Intelligent Software-Defined Mesh Networks With Link-Failure-Adaptive Traffic Management

Ke Bao* Fei Hu*† Sunil Kumar‡
*Electrical and Computer Engineering, The University of Alabama, USA († Corresponding author)
‡Electrical and Computer Engineering, San Diego State University, San Diego, CA, USA

***Abstract*—Wireless Mesh Networks (WMNs) have been studied for a long term due to its flexibility and wide coverage area. Nevertheless, compared with the wired network, the performance of the WMN is still limited due to its dynamic link quality, low bandwidth, difficulties of network management, etc. To overcome these drawbacks, we propose an intelligent Software-Defined Networking based WMN architecture (SD-WMN) in which the entire WMN is monitored and managed in a centralized fashion. The network performance of the WMNs is improved through an intelligent control strategy with a global traffic management. First, a linear programming method is employed to optimize the traffic distribution and wireless link scheduling to take the advantage of SDN's centralized control structure. Second, two supervised-learning-based classification methods are adopted to cope with the unexpected link failure problems caused by node mobility. We then propose a back-up route seeking strategy under link failure, based on our proposed traffic management model. The proposed SD-WMN paradigm is validated in ns-3. Our results demonstrate the throughput enhancement by using the proposed SD-WMN architecture with link-failure-adaptive traffic management.**

Index Terms: Software-Defined Networking (SDN), Wireless Mesh Network (WMN), Link Failure Prediction, Supervised Learning, Traffic Management, Mobility.

I. Introduction

Wireless Mesh networks (WMNs) have drawn many attentions in the past decade. A WMN has some powerful wireless nodes, i.e., Mesh routers (MRs), which aim to build the backbone of a WMN and provide peer-to-peer connectivity to end users in an ad-hoc fashion. Although MRs are able to freely form various network topologies to adapt to varying radio conditions, the WMNs still have many performance limits such as imbalanced bandwidth allocations, difficulties of performing centralized network management for many distributed nodes, etc.

On the other hand, to enable an easy network management and monitoring, the concept of Software-Defined Networking (SDN) has been proposed. A network-wide central controller in SDNs is a firm support for the large, complex network management. The SDN-based network structure has been widely studied in *wired* networks, in which SDN enables a centralized control by separating the control plane from the data plane. Due to its advantages a few elementary studies have expanded the platforms of SDNs from wired networks to wireless scenarios. The SDN is a promising solution to overcome the drawbacks of current WMNs through its agile protocol reconfiguration and optimized resource allocation.

As such, we propose a SDN-based mechanism to improve the performance of WMNs in terms of dynamic routing management under mobility-caused link failures and its associated traffic allocation issues. To adapt to the dynamic WMN topology, we perform a centralized machine learning in the control plane for the prediction of unexpected link failure in WMNs. The proposed work is called PLUS-SW (Prediction-based Link Uncertainty Solution in SD-WMN). Its contribution includes three aspects as follows:

(1) *Dynamic traffic management*: We propose a network-wide traffic management based on the centralized control of SDNs. The traffic balancing and radio resource allocation are periodically re-evaluated by the SDN controller according to the real-time traffic demands and the network topology dynamics. By using this approach, the performance of the WMN can be improved by a global, optimal traffic monitoring and management.

(2) *Intelligent link failure prediction*: SD-WMN needs to handle the unexpected link failure due to node mobility. Conventional distributed WMN management has difficulties to accurately predict the link failure. We employ the supervised learning methods to predict unexpected link failures. This prediction system has a *two-layer* structure to fully explore the advantage of the SDN: In the lower layer, a supervised learning based classification model is used in the SDN switches; In the higher layer, the central controller uses a global learning model to enhance the accuracy of the link failure prediction in the entire WMN.

(3) *Optimal back-up path determination*: Once a link failure is confirmed by the two-layer prediction model, the SDN controller immediately calculates the alternative path and notifies the related nodes to bypass the link that will fail soon. The calculation of the back-up path is based on the pattern of current traffic. The control overhead and traffic congestion caused by the path switching are minimized.

II. Background and Motivation

In SDN the control plane comprises of one or multiple centralized controllers to manage the entire network. The data plane contains underlying routers and switches, which serve as the data forwarding devices. The control plane manages the entire network by writing or updating the forwarding rules in the flow tables of each switch. Correspondingly, a SDN-based switch handles all the ingress packets according to the flow tables stored in its memory. Such a structure allows a network-wide observation and control of the network traffic, as well as

a simplified hardware/software structure for the switches in the data plane.

A WMN consists of Mesh Routers (MRs), Mesh Clients (MCs), and Gateways. The MRs constitute the backbone of WMNs, while a gateway enables the Internet connectivity. MCs stand for end users of a WMN. A SDN-based WMN (SD-WMN) [33] also has the separation of control plane and data plane. The basic structure of SD-WMN is shown in Figure 1. In the SD-WMN, a dedicated central controller is used for the data forwarding control by writing forwarding rules in the flow tables residing in each switch. This central controller could be a gateway or a MR of the WMN. Each MC is thus able to get rid of the complicated self-operating module, and turns into a flow-table-based switch for packet forwarding. The main advantages of SD-WMN include the simplified network management, low-cost MC architecture, and flexibility of vendor control, etc. [6][8].

The motivation of this study is to overcome the shortcomings of a typical SDN when it is applied to WMN traffic management. First, in current SDN structure, it does not have a robust wireless link control among SDN switches when handling the imbalanced traffic flows in WMNs. Second, it does not have a flexible rerouting function that can minimize the rerouting impact on the active data flows by using optimal traffic allocations. Third, while we target the accurate fine-grained network configuration under uncertain radio conditions of the WMN, it is also important to minimize the communication overhead generated by the control traffic.

Due to the round trip time between the controller and switches, a SD-WMN may not be able to deal with a random wireless link failure or node malfunction rapidly. To solve this issue, the most straightforward solution is to enable a localized path selection in a SDN-based switch and handle an uncertain link failure in a distributed manner. Nevertheless, an unforeseen change of the local rerouting behaviors tends to incur many packet loss events, since a SDN switch cannot timely and accurately find a proper back-up path. Therefore, we introduce a supervised learning mechanism to predict the uncertain link behaviors of a SD-WMN. After a prediction of the link failure or the node malfunction is made, a SDN controller can generate a globally optimal rerouting plan and sends it to the relevant switches to establish a back-up routing path.

In order to establish a fine-grained link failure prediction (LFP) for the SD-WMN, we introduce a two-layer prediction paradigm. The basic structure of two-layer LFP system is shown in Figure 2. Since SDN assumes dummy forwarding devices in the data plane, it is improper to add complicated modules in a SDN router. Thus, a pre-trained link prediction model (via supervised learning) is used in the SDN switches to accommodate for the simple structure of switches. Such a model can be calculated by a central controller in advance and pre-installed in the switches. Due to the myopic operations of SDN switches, such a model will only be responsible for the local link failure prediction with one-hop neighboring nodes. Once a potential link failure is detected, the switch will immediately feedback a warning request to the central controller, which will make global traffic allocations.

Besides the above switch-triggered lower-layer LFP, there is an upper-layer LFP located at the central controller, which is able to determine if any node is going to fail by sampling the parameters of the outgoing links of this node. For example, it will check whether a moving node M tends to incur abnormal amplitude fluctuations for its outgoing links. Once a neighbor of M predicts this uncommon situation and forwards a warning message to the central controller, the controller will monitor all the links of M. Based on the link quality pattern analysis, the controller can determine if node M is going to fail in the near future. If a failure is confirmed by the controller, a corresponding back-up routing path will be established.

III. Related Works

As the preliminary studies of SD-WMNs, only a few basic prototypes have been built so far. Those SD-WMN prototypes can be classified into two main categories (based on the pattern of transmissions in control / data traffic [13]), i.e. in-band and out-of-band. The out-of-band structure is consistent with the policy of conventional SDNs used for wired network, in which the packets of control traffic are transmitted separately with the data traffic through an independent channel. Dely et al. [8] proposed the SD-WMN prototype with out-of-band style. The separation of control / data traffic is realized in [25] by assigning different Service Set IDentifiers (SSIDs) to the control / data traffic. Due to the limited resources in WMNs, an in-band based SD-WMN structure, i.e., control / data traffic is transmitted in the same channel, have been realized as well. Both Detti et al. [8] and Yang et al. [33] have designed a SD-WMN platform with in-band fashion. Particularly, Detti et al. [8] employed OLSR as the routing protocol for both control and data traffic, while Yang et al. [33] established an in-band Openflow-based WMN platform for traffic balancing.

All these works aim to establish a platform for SDN-based WMN rather than propose a novel control policy to overcome the shortcomings of SD-WMNs. We believe that the centralized control offered by SDN provides an opportunity for a more intelligent networking management, for example, by using some machine learning schemes. Especially, a SDN can be very helpful to deal with the management difficulties caused by the dynamic WMN topology.

