Routing Protocol Design for Directional and Buffer-limited Terahertz Communication Networks

Qing Xia*, Josep Miquel Jornet†

*Department of Electrical Engineering
University at Buffalo, Buffalo, NY, USA, E-mail: qingxia@buffalo.edu

†Department of Electrical and Computer Engineering
Northeastern University, Boston, MA, USA E-mail: jmjornet@northeastern.edu

***Abstract*— Terahertz (THz) band (0.1-10 THz) communication is envisioned as a potential key wireless technology to satisfy the need for much higher wireless data rates. THz-band communication supports a huge bandwidth. However, this advantage comes at the cost of a very high propagation loss. Thus, highly directional antennas (DAs) are simultaneously utilized in both transmission and reception to establish communication links beyond several meters. The application of highly DAs introduces many challenges for multi-hop routing. Among others, the best routing path dynamically changes since the directional communication links are periodically on and off, as determined by the DAs' current directions. Another challenge for routing protocol design comes from the limited memory or buffer size of THz devices, which is filled quickly when concurrent Terabit-per-second (Tbps) transmissions are handled. The buffer will be easily blocked by a locally stored packet that keeps waiting for the availability of the "best" route. This issue becomes even worse in directional networks, where such route may not be available shortly, and severely affects the network performance. In this paper, an adaptive routing protocol for highly dynamic buffer-limited directional THz communication networks is developed. A simulation framework is developed to study the iterations and updates between network performance and the choice made by each node. Extensive simulation results are provided to demonstrate the improvements of our proposed routing protocol.**

I. Introduction

With the drastic growing amount of wireless devices connected to the Internet, a huge amount of wireless data traffic is been created, shared and consumed every day in our society. At the same time, wireless data rates have grown 18-fold over the last three decades [1], and are approaching the capacity of the wired communication systems. In this context, Terahertz (THz)-band (0.1-10 THz) communication is envisioned as a key technology to satisfy the demand for such very high data-rate [2], [3].

An unprecedentedly large bandwidth is provided by the THz-band, which ranges from several tens of GHz up to a few THz [4]. However, this advantage comes at the cost of a very high propagation loss, which is mainly caused by the spreading loss and the molecular absorption loss at THz frequency. Besides, the limited power of THz transceivers (from tens of microWatts [5] to tens of milliWatts [6]) can only support short range communication with transmission distance of less than one meter. For these reasons, highly directional antennas (DAs) are needed simultaneously in transmission and reception to achieve longer communication range.

The need of very highly DAs introduces many challenges when we move up in the protocol stack. For instance, at the link layer, synchronization is needed to overcome the deafness problem between the transmitters and the receivers [7]–[9]. To overcome this issue, we have recently proposed the use of high speed turning DAs [10]. Although the proposed method solves the deafness problem, it introduces other challenges in several aspects including the neighbor discovery [11], [12], relaying distribution strategies [13] and multi-hop routing. Similar with the issue of multi-hop routing in directional Mobile Ad hoc Networks (MANETs) [14], [15], the best routing path in THz-band communication networks dynamically changes. More specifically, the directional communication links between neighbors are periodically on and off, as determined by the DAs' current directions. As a result, no consistent paths between neighbor pairs exist.

Another challenge for the routing protocol design comes from the limited memory or buffer size of each node. Especially for the THz nodes, which are easier to run out of memory when concurrent Terabits-per-second (Tbps) transmissions are handled. If a locally stored packet keeps waiting until the "best" relay becomes available, it will easily block the buffer and cause other packets to be dropped. The blockage issue becomes even worse in a directional network, in which the "best" relay may not be available shortly. To avoid this problem, THz nodes should keep packets moving, even without being certain of the accuracy of the best route. However, blindly transmitting packets over to any of the existing routes is not encouraged for the lack of adaptivity.

In conventional routing protocol designs, the packets are forwarded to the best relay according to the routing table information. The routing table is updated periodically as in, e.g., proactive routing, or updated on demand as in, e.g., reactive routing. However, the conventional routing protocols cannot be simply reused in THz communication networks, because they do not capture the realtime dynamic variation of the link connectivities and the buffer availabilities of the adjacent relays. As a comparison, the bufferless routing [16] in On-Chip Networks further enables packet deflection to the relay with the second highest priority when the forwarding

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

path is not available. Although the solution alleviates the buffer blockage issue by providing the deflection option, it is still not a realtime adaptive routing algorithm. Thus, a routing protocol which can capture the realtime system performance and adaptively select the next relay is needed.

In this paper, we propose an adaptive routing protocol based on Q-learning algorithm to capture the realtime peculiarities of THz communication networks. Q-learning algorithm has been used for highly dynamic routing protocol design in, for example, optical burst-switched networks [17] and underwater communication networks [18]. However, the design of the reward metrics in the existing solutions do not capture the peculiarities in THz communiation. For this, we design the realtime rewards and the corresponding update policies.

