}$ is defined as the average of the forwarding time for all packets:

$$\overline{T} = \frac{\sum_{i=1}^{N} t_{recv} - \sum_{i=1}^{N} t_{send}}{N} \tag{8}$$

Where $\sum_{i=1}^{N} t_{recv}$ is the sum of the reception time, $\sum_{i=1}^{N} t_{send}$ is the sum of the sending time of all packets respectively, and N is the total number of packets.

In this experiment, the flows respectively send packets at $20 packets/s$, $40 packets/s$, $80 packets/s$, $120 packets/s$ for 5 seconds. We use various strategies to calculate the average delay of the time-sensitive packets.

Figure 6 shows the experimental results. It reveals that compared with GSTAR, the real-time aware strategy apparently reduces the average delay of time-sensitive packets. Also, compared with the sequential forwarding strategy, with the smaller load, the average delay of the real-time aware strategy is slightly lower because the router postpones the store of packets. However, with the larger load, the real-time aware strategy needs to forward more time-sensitive packets together with time-insensitive packets when $STPath < (k - \delta) * LTPath$. As a result, the average delay is larger than that of the sequential forwarding strategy. However, in this case, the sequential forwarding strategy may have the problem that it possibly blocks the time-insensitive packets.

V. Conclusions

In this paper, we have proposed and evaluated the real-time aware strategy to enhance GSTAR to satisfy the demand of real-time applications. We specifically analyze the value and impact of the variation of the threshold by mathematical modeling. By NS-3 based simulation, we manifest that the real-time aware strategy can decrease the average delay of time-sensitive data compared with GSTAR and the sequential forwarding strategy. In future work, we plan to validate the conclusion by applying the method to more unpredictable fluctuation and try to develop the method to dispose of more real-time grades.

Acknowledgment

This research is supported by the program for National Basic Research Program of China (973 Program 2012CB315906) and the National Science Technology Fund of China (NSFC-61472438).

References

[1] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, "Gstar: generalized storage-aware routing for mobilityfirst in the future mobile internet," in *International Workshop on Mobiarch*, 2011, pp. 19–24.

[2] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao, and S. Banerjee, "Mobilityfirst: a mobility-centric and trustworthy internet architecture," *Acm Sigcomm Computer Communication Review*, vol. 44, no. 3, pp. 74–80, 2014.

[3] B. Han, J. Li, J. Su, and M. Guo, "Secrecy capacity optimization via cooperative relaying and jamming for wanets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 1117–1128, 2015.

[4] S. Jain, S. Gopinath, and D. Raychaudhuri, "Star: Storage aware routing protocol for generalized delay tolerant networks," in *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks*, 2011, pp. 1–4.

[5] S. Paul, R. Yates, D. Raychaudhuri, and J. Kurose, "The cache-and-forward network architecture for efficient mobile content delivery services in the future internet," in *Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference*, 2008, pp. 367–374.

[6] N. Somani, A. Chanda, S. C. Nelson, and D. Raychaudhuri, "Storage aware routing protocol for robust and efficient services in the future mobile internet," in *IEEE International Conference on Communications*, 2012, pp. 5849–5853.

[7] C. Wang, K. Long, X. Gong, and S. Cheng, "The study and perspective of queue scheduling algorithms in packet switching networks," *ACTA ELECTRONICA SINICA*, vol. 29, no. 4, pp. 553–559, 2001.
[8] J. F. Kurose and K. W. Ross, "Computer networking: A top-down approach (6th edition)," vol. 18, no. 142, p. 111, 2012.
[9] NS-3, https://www.nsnam.org, 2011.
[10] N. Somani, "Performance evaluation of forwarding algorithms for generalized storage aware routing protocols," *Learning and Instruction*, vol. 32, no. 3, pp. 73–80, 2012.