Regarding the wireless link quality estimation (LQE), some mechanisms have been proposed, such as PRR [5], ETX [7]), filtering (WMEWMA [31], KLE [27], fout-bit [10]), Regression (WRE [32]) and Supervised learning [29]. Two schemes have focused on the link failure prediction issues: in [21] it achieve the failure prediction via data mining algorithm, and in [16] a cross-layer design is proposed. For those works, LQE is achieved by the estimation of the link qualities instead of link failure events. Namely, they are not capable of accurately predicting the unexpected link failures. Their accuracy limit is due to a distributed, localized network management. The estimation of the link status depends on the performance metrics of the link (such as signal-to-noise ratio). Unnecessary network operations (such as rerouting, channel switching, etc.) can be triggered by a mistaken observation of the network. In this work, the central controller will be used to improve the precision of LFP in a global management fashion.

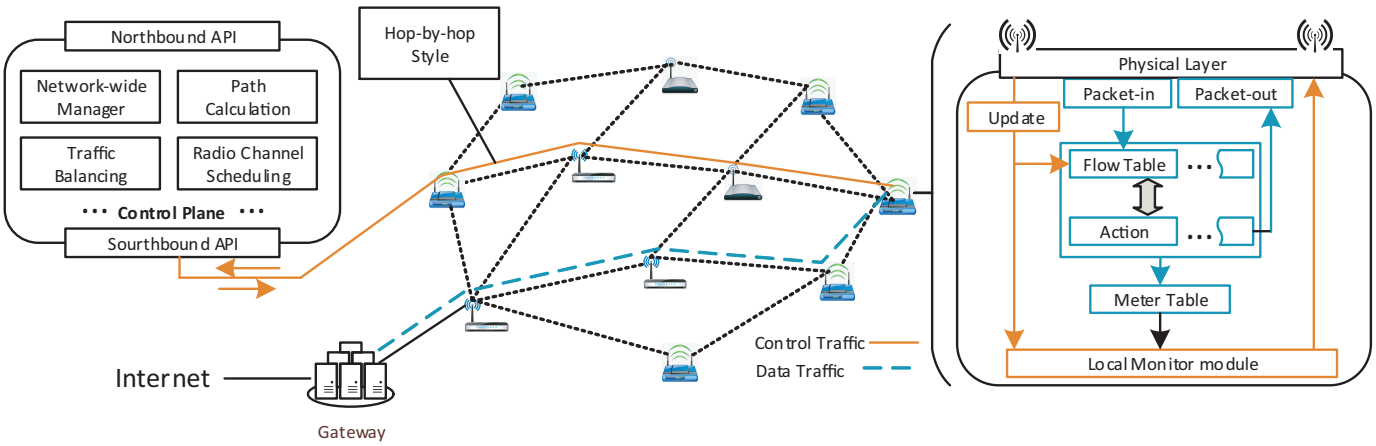


Fig. 1: SD-WMN Architecture

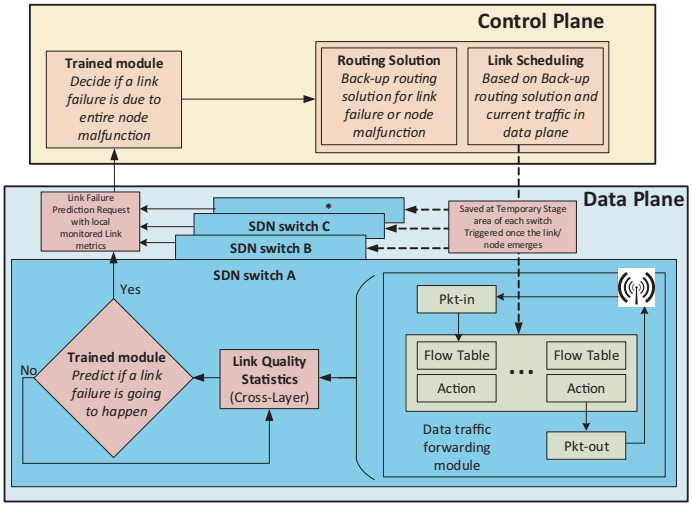


Fig. 2: Two-layer link failure prediction

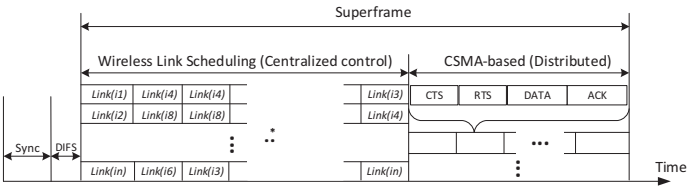


Fig. 3: Proposed SD-WMN MAC layer superframe structure

IV. Proposed Construction Model of SD-WMNs

A. Physical layer model

In the physical layer, various access/antenna technologies have been proposed to improve the bandwidth utilization, such as CDMA, OFDMA, MIMO, multi-beam directional antennas (MBDAs), etc. As in many WMN studies, we assume a multi-channel multi-radio (MC-MR) feature in the links. As an example, in 802.11a standard, there are 12 non-overlapping channels in the 5G Hz range.

B. Data link layer model

In a *wired* SDN, the control plane mainly performs the traffic engineering, and the traffic control commands are distributed to different routers/switches based on a global coordination plan of the control plane. Other than the wired SDN, a SD-WMN not only requires a global control of traffic engineering, a link access scheduling scheme to overcome the wireless link interference is also critical to minimize the link bit error rate (BER). As a result, a corresponding SD-WMN MAC layer is required.

Figure 3 shows the structure of the MAC layer for the proposed SD-WMN. It uses a hybrid scheduling. The time domain is divided into multiple time periods called superframes. Since a centralized link scheduling requires the synchronization of all the SDN switches, a short time slot is reserved to orchestrate the clocks of all the SDN switches. After a DCF Interframe Space (DIFS), the switches start to transmit data according to the link scheduling (calculated by the central controller in advance). Note that our link scheduling policy is adaptive to the link interference conditions in the dynamic network topology, and the scheduling calculation is based on the traffic demands and routing status, as well as the priorities of different data flows. More details about the link scheduling will be explained in Section V.

A centralized scheduling efficiently provides a global management to reduce link interference. Considering the link uncertainty and potential access conflicts, a CSMA-based schedule control is also used in the MAC protocol for link collision avoidance. Furthermore, the CSMA backoff policy is used for unexpected traffic collisions which may not be timely captured by the SDN controller. Thus CSMA-based distributed coordination in the local range can compensate for the difficulty of performing timely regulations by the remote controller.

C. Routing Protocol and Traffic Demand

Although the SDN is promising in terms of traffic balancing, the existing studies on SD-WMN still employ traditional routing protocols (e.g., AODV, DSR, OSLR, etc.) for the management of the network topology and the route discovery [33][8]. OLSR has strong capability of overcoming the impacts of node mobility. Once a SDN controller fetches the information of the SD-WMN topology and the traffic demands, it is able to distribute the data flows of each node to different links for optimized traffic engineering. While OLSR is capable of

performing path discovery, a controller still needs to optimize the traffic management in a global manner.

D. Mathematical model of the SD-WMN

In a network $G = (V, E)$, V represents the set of all vertices of a WMN, while E stands for all links. Let TD_v denote the traffic demand of flows generated in node v (note that the flows relayed by node v are not counted in TD_v). Since the transmissions via different channels are not influenced by each other, we first consider the data flows in single channel. Let $TD_v = \{f_P^{ID:1}, f_P^{ID:2}, f_P^{ID:3}, ..., f_P^{ID:n}\}$ denote traffic demands of the flows from node v with different priorities. Given a traffic distribution Δ_v, we can obtain the outgoing flows' demands in different paths of node v by using the following model:

$$\{f_{A1}^{\rho 1}, f_{A2}^{\rho 2}, ...f_{An}^{\rho n}\} = \Delta_v \times TD_v \qquad \forall v \in V \tag{1}$$

in which f_A^ρ implies an aggregated flow A that is sent out from node v along with path ρ. It is a sum of multiple outgoing flows through the path ρ. Hence, path ρ has a set of edges. f_A^ρ can be represented as:

$$f_A^\rho = \left\{ \alpha_A^{f(ID:1)} f_{P=in}^{ID:1}, \alpha_A^{f(ID:2)} f_{P=in}^{ID:2}, ..., \alpha_A^{f(ID:n)} f_{P=in}^{ID:n} \right\} \tag{2}$$

$\alpha_A^{f(ID:i)}$ stands for the ratio of outgoing flow $f_{P=in}^{ID:i}$ of node v assigned on the aggregated flow f_A^ρ along with path ρ. It is obvious that the aggregated flows on each link must be less than the capacity of the link. Let $c(e)$ represent the capacity of link e in time slot τ, we have:

$$\sum_{\rho \in f(e)} f^\rho \leq c(e) \quad \forall e \in E \tag{3}$$

The sum of all aggregated flows that go through edge *e* in time slot τ has to be smaller than the capacity of edge e.