More specifically, the unique characteristics in the ultra-broad band THz networks are introduced by the expected very high traffic load and the equally high latency as a consequence of directional network. Thus, we design the rewards from the perspective of the DAs facing time priority and the traffic status on each link. For the time reward design, we consider the waiting time for the nodes' DAs to face each other. For link reward design, we consider both buffer occupancy rate of the neighboring node and traffic load condition on the communication link, which includes collision, arrival, forwarding, deflection and reflection (packet is reflected when next relay is unavailable). Besides, a dual reinforcement mechanism is utilized to achieve faster adaptivity in the highly dynamic network. Moreover, nodes should keep the relayed packets moving fast to avoid blockage. Thus, other than the "optimal" relay with the highest priority, several backup relays that have lower priorities can also be the next choice. Then, we describe the simulation framework based on the proposed routing protocol. In the end, we verify the performance improvement of the proposed routing protocol with extensive MATLAB simulation results, which we also benchmark against the shortest path routing protocol.

The remaining of this paper is organized as follows. First, in Sec. II, we introduce the network topology, the unit time of the directional communication system, and the principle of the Q-learning algorithm. In Sec. III, we describe the routing information of the proposed routing protocol, the reward metrics and Q-table definition. Further, in Sec. IV, we describe the detailed protocol operation. In Sec. V, we validate the performance of the routing protocol and compare it with the shortest path routing protocol. We illustrate the improvement of the network performance with extensive simulation results. Finally, we conclude the paper in Sec. VI.

II. System Model

In this section, we first provide an introduction of the network topology considered in this paper. Then, we describe the unit time in the directional communication system. In the end, we introduce the principle of Q-learning algorithm.

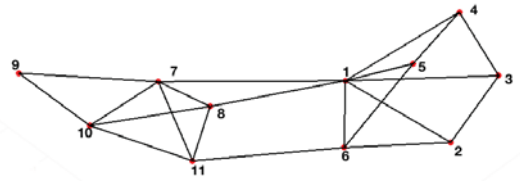


Fig. 1: Network Topology

A. Network Topology

The network topology we studied in this paper is shown in Fig.1. There are $\mathcal{N}$ nodes randomly distributed in the network, following the Poisson distribution. Because of the ultra fast transmission speed, we consider that, each node's learning speed during the proposed routing process is much faster than the node's moving velocity. And, thus, at any simulation time, the network topology remains static. In addition, following neighbor discovery, we consider that each node is aware of its neighbors' location. We denote nodes' connectivities with an $\mathcal{N} \times \mathcal{N}$ matrix $\boldsymbol{M}^{\boldsymbol{nodes}}_{\boldsymbol{connect}}$. For any node pair (i, j), we use $\boldsymbol{M}^{\boldsymbol{nodes}}_{\boldsymbol{connect}}(i, j) = 1$ to represent the communication link between sender i and receiver j is successfully established, where i and j are the row index and the column index of $\boldsymbol{M}^{\boldsymbol{nodes}}_{\boldsymbol{connect}}$, respectively.

B. Unit Time in Directional Communication System

Highly DAs are utilized simultaneously in transmission and reception, and are periodically sweeping the entire area at the same constant speed, but not necessarily in the same direction [10]. We consider that in every DA rotating cycle T_{cycle}, the DAs of each neighbor pair face each other once. Thus, the communication system iterates and updates information every T_{cycle}, which is also considered as the system unit time. Thus,

$$T_{cycle}(d_T) = \left(\frac{L_{data} + N_{ctrl} L_{ctrl}}{R(d_T)} + 2T_{prop}(d_T) \right) \frac{2\pi}{\Delta\theta(d_T)}, \tag{1}$$

where d_T is the distance from the transmitter to the receiver. We consider that all DAs utilize the same beamwidth $\Delta\theta$ and, thus, the same transmission distance for all nodes. R refers to the data-rate in the 3 dB frequency window, T_{prop} is the signal propagation delay. L_{data} and L_{ctrl} represent the frame length of the data packet and the control packet, respectively. N_{ctrl} is the number of control packets. The antenna beamwidth $\Delta\theta$ is calculated as [13]:

$$\Delta\theta(d_T) \leq \sqrt{4\pi \sqrt{\frac{\int_{B(d_T)} S_t(f) \frac{c^2}{(4\pi d_T f)^2} e^{-k_{abs}(f) d_T} df}{N_r(d_T)\, SNR_{min}}}}, \tag{2}$$

where B stands for the 3 dB bandwidth, S_t is the single-sided power spectral density (p.s.d) of the transmitted signal, f refers to frequency, c stands for the speed of light in the vacuum, k_{abs} is the molecular absorption coefficient of the medium, N_r denotes the molecular absorption noise power. SNR_{min} stands for the minimum SNR threshold.

C. Q-Learning Principles

The Q-learning algorithm is derived from the definition of Q-values, which are denoted based on the state-action pairs $Q(s_t, a_t)$, meaning the system takes action a_t in state s_t at time t. The system jumps from one state to another in discrete-time steps. The rewards are thus received after taking actions at certain states. The Q-learning algorithm is formulated as [19]:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_a Q(s_{t+1}, a)\right), \tag{3}$$

where $\alpha \in (0, 1]$ is the learning rate, which indicates to what extent the latest updated Q-value overrides the previous one. A factor of 0 or 1 indicates that the system totally ignores or exploits the most recent reward information correspondingly. The factor r_t represents the direct reward received after the system takes action a_{t-1} from state s_{t-1} and is observed for the current state s_t. The discount factor γ ranges within $(0, 1]$, and determines the importance of future rewards.

In the rest of this paper, if not specifically stated, all matrices are the simplified modules that summarily describe the iterations of system performance or a transient system state of the realtime simulation process.