E. Transmission Collision Model

The conflict model of wireless links denotes the wireless interference among a neighborhood. It can be categorized into protocol interference model (PrIM) and physical interference model (PhIM) [3]. The conflict model in this paper is based on PrIM [12], which is defined as follows: assuming node n_i and node n_j are one-hop neighbors with each other, and a data transmission is in progress between n_i and n_j in period via the radio channel ch_i. We say that the transmission is successful as long as all the neighbors of n_i and n_j that may interfere with link e, will keep silent in ch_i during the entire time slot . This concept is compatible with 802.11 MAC layer protocol (CSMA/CA). In the case that the nodes are equipped with directional antennas, the interfered nodes should be in the coverage of the transmission beam of n_i and reception beam of n_j. If an interfered node is equipped with MBDAs, then only those beams that can reach n_i or n_j must keep silent.

The network model with radio conflict is defined as follows [23]: Given a network model $G = (V, E)$, in which V denotes the set of all switches and E stands for all the edges. For each $e \in E$, there will be a set of edges which can disturb the wireless communications on the edge e, when they are transmitting data on the same channel. Here this set of edges is represented as $I(e)$. It is obvious that all the edges in $I(e)$ must keep silent in channel i while edge e is active on channel i. Given $f_e^{ch(i)}$ as the expected throughput in edge e during time slot τ, recall that $c_{ch(i)}(e)$ is the capacity of edge e in channel I during time slot τ. Then, the constraint can be represented by the equation below:

$$\frac{f_e^{ch(i)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(i)}}{c_{ch:i}(e')} \leq 1 \quad \forall e \in E \tag{4}$$

While edge e is idle, multiple edges in $I(e)$ could be activated. Hence we have:

$$\frac{f_e^{ch(i)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(i)}}{c_{ch:i}(e')} \leq C_{I(e)} \quad \forall e \in E, i \in K \tag{5}$$

The value of constant $C_{I(e)}$ is determined by the network topology and the link interference with each other. Thus we can calculate it when we know the link interference matrix of network G.

F. Flow smoothness

In order to prevent each node from queue overflow (i.e., traffic congestion), the input and output data amount for a relay node has to be equal in period T. When a flow f goes through a relay node v, node v may use different channels to receive and send flow f. Hence, we expand the traffic analysis from a single channel k to all the channels. Let $F_{v:in}^{ch:k}$ and $F_{v:out}^{ch:k}$ denote the amount of ingress and egress flows of node v, respectively. We have:

$$F_{v:in}^{ch:k} = \sum_{e \in E_{in}(v)} f_e^{ch:k}; \qquad F_{v:out}^{ch:k} = \sum_{e \in E_{out}(v)} f_e^{ch:k} \tag{6}$$

Assuming node v possesses K channels for data transmissions. Thus we have:

$$\sum_{k \in K} F_{v:out}^{ch:k} = \sum_{k \in K} F_{v:in}^{ch:k}$$
$$\forall v \in V, \{F_{v:out} \cup F_{v:in}\} \not\subset \{f(v, \sim), f(\sim, v)\} \tag{7}$$

Because a node v may be a source or destination of data flows, the amount of total ingress and egress flows should not contain the flows whose source or destination is node v. Here $f(v, \sim)$ and $f(\sim, v)$ represent those two kinds of flows, respectively. Thus, the amount of ingress and egress data of each relay node has to be equal when we ignore $f(v, \sim)$ and $f(\sim, v)$.

After listing all the constraints above, we can now maximize the throughput of the entire network. The most straightforward method is to maximize the amount of flows transmitted in the network as long as they obey the constraints defined above. Furthermore, we also add weight factors ω to each of flows according to their priorities in order to meet their QoS requirements.

$$\textbf{Max}: \sum \omega_P \times \alpha_A^{f(ID:i)} f_P^{ID:i}$$
$$\textbf{Subject to:}$$
$$\{f_{A1}^{\rho 1}, f_{A2}^{\rho 2}, ... f_{An}^{\rho n}\} = \Delta_v \times TD_v \qquad \forall v \in V$$
$$\sum_{\rho \in f(e)} f^{\rho} \leq c(e) \quad \forall e \in E$$
$$\sum_{k \in K} F_{v:out}^{ch:k} = \sum_{k \in K} F_{v:in}^{ch:k}$$
$$\forall v \in V, \{F_{v:out} \cup F_{v:in}\} \not\subset \{f(v, \sim), f(\sim, v)\}$$
$$\frac{f_e^{ch(i)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(i)}}{c_{ch:i}(e')} \leq C_{I(e)} \quad \forall e \in E, i \in K$$

V. Link scheduling for collision prevention

A. Out-of-band transmissions of control/data traffic

Control messages of SDNs play a significant role in the network management since the data plane becomes so dummy and only the control panel issues the transmission rules. To guarantee the transmission quality of the control traffic, we assign a secure path between central controller and each switch of the data plane. Thus we adopt a out-of-band style [13] for the SD-WMN structure, which means that the bandwidth of the control traffic cannot be used for the data traffic. In this way, an control channel is independent of the traffic in the data plane. Comparing with the link scheduling of the data traffic, the control traffic forwarding system has a relatively stationary architecture. As long as the control traffic is transmitted efficiently, the path between controller and each switch does not need to change. This invariance is also applied to the channel assignment on the links of these paths. Note that only a single path is assigned to the link between a SDN controller and a switch. In other words, a switch exchanges data with a central controller through a stationary and independent path.

Except for the resources occupied by the control traffic, all the other available channels are dynamically allocated for data plane traffic. The following section explains the scheduling strategy with respect to the channels for data traffic.

B. Single Channel scheduling

Our scheduling scheme aims to reduce the occurrences of traffic transmission bottlenecks by periodically re-allocating the radio resources in the entire network. Assume that the central controller regularly manages the entire network in every period T. That is, in each T, a central controller arranges the transmission of wireless links in multiple super-frames. Each super-frame contains a series of time slots, and the packets are forwarded in the links with the pre-scheduled channels during each time slot without interference with each other.

Let $FD(e_i)$ denote the data traffic on edge e_i including both types of flows, i.e., generated or relayed by the sender of edge e_i. Since the traffic assignment Δ can be performed according to traffic demand TD, it is straightforward to obtain $FD(e_i)$ over the entire WMN. The set of flows in node v upon edge e_i is denoted by $PRI(e_i)$. Since both QoS and fairness of data forwarding have to be considered in link scheduling policy, we introduce a weight factor $\omega_f(\tau)$ to measure the comprehensive urgency level of forwarding a flow in time slot τ. Specifically, the weight factor $\omega_f$is related to both the priority of flows and waiting time γ_f in the queue of a node. For each flow, we have:

$$\omega_f = pri_f + k \times \gamma_f \quad pri_f \in PRI \tag{8}$$

Where, pri_f is the priority of flow f, and k is a constant coefficient to measure how much the waiting time γ_f contributes to the urgency of the flow. Let X_τ denote the expected transmission rate of the flows scheduled in time slot τ on edge e_i, and $W_{X_\tau(e_i)}$ represents the set of corresponding weight factors ω_f for the flows in X_τ. Thus, the objective of the link scheduling in each time slot τ can be formulated as follows:

$$Maximize: \sum_{e_i \in E} W_{X_\tau(e_i)} \cdot X_\tau(e_i) \tag{9}$$

In order to reflect the dynamic channel conditions in link scheduling process, we add a constraint with respect to the link quality. We assume that the channel conditions can be predicted and collected by a central controller. To simplify the model of channel conditions in link scheduling, we employ the signal-to-noise ratio (SNR) as the metric. Assume a vector $G = \{g(e_1), g(e_2), ..., g(e_i), ...\}$ represents the gain of each link. The data transmission power is denoted by a vector $P = \{p(e_1), p(e_2), ..., p(e_i), ...\}$. Then, the SNR can be calculated as:

$$SNR_v(P(e_i)) = \frac{g_v(e_i) p_v(e_i)}{n_v} \tag{10}$$

in which n_v stands for both background noise and the signal strength from other nodes at v. Let the vector $R = \{r^1, r^2, ..., r^m, ..., r^M\}$ denote a series of transmission rates. If node v expects to achieve a transmission rate r^m through link e_i ($X_\tau(e_i) = r^m$) for any wireless modulation scheme, the $SNR_v(P(e_i))$ must be higher than a certain level, such as $SNR_v(P(e_i)) \geq H(r^m)$, where H is a function to calculate the minimum required SNR in terms of transmission rate r^m. Then, we can deduce the minimum power $p_{\min}(e_i)$ to achieve the transmission rate r^m while keeping the packet loss rate (PLR) in a low level.

$$p_{\min}(e_i) = SNR^{-1}(H(r^m)) = SNR^{-1}(H(X_\tau(e_i))) \tag{11}$$

Then, a constraint (via the linear programming model) in link scheduling is given by:

$$SNR^{-1}(H(X_\tau(e_i))) \leq P_{\max} \quad \forall e_i \in E \tag{12}$$

Here $P_{\max}$ stands for the maximum transmission power threshold. This constraint is used to restrict the flow amount for a low link quality.

Furthermore, all the activated links in time slot τ must satisfy the link interference constraint as well:

$$\frac{f_e^{ch(i)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(i)}}{c_{ch:i}(e')} \leq 1 \quad \forall e \in E_\tau, e' \in E_\tau, e \neq e' \tag{13}$$

Where E_τ is the set of edges activated in time slot τ. This constraint is the sufficient condition of wireless link interference. Hence, the linear programming model of link scheduling in single channel i is as below:

$$\textbf{Max:} \sum_{e_i \in E} W_{X_\tau(e_i)} \cdot X_\tau(e_i)$$
$$\textbf{Subject to:}$$
$$SNR^{-1}(H(X_\tau(e_i))) \leq P_{\max} \quad \forall e_i \in E$$
$$\frac{f_e^{ch(i)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(i)}}{c_{ch:i}(e')} \leq 1 \quad \forall e \in E_\tau, e' \in E_\tau, e \neq e'$$

Algorithm 1 shows the data forwarding scheduling policy for a single channel case.

Algorithm 1 Single channel scheduling

1: Input: Traffic Demand of edge e_i with Priority label: $FD(e_i)$
2: Initialize: Schedule $S_T(e_i^\tau)$, Time slot $\tau = 0$,
3: **while** $T \geq \tau$ **do**
4: Solve the Linear programming problem of link scheduling during time slot τ on a single channel i
5: Set the $S_T(e_i^\tau)$ with the link scheduling results
6: Upgrade $FD(e_i)$
7: $\tau = \tau + \tau'$
8: **end while**

C. Multi-channel scheduling

Algorithm 1 can be easily extended to multi-channel case. First of all, the link scheduling policy must fully utilize all the channels. Given K available channels in the transmissions, we have:

$$\textbf{Max:} \sum_{\sigma=1}^{K} \sum_{e_i \in E} W_{X_\tau(e_i)}^{ch(\sigma)} \cdot X_\tau^{ch(\sigma)}(e_i) \tag{14}$$

Regarding the transmission power constraint, each of the active links (no matter which channel is used) in time slot τ must be smaller than the maximum allowable power level. That is,

$$\beta_\tau^{ch(\sigma)}(e_i) X_\tau^{ch(\sigma)}(e_i) \leq P_{\max} \quad \forall e_i \in E, \sigma = 1, ..., K \tag{15}$$

Similarly, the constraint of link interference can be represented as:

$$\frac{f_e^{ch(\sigma)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(\sigma)}}{c_{ch:i}(e')} \leq 1$$
$$\forall e \in E_\tau, e' \in E_\tau, e \neq e', \sigma = 1, ..., K \tag{16}$$

The edge scheduling strategy can vary based on the features of the physical layer. Particularly, OFDMA cannot receive and send data through different channels at the same time, while MR-MC and MBDAs are able to receive or send data simultaneously via different channels (or beams). We simplify the data plane operations into two categories: first, in each time slot, the node only operates in receiving or sending mode, which is suitable for OFDMA-like techniques. Second, a node is capable of receiving and sending data at the same time (due to MR-MC features). In the case of MBDAs, it increases the throughput via multiple directional beams for space reuse purpose. Assuming the link interference with various physical layer techniques (e.g. MBDAs, MC-MR, OFDMA, ETC.) can be reflected in the link interference matrix $I(e_i)$, we propose to use *Algorithm 2* to achieve the link scheduling in multi-channel circumstances.

Algorithm 2 Multi-channel scheduling

1: Input: Traffic Demand of edge e_i with Priority label: $FD(e_i)$ $\forall e_i \in E$
2: Initialize: Schedule $S_T(e_i^\tau)$, Time slot $\tau = 0$, Available channels K_{e_i} of edge e_i ($e_i \in E$)
3: **while** $T \geq \tau$ **do**
4: Calculate the multi-channel link scheduling problem by using equations(14)(15)(16)
5: Traversal entire network, find the traffic demand $FD(e_i)$ with the highest priority and corresponding edge e. If the edges (channels) of the traffic demand $FD(e_i)$ are all occupied, move to next top-priority traffic demand.
6: **while** K_{e_i} is not empty **do**
7: Assign the Traffic Demand $FD(e_i)$ to available channels (High priority first)
8: Set the $S_T(e_i^\tau)$ with the link scheduling results
9: Upgrade $FD(e_i)$
10: **end while**
11: $\tau = \tau + \tau'$
12: **end while**

VI. Training Model for Link Failure Prediction

Once a link failure occurs, a SDN router sends a request message to a SDN controller to update the flow tables. This is because the SDN model assumes that the dummy routers are not capable of solving an unexpected traffic collision issues by themselves. When a controller receivers a request from the data plane, it calculates the solution from a global viewpoint and sends new flow table rules to the relevant routers.

When a controller intends to reroute a data flow caused by an access collision or link failure, it will update the flow tables in the routers/switches belonging to the path of the data flow. A controller can thus indirectly forward control messages to all the routers on the path via the flow table rule changes. Note that all the flow tables of routers belonging to the same path have to be updated simultaneously to prevent inconsistency among the flow table rules.

However, a SD-WMN can still suffer from unexpected link failures because a SDN-based control may be not able to

handle the link failure problem fast enough due to the flow table update delay. Hence, we propose a link failure prediction (LFP) strategy, in the sense that a back-up routing solution can be made by the controller according to the LFP result, and the back-up link IDs can be saved in the relevant routers in advance. In this approach, a router is able to immediately respond to a link failure event. Thus the waiting time of receiving commands from the controller, could be significantly reduced.

In WMNs, a number of issues could cause packet loss or data corruptions, such as signal blocking from a metal object, link noise, node capture issue from directional antennas, traffic congestion, node mobility, etc. A WMN may be able to tolerate packet loss temporarily, but a permanent link failure has to be fixed as soon as possible. The IEEE 802.11 DCF assures a link failure when the RTS-CTS handshake cannot be established after a sender sends a neighbor RTS attempts for 7 times [2]. Nevertheless, such a simple link failure detection has two main drawbacks. First, a link failure decision highly depends on the frequency of RTS attempts. Even a receiver is unavailable in terms of one of its neighboring senders in a short time due to some reasons mentioned above, a false link failure may still happen due to the failure of a certain number of RTS attempts. Second, there would be certain response delay when a router starts to make decisions after 7 times of RTS attempts. Hence, a link failure prediction is necessary in a WMN, especially under the node mobility.

At present, some works have studied the link failure detection issues in WMNs. Most of them mainly concentrate on the node mobility problem. A number of metrics and mechanisms are used to deal with the problem, including packet reception ratio (PRR), ETX [7], Receiving Signal Strength Indicator (RSSI) [28], SNR [1], etc.

Since SDN assumes that the routers in data plane have a dummy structure, the link failure prediction (LFP) strategy in a SDN-based wireless router cannot adopt a complicated strategy. And regarding the feasibility and performance of a LFP in SD-WMN, three aspects have to be considered:

- *Simplicity*: The LFP strategy cannot use complex algorithms in the data plane.
- *Accuracy*: The LFP has to be accurate. Otherwise the control plane will generate much unnecessary control traffic due to incorrect routing control.
- *Timeliness*: A link failure must be confirmed in a fast manner to leave enough time for back-up routing establishment before the link failure happens.

In traditional WMNs, the link failure detection is part of the routing discovery process, such as link-state routing protocol [22]. The routing protocol actively broadcasts HELLO messages to neighboring nodes from time to time. The neighboring nodes can then establish the connectivity with each other after receiving these HELLO messages. On the contrary, if a node cannot receive the HELLO messages, then the link between them is determined as being broken. In WMNs the packet loss could happen frequently because many factors are able to significantly interfere with a wireless transmission. There are some drawbacks in the link-state routing protocol in WMNs. First, the active neighboring detection causes much overhead. Second, a detection delay can be very long since the link failure detection highly depends on the forwarding frequency of HELLO messages.

Unlike the link-state routing protocol, some cross-layer detection schemes are proposed [15][20][26][15]. Specifically, these approaches utilize the acknowledgement in MAC layer to detect the failure. The overhead caused by HELLO messages thus can be waived due to the passive link failure detection. Nevertheless, these approaches require cross-layer design, where a link failure in MAC-layer will have impacts on the behaviors in the routing layer. The inconsistency between the routing and MAC in terms of link detection results may also decrease the transmission reliability of the network. Moreover, using acknowledgement alone in MAC layer is inappropriate in certain cases, since some other factors may cause the consecutive loss of ACKs. In [20][26][15] they assume a fixed data rate in physical layer, which is not applicable to general WMNs. In [21] it gives a data mining based LFP scheme. Nevertheless, it is merely based on traditional distributed WMN system (thus involving much control overhead). And it does not consider node malfunction/mobility.