III. Protocol Overview

In this section, we first describe the routing information of the proposed routing protocol. Then, we introduce reward metrics and Q-table definition.

A. Routing Information

We consider that each relay has a First-In-First-Out (FIFO) buffer and maintains a routing table and a Q-table that contains the latest Q-values of its adjacent relays. The routing table is used to decide the next relay with the highest priority in order to reach each destination, and is updated on demands based on the hop count router metric. Before sending out the first packet from its buffer, the relay checks the Q-table. The Q-table maintains the rank of the neighboring nodes in order to reach each destination based on the realtime rewards. The rewards mainly come from the DAs facing time priority (*time rewards*) and the traffic status on each link (*link rewards*). The *time rewards* stand for the priorities with regard to the waiting time of next facing for node pairs. The *link rewards* are contributed by several factors, which include the buffer occupancy rate of neighboring nodes, the traffic load conditions on each link by considering collision and arrival, as well as the shortest path factors indicated by forwarding, deflection and reflection.

In order to maintain the rewards updated in realtime, the proposed routing protocol incorporates both forward and backward Q-learning algorithms. For the forward Q-learning algorithm, when a node hears a packet, it extracts the neighbor's information and updates the corresponding entry in its neighbor list. In contrast, for the backward Q-learning algorithm, after a node received a data packet, it sends a very small feedback packet to the sender, so that the sender can update the rewards as well.

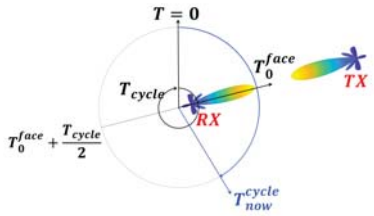


Fig. 2: Time rewards analysis

B. Reward Metrics

The time rewards analysis is based on Fig. 2. The receiver DA keeps rotating while the transmitter DA maintains pointing to the receiver [10]. We define the duration of one DA rotating cycle as T_{cycle} and the initial time of the rotating cycle is $T = 0$. $\boldsymbol{T_0^{face}}$ is the first facing time of the transmitter DA and the receiver DA, which is set up during neighbor discovery procedure of each node. We define $\boldsymbol{T_{now}^{cycle}}$ as the time duration after the receiver DA passes away from initial time, which is calculated as:

$$\boldsymbol{T_{now}^{cycle}} = \left\| t\boldsymbol{M_{connect}^{nodes}} / T_{cycle} \right\|, \tag{4}$$

where the current time t is only mapped to those connected nodes by multiplying with $\boldsymbol{M_{connect}^{nodes}}$. The modulus operator $\|\cdot\|$ renews $\boldsymbol{T_{now}^{cycle}}$ in every cycle. Then, the time reward matrix is formulated as:

$$\boldsymbol{R_t} = \begin{cases} \frac{\boldsymbol{T_{now}^{cycle}}}{T_{cycle}}, & \text{if } 0 \le \boldsymbol{T_{now}^{cycle}} < \boldsymbol{T_0^{face}}; \\ -\frac{\left(\boldsymbol{T_{now}^{cycle}} - \boldsymbol{T_0^{face}}\right)}{T_{cycle}}, & \text{else if } \boldsymbol{T_{now}^{cycle}} < \boldsymbol{T_0^{face}} + \frac{T_{cycle}}{2}; \\ \frac{\boldsymbol{T_{now}^{cycle}} - \left(T_{cycle}/2 + \boldsymbol{T_0^{face}}\right)}{T_{cycle}}, & \text{otherwise}; \end{cases} \tag{5}$$

which indicates how fast the DAs of neighboring node pairs can face each other. The denominator T_{cycle} is used to normalize the time reward. When the first condition is satisfied, the node pair i and j haven't met each other yet. In this case, an increasing positive reward value is abtained as the receiver DA keeps rotating towards to the facing direction. When the second condition is met, i and j already missed each other. Thus, when the receiver DA keeps rotating farther away from the facing direction, an increasing negative reward value is gained to prevent the system from selecting this node as the next relay. In the last condition, the rotating receiver DA already passed the worst point at $\boldsymbol{T_0^{face}} + \frac{T_{cycle}}{2}$ and is rotating back toward the facing direction, an increasing positive reward value is assigned to this node again.

The link reward $\boldsymbol{R_l}$ is derived based on the buffer occupancy rate and the load traffic status on each link, and is presented as:

$$\begin{aligned} \boldsymbol{R_l} = & w_b \boldsymbol{M_{left}^{buf}} + w_a \boldsymbol{M_{arr}^{load}} + w_f \boldsymbol{M_{fwd}^{load}} \\ & - w_c \boldsymbol{M_{col}^{load}} - w_d \boldsymbol{M_{def}^{load}}, \end{aligned} \tag{6}$$

where $\boldsymbol{M_{left}^{buf}}$ is the buffer vacancy rate, which is multiplied by the corresponding weight value w_b. $\boldsymbol{M_{arr}^{load}}$ and $\boldsymbol{M_{fwd}^{load}}$ stand for the links that recently handled the arrived and forwarded traffic loads, respectively, and their corresponding weight values are denoted as w_a and w_f, respectively. These matrices