The proposed LFP particularly utilizes the advantages of SD-WMNs, and is based on the packet loss or corruption statistics during data transmissions. A data transmission error can be categorized into two types: transient errors or permanent link failure. A transient error stands for a temporary fluctuation of the link quality. It does not need the change of the routing rules in switches. On the contrary, a permanent failure means that the link does not work anymore. As a result, this issue requires an update of flow tables to reroute the corresponding data flows. Such an update has to be operated by a SDN controller. The objective of the proposed LFP is to rule out the cases of transient errors and detect the permanent link failure in advance. This process relies on the observation that a permanent link failure can cause transmission errors with a particular pattern. In order to improve the accuracy and reliability of the prediction, a two-layer LFP structure is proposed in this work.

A. Link quality measurement

The SNR is employed as the measure of link quality. According to 802.11 standard [14], the measurements of receiving signal strength (RSSI) and background noise are typically observed by the firmware and drivers. Regarding the monitoring period T_m, the setup of its value is a trade-off between the prediction accuracy and delay. A longer SNR monitoring period could collect more data and thus improve the accuracy of the prediction. However, the node can take more time to generate the prediction result.

The measurement of SNR fluctuates, and hardly reflects the distance between the sender and receiver due to the noise from the air. To overcome this issue, Kalman Filter [30] can be used to smooth out the fluctuations of the measured SNR. The goal of Kalman Filter is to obtain a relatively precise state through a number of inaccurate measurements observed over noises. Let x_k stand for the state in time k, we have:

$$x_k = Ax_{k-1} + Buk - 1 + w_{k-1} \tag{17}$$

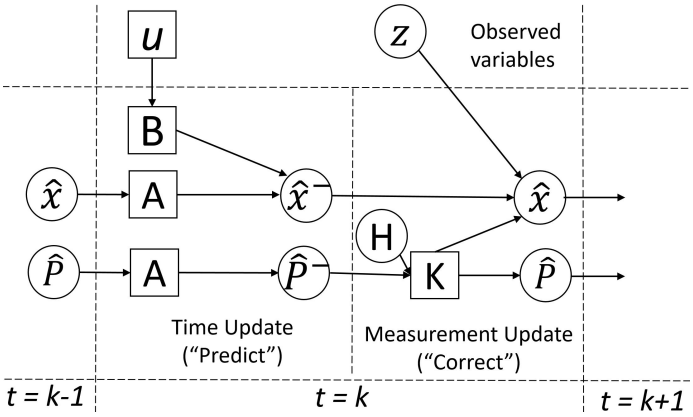


Fig. 4: Model structure of Kalman Filter

Then, the observed value Z_k could be obtained by the following equation:

$$z_k = Hx_k + v_k \tag{18}$$

In the above equations, w_k and v_k stand for two independent noises. A is a $n*n$ matrix containing the correlation information between consecutive data points. The control term u_k is determined by the matrix B. While H is the coefficient matrix to define the relations between the observed value z_k and the state x_k.

Since the state x_k cannot be directly observed, Kalman filter evaluates the state x_k by using a priori state estimate, observed value and posteriori state estimate. The iteration of state estimation of Kalman filter includes two steps (Figure 4). First, given state estimate $\hat{x}_{k-1}$ and priori estimate error covariance P_{k-1}^- as the inputs of the system, priori state estimate $\hat{x}_k$ and priori estimate error co-variance P_k^- at time k can be obtained as:

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \tag{19}$$

$$p_k^- = AP_{k-1}A + Q \tag{20}$$

In Step 2, the matrix K, a gain factor, can be obtained as:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{21}$$

Meanwhile, the state estimation $\hat{x}_k$ and estimate error covariance P_k^- are updated by the following equations:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) \tag{22}$$

$$p_k = (I - K_k H)P_k^- \tag{23}$$

Figure 5 shows the result obtained via Kalman Filter. The blue spots denote the observed SNRs, while the red curve shows the filtered SNRs via Kalman Filter. Due to the noise the filtered SNR curve is still oscillating to a certain extent. However, the filtered SNR has a linear relationship with sender-receiver distance (see the green dash line in Figure 5). Thus, according to the distance (that is related to SNR), we can detect the moving tendency of a node. It is further feasible that we can predict the link failure caused by node mobility.

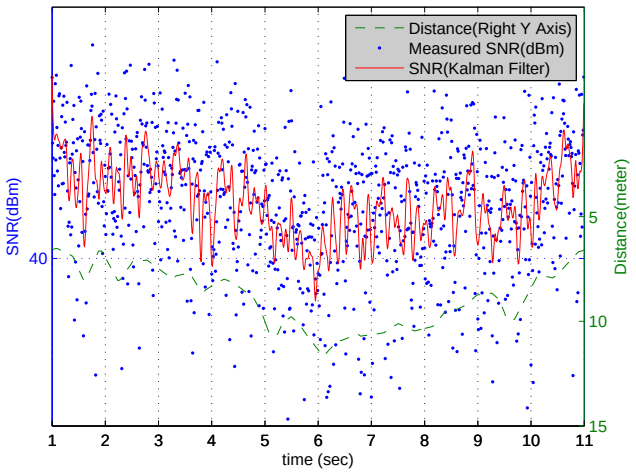


Fig. 5: Filtered SNR (Kalman Filter)

B. *Prediction process in the* data plane

After the noise is filtered from the observed SNR via Kalman filter, the next step is to extract the observed SNRs in the link history and make a comparison between the observed data and the training data in each step. If the pattern of observed data is similar to the training data labeled as link failure, a failure will likely happen in this link. In order to complete this task, a normalized cross-correlation [19] is employed to find the pattern similarity between the observed and training data. Assuming the length of observed SNR is l and the delay is d, the normalized cross-correlation $\gamma(d)$ can be computed as:

$$\gamma(d) = \frac{\sum_{i=1}^{l}[q(i) - \overline{q}][t_{a,b}(d+i) - \overline{t}_{a,b}]}{\sqrt{\sum_{i=1}^{l}[q(i) - \overline{q}]^2 \sum_{i=1}^{l}[t_{a,b}(d+i) - \overline{t}_{a,b}]^2}} \tag{24}$$

The idea in this equation is straightforward: we can predict the link quality by calculating the normalized cross-correlation between the observed SNR and training data [9]. Specifically, $q(i)$ stands for the observed SNR value, and $t_{a,b}(d+i)$ represents the corresponding SNR in the training data for the link between the sender and receiver.

C. *Link failure prediction in the* control plane

The above is low-layer (in data plane) prediction, which only considers local node mobility. To improve the accuracy and exploit the global control of the SDNs, the control plane possesses a link prediction model as well. As long as a node mobility is detected in the data plane, a warning message will be sent and received by the SDN controller. Then the controller will pull out the SNR information from the neighbors of the suspect node to confirm its mobility status. Unlike the LFP in the data plane, the controller has multiple sets of SNR data and can thus make an accurate, comprehensive decision. Considering that this is a classification problem with multiple high-dimensional inputs, the Support Vector Machine (SVM) method [11] is employed. SVM is a supervised learning based

classification method that has been used in many pattern recognition applications. In the following discussions the procedure of the SVM-based LFP training in the controller is described.

Given a set of training data points $\top = \{\mathbf{x}_i, y_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbf{R}^N$ represents the i-th input feature vector, and $y_i \in \{-1, 1\}$ is the vector label of $\mathbf{x}_i$. The SVM aims to classify the set of observed data into two classes via the optimal hyperplane $\boldsymbol{\omega}\phi(\mathbf{x}) + b = 0$, where $\boldsymbol{\omega}$ is a weight vector, b is a bias and $\phi(\mathbf{x})$ stands for the mapping from input $\mathbf{x}_i$ space to a high-dimensional space. In order to find the maximum margin hyperplane to separate the data points with positive and negative labels, we have to decide the weight vector $\boldsymbol{\omega}$ and bias b. This problem can be solved by minimizing the following function:

$$R = \frac{1}{2}\left\|\omega\right\|^2 + C\frac{1}{len}\sum_{i}^{len}\left|y_i - f(x_i)\right|_{\varepsilon} \tag{25}$$

Here the norm of $\boldsymbol{\omega}$, denoted as $\|\omega\|$, is used to constrain the model complexity for better efficiency. C is a factor for the trade-off between training errors and flatness. *Equation* (25) can be transformed to *Equation* (26) by setting ε-intensive zone and further introducing the positive slack variables ξ_i and ξ_i^*:

Minimize:

$$R = \frac{1}{2}\left\|\omega\right\|^2 + C\frac{1}{len}\sum_{i}^{len}\left|\xi_i - \xi_i^*\right| \tag{26}$$

Subject to:

$$\omega\phi(x_i) + b - y_i \le \varepsilon + \xi_i^* \qquad i = 1, 2, ..., len$$
$$y_i - \omega\phi(x_i) - b \le \varepsilon + \xi_i \qquad i = 1, 2, ..., len$$
$$\xi_i, \xi_i^* \ge 0 \qquad i = 1, 2, ..., len$$