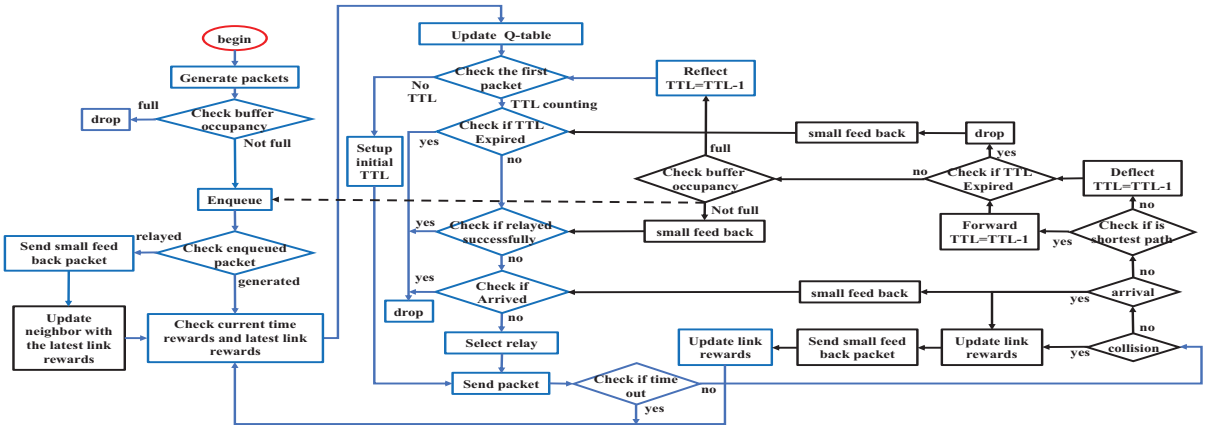


Fig. 3: Flow Chart of the Proposed Routing Protocol

will positively affect the link rewards so that the links with higher M_{arr}^{load} and M_{fwd}^{load} values have comparatively higher probabilities to be selected next time. On the contrary, $\boldsymbol{M_{col}^{load}}$ and $\boldsymbol{M_{def}^{load}}$ represent the links who suffered from collisions and deflections, respectively, and their corresponding weight values are denoted as w_c and w_d, respectively. The collision and deflection matrices are assigned with negative signs to prevent such links from being selected next time. We consider the same weight value for all matrices and set them as $1/3$ for the purpose of normalization.

Since reflection is only triggered by the rejection of a fully buffered relay, and the corresponding buffer information has been updated in $\boldsymbol{M_{left}^{buf}}$, we will not duplicate the buffer status information by involving reflection in the link reward function.

C. Q-table Definition

We formulate the basic framework of our router metrics by mapping the Q-learning algorithm to the proposed routing protocol and rewrite (3) as:

$$\boldsymbol{Q(s,a)} \leftarrow (1-\alpha)\boldsymbol{Q(s,a)}+\alpha\left(w_t\boldsymbol{R_t}+w_l\boldsymbol{R_l}+\gamma\boldsymbol{M_{relay}^{sp}}\right), \tag{7}$$

where the Q-table $\boldsymbol{Q(s,a)}$ maintains the Q-values for all neighbor pairs. w_t and w_l are the weights for time rewards and link rewards, respectively. For the highly dynamic network, the optimal future reward is very difficult to estimate. Moreover, it is even harder to derive the global optimum from a local optimum. Thus, we consider the shortest path between source and destination as the estimation of optimal future reward. Here, $\boldsymbol{M_{relay}^{sp}}$ is used to represent the next relays on the shortest path according to the routing table information.

IV. Protocol Operation

In this section, we describe the proposed routing protocol presented by the flowchart in Fig. 3. More specifically, we explain the major procedures in the protocol, including enqueue of new packets, selection of relay, reactions to collision and arrival as well as relaying. We mark the procedures carried out by each node itself with blue color and mark the procedures executed by the relaying nodes with black color.

The procedures are described by the matrices which contain the realtime information of each communication link. The matrices are updated every T_{cycle} according to system iterations and information updates during the simulation process. The system performance mainly constrained with two limitations including the bounded buffer size $\mathcal{N}_{max}^{buf}$ and the limited Time To Live (TTL) value $\mathcal{N}_{max}^{ttl}$. Dropping occurs when any of these two limitations is met.

A. Enqueueing of New Packets

There are two types of new packets entering a node, i.e., node generated packets and relayed packets. In both cases, the packet can only be enqueued in the buffer that is not fully occupied. Otherwise, the node generated packet is dropped and the relayed packet is rejected and reflected back to the previous sender. Since the buffer queue is a FIFO data structure, any new arrived packet is stored in the end of the buffer. If the new packet has been forwarded or deflected from a neighboring node, according to the forwarding Q-learning algorithm, the current node extracts the corresponding link information and updates link rewards of the entry. A small feedback packet is sent to the neighboring node who generated or relayed this packet. The feedback packet contains information of the residual buffer size and the latest link status of the current node.

In the simulation, we implement the buffer module $\boldsymbol{M^{buf}}$ as an $\mathcal{N}_{max}^{buf}\times\mathcal{N}$ matrix, where the column index indicates the node ID and the row index indicates the slot ID of the buffer. Each packet stored in $\boldsymbol{M^{buf}}$ carries three information fields including original source ID, Final Destination ID and current TTL value. The residual buffer size of each node is calculated by subtracting the number of used slots from the maximum buffer size, thus:

$$\boldsymbol{N_{left}^{buf}}(j)=\mathcal{N}_{max}^{buf}-\boldsymbol{slot}(j),\ \forall j\in[1,\mathcal{N}]\cap \left[\boldsymbol{M^{buf}}(\boldsymbol{slot}(j),j)>0\cap\boldsymbol{M^{buf}}(\boldsymbol{slot}(j)+1,j)=0\right], \tag{8}$$

where $\boldsymbol{N_{left}^{buf}}$ is a $1\times\mathcal{N}$ matrix, whose index j indicates node ID. The condition of $\boldsymbol{M^{buf}}(\boldsymbol{slot}(j),j)>0\cap\boldsymbol{M^{buf}}(\boldsymbol{slot}(j)+1,j)=0$ guarantees that $\boldsymbol{slot}(j)$ is the

index of the last occupied slot in buffer j. Thus, the buffer vacancy rate in (6) is calculated as:

$$\boldsymbol{M}^{\boldsymbol{buf}}_{\boldsymbol{left}} = \mathbf{1}^{\mathcal{N}\times 1}\boldsymbol{N}^{\boldsymbol{buf}}_{\boldsymbol{left}}\boldsymbol{M}^{\boldsymbol{node}}_{\boldsymbol{connect}}/\mathcal{N}^{buf}_{max}. \tag{9}$$

The node checks the first packet in the buffer. Thus, the packet transmission matrix is presented as:

$$\boldsymbol{M}^{\boldsymbol{G}}(j, \boldsymbol{M}^{\boldsymbol{buf}}(1,j).Dest) = 1, \forall j \in \left[\boldsymbol{N}^{\boldsymbol{buf}}_{\boldsymbol{left}}(j) \neq 0\right], \tag{10}$$

where j indicates the source node ID of the generated packet. $\boldsymbol{M}^{\boldsymbol{buf}}(1,j).Dest$ stands for the destination node ID of the first packet stored in node j.

B. Relay Selection

In order to avoid blockage in a dynamic network, the node should try to send out the buffered packets as soon as possible. To achieve this, at the beginning of current T_{cycle}, the node checks the DAs facing time priority with its neighbors and updates the latest time reward for each entry. The most recent link rewards have already been updated either by extracting information from the relayed packets that are enqueued in this node, or by obtaining the information from the small feedback packets sent from neighbors. For this reason, the node updates the Q-values for all the entries in its neighbor list based on (7), and configures the updated Q-table. The entry with the highest Q-value is selected as the next relay. In our design, we select other two backup relays as well. The priorities of the backup relays are also determined by their Q-values. In case the best relay is unavailable, the node sends the packet to another relay who has the next highest Q-value. To achieve this design, the node always sends the copy of the first packet instead of the original packet. The node then decides to keep or discard the first packet based on the feedback information sent by the next relay or the destination that receive the copied packet. Before sending out the copied packet, the node needs to check if the packet already has a valid TTL in counting. If this is the case, the node just sends out the copied packet to the best relay without any manipulation. Otherwise, the node is dealing with a new node generated packet. In this case, the node needs to setup the maximum TTL in the header of the packet, then sends the copied packet to the best relay. Thus, the relay selection matrix is presented as:

$$\begin{aligned}&\boldsymbol{M}^{\boldsymbol{P}}(j,p) = 1, \forall j \in \left[\boldsymbol{N}^{\boldsymbol{buf}}_{\boldsymbol{left}}(j) \neq 0\right], p \in [1,\mathcal{N}] \\ &\cap\, j, p \in \left[\max \boldsymbol{Q}(j,:) = \boldsymbol{Q}(j,p)\right],\end{aligned} \tag{11}$$

where $\boldsymbol{Q}$ is a data structure of the Q-table shown in (7). The condition of $\max \boldsymbol{Q}(j,:) = \boldsymbol{Q}(j,p)$ indicates that p is the best relay among all neighbors of j, as it has the highest Q-value. The second best relay matrix is derived by applying the same logic with selecting the relay based on the second highest Q-value from the neighbor list of the sender.

C. Collisions and Arrivals

The collision occurs when more than one sender contends for the same relay at the same time. More specifically, all senders are within one sector coverage area of the receivers' DA, and are pointing their DAs toward the receiver simultaneously, which rarely occurs. When there is a collision, the receiver can not extract the neighbor's information from the destroyed packet, but it can still update the collision information of the corresponding entry in its neighbor list. Then, before the receiver turns to the next sector, it broadcasts a small feedback packet towards the link that experienced collision. The sender retransmits the packet if no feedback packet that indicates successful arrival or relaying is received before time out. The collision matrix is shown as:

$$\begin{aligned}&\boldsymbol{M}^{\boldsymbol{load}}_{\boldsymbol{col}}(j^1,p) = ... = \boldsymbol{M}^{\boldsymbol{load}}_{\boldsymbol{col}}(j^n,p) = 1, \forall j^1 \neq ... \neq j^n, p, \\ &n \in [1,\mathcal{N}] \cap [\boldsymbol{M}^{\boldsymbol{P}}(j^1,p) = 1] \cap ... \cap [\boldsymbol{M}^{\boldsymbol{P}}(j^n,p) = 1] \cap \\ &[\boldsymbol{R}_{\boldsymbol{t}}(j^1,p) = ... = \boldsymbol{R}_{\boldsymbol{t}}(j^n,p)],\end{aligned} \tag{12}$$

which means there are totally n senders, denoted as $j^1, ..., j^n$, which sent their packets to the same relay p at the same time according to the same time reward.