Here ξ_i and ξ_i^* are respectively the upper and lower training errors from the ε-intensive tube. After introducing Lagrange multipliers α_i and kernel function k, *Equation* (26) can be expressed as:

$$f(x, \alpha_i, \alpha_i^*) = \sum_{i=1}^{len}(\alpha_i - \alpha_i^*)K(x, x_i) + b \tag{27}$$

Then, the linear programming problem can be transformed to a dual QP problem that can be solved by using the Gaussian radial basis function kernel (see *equation* (28) below).

$$K(x_i, x_j) = exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \tag{28}$$

Thus, we have:

Maximize:

$$R(\alpha_i, \alpha_i^*) = -\frac{1}{2}\sum_{i,j=1}^{len}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j)$$
$$-\varepsilon\sum_{i=1}^{len}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{len}y_i(\alpha_i - \alpha_i^*)$$

Subject to:

$$\sum_{i=1}^{len}(\alpha_i - \alpha_i^*) = 0$$
$$0 \le \alpha_i \le C, i = 1, 2, ..., len$$
$$0 \le \alpha_i^* \le C, i = 1, 2, ..., len$$

In the above QP problem, the constraint can be adjusted by changing the factors C and ε. As a necessary and sufficient condition, Karush-Kuhn-Tucker (KKT) theorem is applied to the QP problem [18]. As a result, we can get rid of most of the coefficients when $(\alpha_i - \alpha_i^*)$ has to differ from zero in equation 27. In this way, support vector can be determined in the corresponding training data [4]. Meanwhile, the bias factor b can be determined as well.

After the SVM module finishes training, the controller will trigger the SVM classification algorithm once the link failure warning message is received from the data plane. Specifically, the controller will pull out the very recent SNR data related to the target node with failed links. This information can be either directly collected from the target node or its neighbors. Since the locations of the neighbors are distinct, the SVM of central controller aims to detect the moving status of the target node. More concisely, the SVM is used to confirm if the target node is keeping moving away from this area in a fixed direction. Besides the SNR data from the node that sent the warning message, we can collect more sets of SNR data from different neighbors of the target node.

As long as a link failure is confirmed, the controller will immediately assign a new back-up path (to be discussed in the next section) to the affected nodes in order to bypass the moving-away node. On the other hand, there is a chance that the controller cannot pull out enough sets of SNR data due to the random network topology and traffic. As a result, the controller will directly send a back-up route to the affected nodes. Then, the affected nodes will launch the back-up paths when they confirm the link failure by using the traditional method such as IEEE 802.11.

VII. Back-up route determination after a link failure

The proposed traffic management (section IV) and link scheduling (section V) methods are based on periodic network behavior control, and the central controller calculates the traffic allocation and scheduling in each cycle. By following the same periodic control style, a back-up route determination will be made to handle a predicted link failure.

The data forwarding on each link is scheduled by the SDN controller in each control cycle. But a link failure can also

trigger a set of other control behaviors, such as link re-scheduling and flow re-routing. According to the global view of the central controller, the alternative paths for the link failure can be easily obtained. Thus, the back-up route seeking is actually a task of finding the best path among the alternative paths. Basically, the best alternative path has to satisfy the following conditions: (1) the shortest path distance; (2) the lowest control overhead; and (3) the least impact on other regularly scheduled data traffic.

Due to the link error accumulations in multi-hop transmissions (i.e., each time the data passes through one more hop, the probability of losing packets gets higher), the distance (the number of hops) of an alternative path has to be as small as possible. The SDN controller has to send the notification message to the nodes on the alternative path, when the information of the rerouted data flow is not cached in the flow tables of those nodes. In other words, when a SDN router receives a new type of data flow that it did not handle before, it has to request the corresponding forwarding rules in its flow tables, and such rules should be sent from the SDN controller. This will generate certain control traffic. Hence, an entirely new alternative path (i.e., without any overlap links with the previous path) will incur more control overhead. Similarly, an alternative path also tends to affect other originally scheduled data flows.

Figure 6 gives an example of alternative paths. Assuming path A is the original path with an unexpected link failure, there are two alternative paths available (path B and path C). Path B merely needs to update the flow tables for two nodes, and most nodes of the original path A remain in path B. On the contrary, path C needs to update the flow tables of all its nodes, which will generate more control overhead than path B. In addition, path C has higher probability of affecting other existing routing paths as well due to its larger scope of link update.

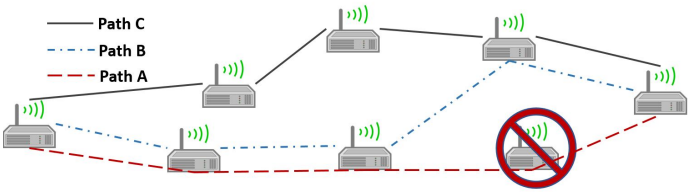


Fig. 6: Alternative paths for link failure

Therefore, it is preferable to keep the relay nodes of the original path untouched (as many nodes as possible) in the alternative path. Based on the linear programming equations in Section IV-D, it is necessary to meet several necessary conditions for the alternative path. Given ρ_{alt} and f_{rrt} as an alternative path and rerouted flow, respectively, we have:

$$\sum_{\rho \in f(e)} f^{\rho} + f_{rrt} \leq c(e) \quad \forall e \in \rho_{alt}$$

$$\sum_{k \in K} F_{v:out}^{ch:k} = \sum_{k \in K} F_{v:in}^{ch:k}$$

$$\forall v \in \rho_{alt}, \{F_{v:out} \cup F_{v:in}\} \not\subset \{f(v,\sim), f(\sim,v)\}$$

$$\frac{f_e^{ch(i)}}{c_{ch:i}(e)} + \sum_{e' \in I(e)} \frac{f_{e'}^{ch(i)}}{c_{ch:i}(e')} \leq C_{I(e)} \quad \forall e \in \rho_{alt}, i \in K$$

The above three equations describe the features for the edges and nodes of the alternative path when dealing with a predicted link failure. They denote the constraints for available capacity of each edge, node throughput consistency, and radio frequency collision avoidance, respectively.

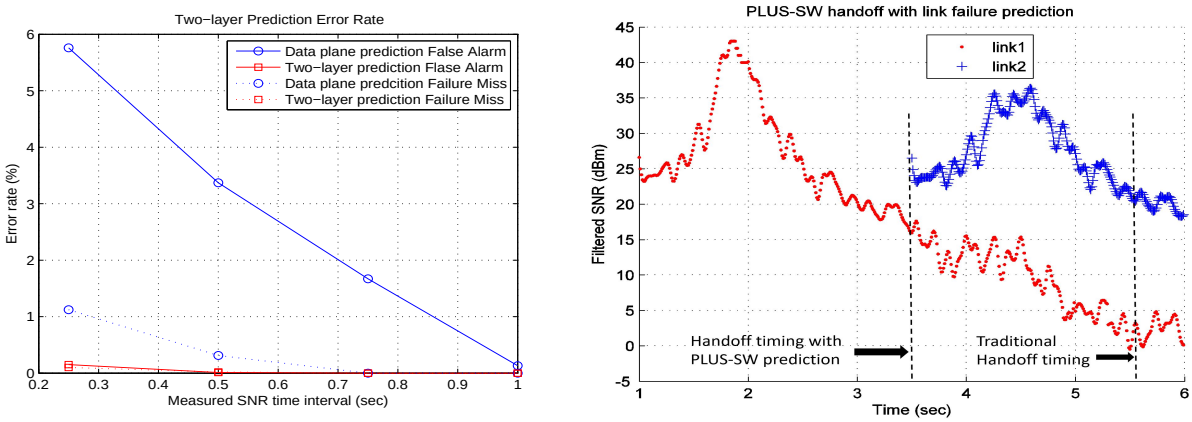
The back-up routing calculation is processed in the central controller. This process is triggered by the proposed two-layer LFP model. Specifically, nodes of data plane keep sensing the mobility status of their neighbors. Once a suspect node is detected, it will send a warning message to the controller, which will double check the mobility status of this suspect node via SVM training model from a global perspective. As long as the link failure is confirmed by the controller, it immediately seeks the alternative path and sends the necessary re-routing commands to the relevant nodes. Based on the traffic engineering model described in section IV, the alternative route is determined by three necessary conditions for the purpose of minimizing the network overhead and impacted network area caused by the link failure.

VIII. Performance Validations

The section presents the performance evaluation for the proposed traffic engineering strategies in a SD-WMN. Specifically, we have implemented the proposed centralized traffic engineering, the two-layer LFP model and the SDN-based WMN structure, in a ns-3 based simulation system.