When a packet arrives at destination we assume it is not put into any buffer. Instead, the receiver directly processes and checks the header of the coming packet during reception. In this case, the packet arrives at the destination without any buffer size limitation. The receiver extracts the link information from the packet and updates the link reward to the previous sender with a small feedback packet. The arrival matrix is presented as:

$$\begin{aligned}\boldsymbol{M}^{\boldsymbol{load}}_{\boldsymbol{arr}}(j,p) =& 1, \forall j, p \in [1,\mathcal{N}] \cap \left[\boldsymbol{M}^{\boldsymbol{P}}(j,p) = 1\right] \\ &\cap \left[p = \boldsymbol{M}^{\boldsymbol{buf}}(1,j).Dest\right],\end{aligned} \tag{13}$$

meaning the selected relay ID is exactly the destination node ID of the first packet stored in the buffer of the sender. Once a node receives the feedback packet indicating the copy of the first packet has arrived, the node discards the first packet.

D. Relaying

If the packet neither collides with other packets nor arrives at the destination, it is relayed. In this case, the packet may experience three types of procedure including forwarding, deflection and reflection.

Forwarding occurs when the packet is sent to exactly the best relay between the previous node and the destination according to the shortest path algorithm. Otherwise, the packet is considered as *deflected* to next relay. In both cases, since the packet already attempted to access the next relay, the TTL of the packet is decreased by one. If the TTL counts down to zero, it expires. This information is synchronized at the sender by receiving a small feedback packet sent from the next relay. In this case, both the copied packet and the original packet have to be dropped. Otherwise, the copied packet should wait to be enqueued in the buffer of the forwarder or the deflector. If the remaining buffer size of the next relay is available to handle one more packet, the copied packet is stored in the end of the next relay's buffer and the process repeats from the beginning as we described in Sec. IV-A. Correspondingly, the sender discards the first packet from its buffer after receiving the feedback packet from the next relay. However, when next

relay's buffer is fully filled, the relay rejects the copied packet, which is the *reflection* procedure. The next relay first processes the header of the copied packet and decreases the TTL by one, then sends it back to the sender. After the sender receives the reflected copied packet, it checks the TTL in packet header, if the TTL is expired, the sender discards both the copied packet and the original packet. If the TTL of the copied packet is still valid, the sender synchronizes the latest TTL in the header of the original packet and tries to send the copied packet again to the next best relay with a lower priority in current T_{cycle}. Thus, the forwarding matrix and deflection matrix are derived as follows:

$$\begin{aligned} \boldsymbol{M}_{\boldsymbol{fwd}}^{\boldsymbol{load}} = &\left(\boldsymbol{M}^{\boldsymbol{P}} - \boldsymbol{M}_{\boldsymbol{col}}^{\boldsymbol{load}} - \boldsymbol{M}_{\boldsymbol{arr}}^{\boldsymbol{load}}\right) \\ &\cdot \boldsymbol{M}_{\boldsymbol{relay}}^{\boldsymbol{sp}} \left[\mathcal{L}\left(\boldsymbol{M}^{\boldsymbol{buf}}(1,:).TTL\right)\right]^T \mathbf{1}^{1\times\mathcal{N}}, \end{aligned} \tag{14}$$

$$\begin{aligned} \boldsymbol{M}_{\boldsymbol{def}}^{\boldsymbol{load}} = &\left(\boldsymbol{M}^{\boldsymbol{P}} - \boldsymbol{M}_{\boldsymbol{col}}^{\boldsymbol{load}} - \boldsymbol{M}_{\boldsymbol{arr}}^{\boldsymbol{load}} - \boldsymbol{M}_{\boldsymbol{fwd}}^{\boldsymbol{load}}\right) \\ &\cdot \left[\mathcal{L}\left(\boldsymbol{M}^{\boldsymbol{buf}}(1,:).TTL\right)\right]^T \mathbf{1}^{1\times\mathcal{N}}, \end{aligned} \tag{15}$$

where $\mathcal{L}(.)$ is the logical function that converts numeric values to logical values, which indicates the validity of the TTL of the first packets stored in the buffers.

V. Simulation Results

In this section, we investigate the performance of the proposed routing protocol meanwhile the traffic load increases in the system. The network topology is illustrated in Fig. 1. MATLAB is utilized to simulate the routing protocol. Several performance metrics are used to demonstrate the benefits of our proposed protocol, which include traffic performance, network throughput and node recovery rate. The node recovery rate is calculated as the average number of decreased loads on the hot points during simulation. For each set of parameters, simulations are ran 20 times with the duration of 300 T_{cycle}.

We assign the learning rate α and the discount factor γ in (7) as 0.5 and 0.5, respectively, as we discussed in Sec. II-C. The weights of time reward and link reward are both assigned to 0.5, as we consider they are equally important. The optimal set of all weights will be developed in our future work. During each T_{cycle}, each node generates at least one packet to the randomly selected destination node. We assign the TTL limitation $\mathcal{N}_{max}^{ttl} = 11$, which is same as the total number of nodes in the network. We apply the same physical layer setting as introduced in [10].

We analyze two scenarios, a scenario with light traffic load as shown in Fig. 4, and another scenario with heavy traffic load as shown in Fig. 5. In the light traffic load scenario, each node generates 1 packet per T_{cycle}. We test the performance with an increasing maximum buffer size $\mathcal{N}_{max}^{buf}$, which changes from 100 to 500. We compare the performance of applying the shortest path (SP) routing protocol with the proposed routing protocol (BDT). As shown in Fig. 4a and 4d, the network is more active with BDT protocol. More specifically, BDT protocol encourages packets to explore different relays, we observe that packets pass through more hops with BDT protocol than that with SP protocol during simulation. The comparison of Fig. 4b and 4e illustrates that the buffer blockage rate (green line) with BDT protocol is significantly lower than that with SP protocol. As a result, the packet arrival rate (purple line) is improved with BDT protocol. However, the improvement comes with the cost of detouring and, thus, we see drops caused by TTL expiration (yellow line) occur in Fig. 4e as well as the lower average throughput as shown in Fig. 4c. Thanks to the learning capability, we observe from Fig. 4f that BDT protocol is more adaptive to the light loaded dynamic network, which is indicated by a higher average recovery rate.

The test with heavy traffic load is designed with a fixed maximum buffer size $\mathcal{N}_{max}^{buf} = 100$ and the packet generation rate of each node changes from 1 to 11 per T_{cycle}. As shown in Fig. 5a and 5d, with both SP and BDT protocols, there are more traffic activities than that in the light load scenario. Still, the traffic with BDT protocol is much more active. Because the fast generated traffic can easily surpass the buffer capability, we observe reflections occur in Fig. 5d. In Fig. 5b and 5e, BDT still performs much better than SP in terms of both buffer blockage rate (green line) and packet arrival rate (purple line). In both scenarios, dropping caused by limited buffer size occurs (blue line). And, with the increasing traffic loads, limited buffer size gradually become the major reason of dropping. In Fig. 5c, the packets detouring in the BDT scenario still lead to a lower network throughput. The gap between the average throughput of two cases has increased, but still in the same scale. In Fig. 5f, the recovery rates in both protocols are almost the same and are both higher than that in the light loaded condition. However, it is difficult to observe the contribution of learning capability just based on the recovery rate in such an overloaded and thus extremely dynamic network.

VI. Conclusion

In this paper, we proposed an adaptive routing protocol for the highly dynamic buffer-limited directional THz-band communication networks. The proposed routing protocol is described in a simulation framework based on the Q-learning algorithm. Mechanisms to determine the time rewards and link rewards are provided based on the DAs performance, buffer status and all possible link traffic status. The extensive simulation results have been presented to illustrate the improvement of the proposed protocol in reducing the buffer blockage rate and achieving higher packet arrival rate.

Acknowledgment

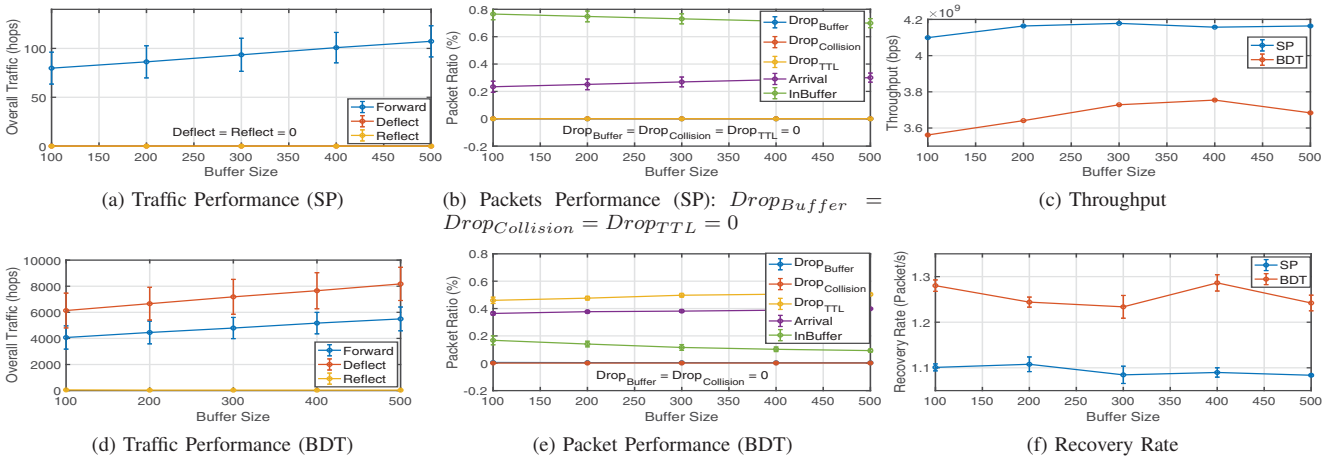
This work was partially supported by the U.S. National Science Foundation under Grant CNS-1846268.

References

[1] Cisco visual networking index: Global mobile data traffic forecast update, 20162021 white paper.

[2] I. F. Akyildiz, J. M. Jornet, and C. Han, "Terahertz band: Next frontier for wireless communications," *Physical Communication*, vol. 12, pp. 16–32, 2014.