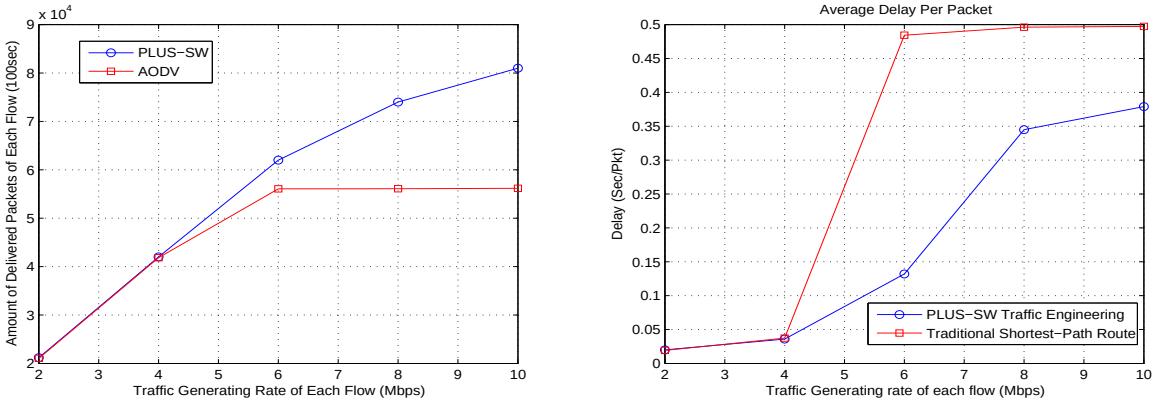
The network contains 50 nodes deployed in an $300m \times 1500m$ area. Regarding the transceiver capacity, we adopt the $DsssRate11Mbps$ model from ns-3. This means that the physical bandwidth of each node is $11Mbps$. The transmitter's power is set to $7.5dBm$. All the wireless nodes in the data plane is managed by a SDN controller. The packet size is set to 1200 bytes.

Figure 7 demonstrates the performance of the proposed two-layer LFP strategy. In Figure 7a, the prediction error rate is displayed in terms of false alarm rate and failure detection missing rate. The X-axis is the measurement time interval, while the Y-axis is the error rate (in percentage). As shown in Figure 7a, while the individual data plane prediction can keep the error rates in a relative low level, the two-layer prediction scheme significantly improves the accuracy of failure prediction. Meanwhile, a longer SNR measurement time would also lower the prediction error rate. Figure 7b illustrates the benefits of using the proposed two-layer prediction, i.e., the PLUS-SW enables a timely link hand-off control (i.e., switching the link or the channel in use) once a link failure is predicted. As a result, the network actively avoids the packet loss caused by



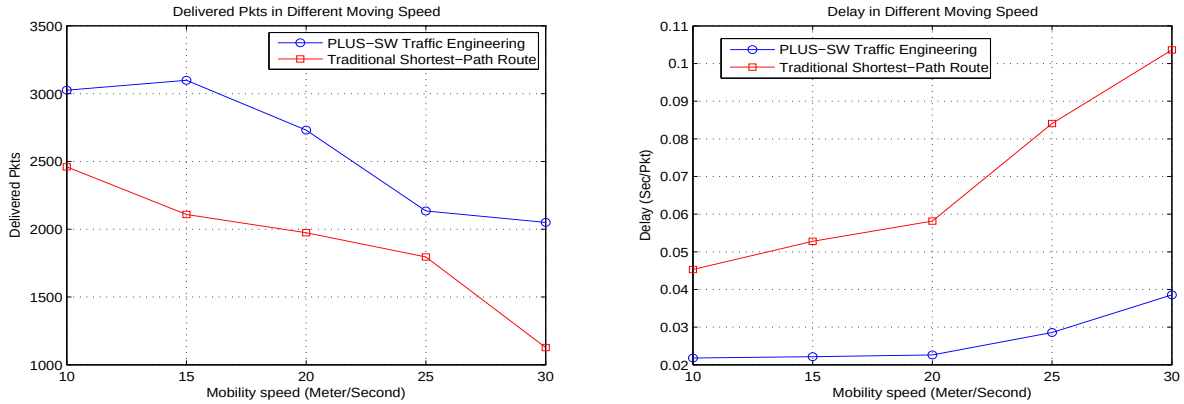
(a) Prediction error rate (b) Handoff time with PLUS-SW failure prediction

Fig. 7: Performance evaluation of two-layer prediction



(a) Amount of delivered packets (b) Delay performance

Fig. 8: Performance comparison between PLUS-SW and conventional routing



(a) Amount of delivered packets (b) Delay per packet

Fig. 9: Performance comparisons under different moving speeds

low SNR in data transmissions. Based on our experiment, the proposed two-layer LFP is able to precisely predict the link failure events for the general mobility cases, such as human walking, vehicle moving, etc.

To further estimate the performance of the proposed prediction model, we will consider multiple data flows with different traffic generating rates and mobility speeds. As such, the two-layer LFP can be comprehensively assessed in the entire SD-WMN. In this experiment, we aim to compare the performance of the proposed PLUS-SW with the traditional distributed network control. The throughput and delay are collected under different traffic generating rates (from $2Mbps$ to $10Mbps$ for each flow). we create 4 different data flows in the network (unlike the previous experiment with only one data flow).

Figure 8 illustrates the performance of the network with different traffic generating rates. As shown in Figure 8a, the proposed PLUS-SW traffic management outperforms the tradition routing protocol based on the shortest-path strategy (such as AODV). Since the traffic flows are scheduled via a centralized control, PLUS-SW achieves better traffic balancing level than the traditional distributed routing protocols. On the other hand, the delay of PLUS-SW is also lower than AODV (see Figure 8b). This indicates that the data traffic transmission collisions can be effectively prevented through PLUS-SW.

Since we employ the LFP strategy to cope with mobility issue in the WMN environment, the performance evaluation for different mobility speeds is also conducted in ns-3 simulator. As in the previous experiments, the simulation contains 50 nodes with total 4 data flows. The traffic generating rate of each data flow is $2Mbps$.

As shown in the Figure 9, the proposed PLUS-SW has a better performance under mobility. For the same traffic generating rates, the PLUS-SW achieves a higher data delivery rate than the traditional routing scheme, under different mobility speeds (see Figure 9a). And the delay of PLUS-SW is lower (Figure 9b). Thus, the proposed LFP strategy is efficient in mobility environment.

The PLUS-SW supports the QoS by scheduling the data flows with different priorities. As one of the widely used metrics, jitter plays an important role for real-time multimedia transmissions. Figure 10 illustrates the jitter performance under various traffic generating rates and moving speeds of the nodes. As in the previous experiments, the traffic generating rates change from 2 Mbps to 10 Mbps for each data flow. As shown in Figure 10a, the jitter of the proposed PLUS-SW traffic engineering model has the lower jitter than the tradition routing scheme. On the other hand, the faster moving speed can cause higher jitter, since the network has to discover a new path when the path is broken due to mobility. Nevertheless, the proposed PLUS-SW still exceeds the performance of conventional routing with respect to jitter.

To further evaluate the performance of PLUS-SW, we test the packet delivery rate (PDR), end-to-end delay, and the jitter of data flows with different number of hops. In this experiment, we create multiple data flows with different path lengths (i.e., number of hops). As in the previous simulations, 4 data flows are used in the entire network.

As shown in the Figure 11, more hops can cause higher packet loss rate. This is due to traffic conflicts, radio interference, etc. Through the proposed network-wide management in PLUS-SW, the packet delivery rate is higher than the traditional routing protocol. At the same time, the delay of PLUS-SW is smaller. This is because the optimal path selection through a global control tends to manage the traffic allocations with less conflicts (Figure 11b). In Figure 11c, the jitter of PLUS-SW is lower as well. The repeatedly forwarding times (shown in Figure 11) matches with the result of Figure 10, since the less repeated forwarding times in PLUS-SW protocol can lower the jitter in each data flow's transmission.

IX. Conclusions

In this paper, we have presented an innovative SDN-based WMN structure. This paradigm can fully explore the centralized SDN control features and overcome some challenging issues in SD-WMN such as the difficulty to handle mobility. First, we utilize the central controller to implement a globally optimized traffic engineering. Second, two supervised learning models are introduced to cope with the permanent link failure issues caused by WMN mobility. Third, a back-up route seeking scheme is proposed based on a new traffic management model under link failure. The entire system is validated on the ns-3 simulator, where the performance of the SD-WMN is improved in both static and mobility network environments.

Because SDN offers a great opportunity for intelligent network control, in the future work we will use new intelligent models (for example, based on machine learning algorithms) for SD-WMN routing/congestion control. Meanwhile, the overhead of the control traffic will be reduced through low-complexity SDN control schemes in our future research.

References

[1] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11 b mesh network. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 121–132. ACM, 2004.

[2] Giuseppe Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, 18(3):535–547, 2000.

[3] Gurashish Brar, Douglas M Blough, and Paolo Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 2–13. ACM, 2006.

[4] Lijuan Cao. Support vector machines experts for time series forecasting. *Neurocomputing*, 51:321–339, 2003.