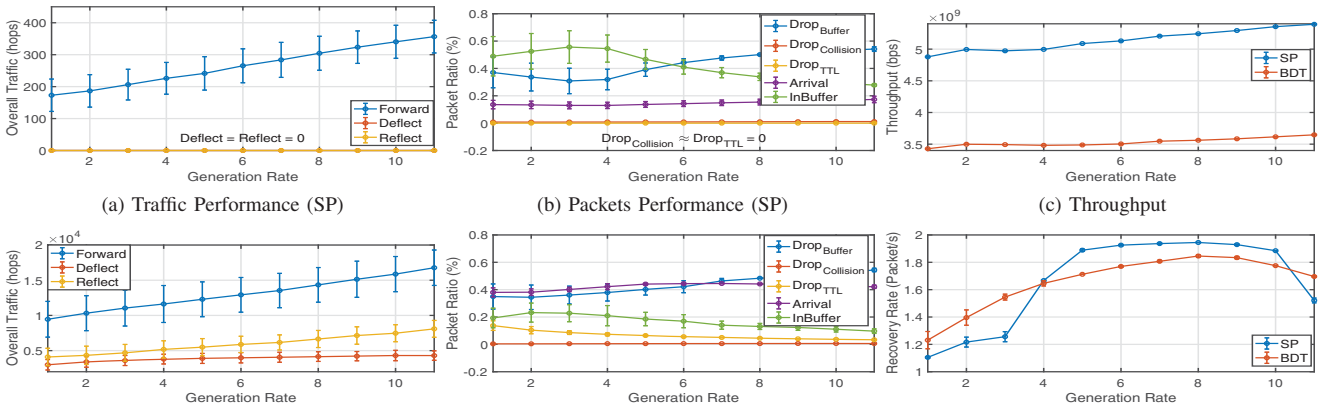
[3] T. Kurner and S. Priebe, "Towards THz Communications-Status in Research, Standardization and Regulation," *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 35, no. 1, pp. 53–62, 2014.



(a) Traffic Performance (SP) (b) Packets Performance (SP): $Drop_{Buffer} = Drop_{Collision} = Drop_{TTL} = 0$ (c) Throughput

(d) Traffic Performance (BDT) (e) Packet Performance (BDT) (f) Recovery Rate

Fig. 4: Comparison of System Performance in Light Traffic Load Scenario



(a) Traffic Performance (SP) (b) Packets Performance (SP) (c) Throughput

(d) Traffic Performance (BDT) (e) Packet Performance (BDT) (f) Recovery Rate

Fig. 5: Comparison of System Performance in Heavy Traffic Load Scenario

[4] J. M. Jornet and I. F. Akyildiz, "Channel modeling and capacity analysis of electromagnetic wireless nanonetworks in the terahertz band," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3211–3221, Oct. 2011.

[5] V. Radisic, K. Leong, D. Scott, C. Monier, X. Mei, W. Deal, and A. Gutierrez-Aitken, "Sub-millimeter wave InP technologies and integration techniques," in *IEEE MTT-S International Microwave Symposium (IMS)*, May 2015, pp. 1–4.

[6] S. Slivken and M. Razeghi, "High power, electrically tunable quantum cascade lasers," in *SPIE OPTO*. International Society for Optics and Photonics, 2016, pp. 97 550C–97 550C.

[7] S. Hur, "Millimeter wave beamforming for wireless backhaul and access in small cell networks and practical approaches in software-defined radio," Ph.D. dissertation, West Lafayette, IN, USA, 2013.

[8] K. Chandra, R. V. Prasad, B. Quang, and I. G. M. M. Niemegeers, "Cogcell: cognitive interplay between 60 ghz picocells and 2.4/5 ghz hotspots in the 5g era," *IEEE Communications Magazine*, vol. 53, no. 7, pp. 118–125, July 2015.

[9] W. Tong and C. Han, "Mra-mac: A multi-radio assisted medium access control in terahertz communication networks," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[10] Q. Xia, Z. Hossain, M. J. Medley, and J. M. Jornet, "A link-layer synchronization and medium access control protocol for terahertz-band communication networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[11] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 4, March 2005, pp. 2502–2512 vol. 4.

[12] J. Ning, T.-S. Kim, S. V. Krishnamurthy, and C. Cordeiro, "Directional neighbor discovery in 60 ghz indoor wireless networks," *Performance Evaluation*, vol. 68, no. 9, pp. 897 – 915, 2011, special Issue: Advances in Wireless and Mobile Networks.

[13] Q. Xia and J. M. Jornet, "Cross-layer analysis of optimal relaying strategies for terahertz-band communication networks," in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2017, pp. 1–8.

[14] H. Li and Z. Xu, "Routing protocol in vanets equipped with directional antennas: Topology-based neighbor discovery and routing analysis," *Wireless Communications and Mobile Computing*, 2018.

[15] N. Eshraghi, B. Maham, and V. Shah-Mansouri, "Millimeter-wave device-to-device multi-hop routing for multimedia applications," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–6.

[16] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," *SIGARCH Comput. Archit. News*, vol. 37, no. 3, pp. 196–207, Jun. 2009.

[17] S. Haeri, M. Arianezhad, and L. Trajkovic, "A predictive q-learning algorithm for deflection routing in buffer-less networks," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2013, pp. 764–769.

[18] T. Hu and Y. Fei, "Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 796–809, June 2010.

[19] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.