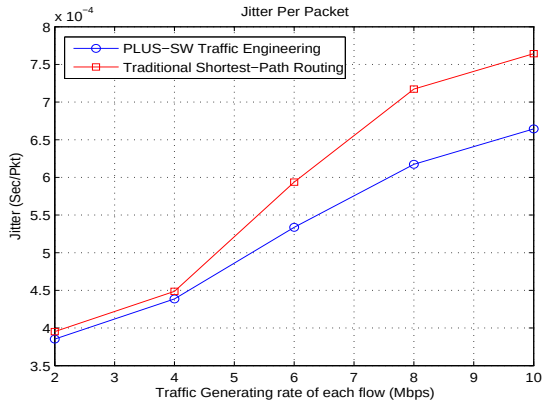
[5] Alberto Cerpa, Jennifer L Wong, Miodrag Potkonjak, and Deborah Estrin. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 414–425. ACM, 2005.

[6] Salvatore Costanzo, Laura Galluccio, Giacomo Morabito, and Sergio Palazzo. Software defined wireless networks: Unbridling sdns. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, pages 1–6. IEEE, 2012.

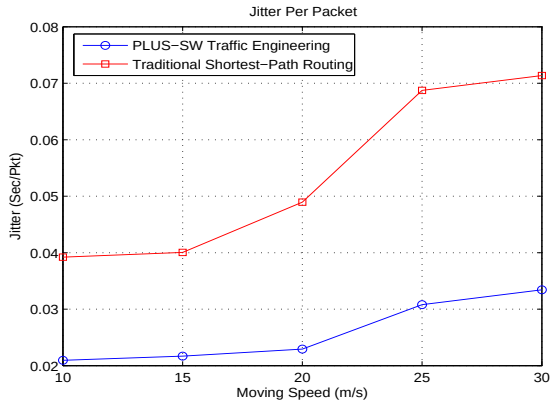
[7] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, 2005.

[8] Andrea Detti, Claudio Pisa, Stefano Salsano, and Nicola Blefari-Melazzi. Wireless mesh software defined networks (wmsdn). In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*, pages 89–95. IEEE, 2013.

[9] Károly Farkas, Theus Hossmann, Franck Legendre, Bernhard Plattner, and Sajal K Das. Link quality prediction in mesh networks. *Computer Communications*, 31(8):1497–1512, 2008.

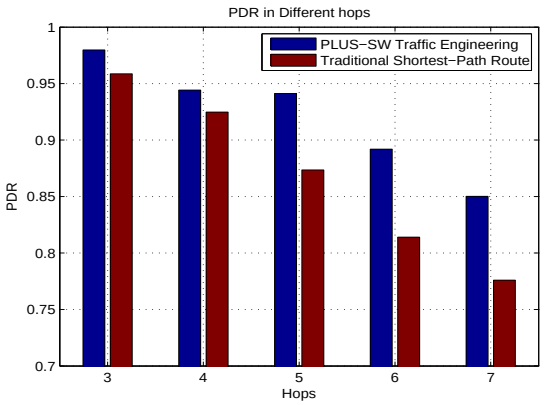


(a) Jitter for different traffic generating rates

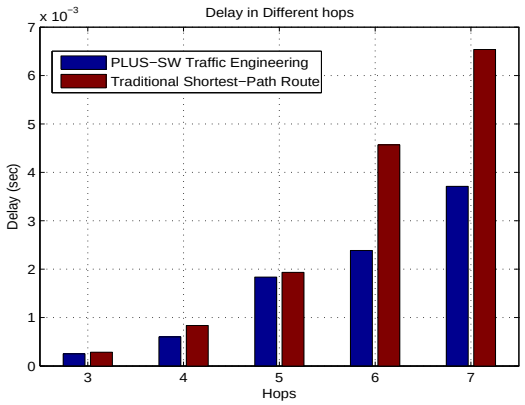


(b) Jitter for different moving speeds

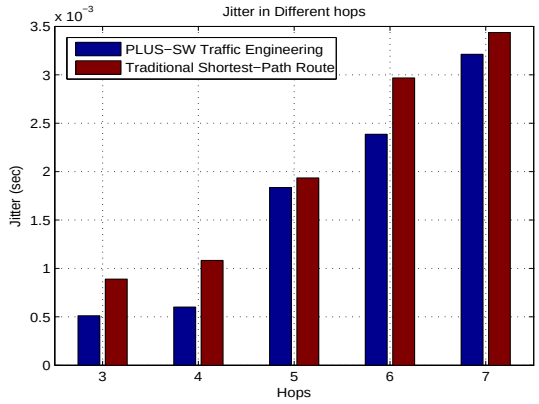
Fig. 10: Jitter Comparison



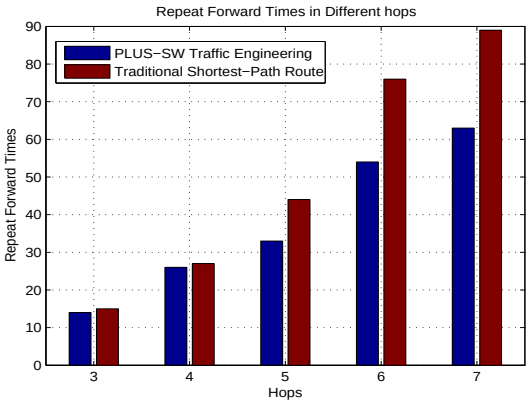
(a) PDR for different number of hops



(b) Delay for different number of hops



(c) Jitter for different number of hops



(d) Repeatedly forward times for different number of hops

Fig. 11: Performance for different number of hops

[10] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. Four-bit wireless link estimation. In *HotNets*, 2007.

[11] Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14:85–86, 1998.

[12] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.

[13] Huawei Huang, Peng Li, Song Guo, and Weihua Zhuang. Software-defined wireless mesh networks: architecture and traffic orchestration. *IEEE Network*, 29(4):24–30, 2015.

[14] IEEE. Ieee 802.11 standard, 1999.

[15] Yu-Chih Jen. Method for detecting radio link failure in wireless communications system and related apparatus, September 20 2007. US Patent App. 11/902,303.

[16] Golnaz Karbaschi and Anne Fladenmuller. A link-quality and congestion-aware cross layer metric for multi-hop wireless routing. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 7–pp. IEEE, 2005.

[17] Keith Kirkpatrick. Software-defined networking. *Communications of the ACM*, 56(9):16–19, 2013.

[18] Pavel Laskov. An improved decomposition algorithm for regression support vector machines. In *NIPS*, pages 484–490, 1999.

[19] JP Lewis. Fast normalized cross-correlation. In *Vision interface*, volume 10, pages 120–123, 1995.

[20] Qing Li, Cong Liu, and Han-hong Jiang. The routing protocol of aodv based on link failure prediction. In *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pages 1993–1996. IEEE, 2008.

[21] Timo Lindhorst, Georg Lukas, Edgar Nett, and Michael Mock. Data-mining-based link failure detection for wireless mesh networks. In *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, pages 353–357. IEEE, 2010.

[22] Vincent D Park and M Scott Corson. A performance comparison of the temporally-ordered routing algorithm and ideal link-state routing. In *Computers and Communications, 1998. ISCC'98. Proceedings. Third IEEE Symposium on*, pages 592–598. IEEE, 1998.

[23] Krishna N Ramachandran, Elizabeth M Belding-Royer, Kevin C Almeroth, and Milind M Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Infocom*, volume 6, pages 1–12, 2006.

[24] Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2223–2234. IEEE, 2005.

[25] Stefano Salsano, Giuseppe Siracusano, Andrea Detti, Claudio Pisa, Pier Luigi Ventre, and Nicola Blefari-Melazzi. Controller selection in a wireless mesh sdn under network partitioning and merging scenarios. *arXiv preprint arXiv:1406.2470*, 2014.

[26] D Satyanarayana and SV Rao. Link failure prediction qos routing protocol for manet. 2007.

[27] Murat Senel, Krishna Chintalapudi, Dhananjay Lal, Abtin Keshavarzian, and Edward J Coyle. A kalman filter based link quality estimation scheme for wireless sensor networks. In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 875–880. IEEE, 2007.

[28] Masashi Sugano, Tomonori Kawazoe, Yoshikazu Ohta, and Masayuki Murata. Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. *Target*, 538:050, 2006.

[29] Lei Tang, Kuang-Ching Wang, Yong Huang, and Fangming Gu. Channel characterization and link quality assessment of ieee 802.15. 4-compliant radio for factory environments. *IEEE Transactions on industrial informatics*, 3(2):99–110, 2007.

[30] Greg Welch and Gary Bishop. An introduction to the kalman filter. 1995.

[31] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27. ACM, 2003.

[32] Dong Yang, Youzhi Xu, and Mikael Gidlund. Coexistence of ieee802. 15.4 based networks: A survey. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, pages 2107–2113. IEEE, 2010.

[33] Mao Yang, Yong Li, Depeng Jin, Lieguang Zeng, Xin Wu, and Athanasios V Vasilakos. Software-defined and virtualized future mobile and wireless networks: A survey. *Mobile Networks and Applications*, 20(1):4–18, 2015.