Multi-Armed Bandit Congestion Control in Multi-Hop Infrastructure Wireless Mesh Networks

A. B. M. Alim Al Islam, S. M. Iftekharul Alam, Vijay Raghunathan, and Saurabh Bagchi
School of ECE, Purdue University, West Lafayette, IN 47907
Email: {abmalima, alams, vr, sbagchi}@purdue.edu

Abstract*—Congestion control in multi-hop infrastructure wireless mesh networks is both an important and a unique problem. It is unique because it has two prominent causes of failed transmissions which are difficult to tease apart - lossy nature of wireless medium and high extent of congestion around gateways in the network. The concurrent presence of these two causes limits applicability of already available congestion control mechanisms, proposed for wireless networks. Prior mechanisms mainly focus on the former cause, ignoring the latter one. Therefore, we address this issue to design an end-to-end congestion control mechanism for infrastructure wireless mesh networks in this paper. We formulate the congestion control problem and map that to the restless multi-armed bandit problem, a well-known decision problem in the literature. Then, we propose three myopic policies to achieve a near-optimal solution for the mapped problem since no optimal solution is known to this problem. We perform comparative evaluation through `ns-2` simulation and a real testbed experiment with a wireline TCP variant and a wireless TCP protocol. The evaluation reveals that our proposed mechanism can achieve up to 52% increased network throughput and 34% decreased average energy consumption per transmitted bit in comparison to the other end-to-end congestion control variants.

I. Introduction

Multi-hop Wireless Mesh Networks (WMNs) have emerged as a popular communication paradigm with a variety of deployment scenarios. Examples include wireless metropolitan networks [21], automatic meter reading (AMR) [27], and construction site monitoring [34]. Many of these deployments have gateways that connect the mesh nodes to the infrastructure network (such as, the wireline internet). Therefore, we term such networks as infrastructure WMNs in this paper. A number of applications over the infrastructure WMNs necessitate reliable data transmission during their operations. The reliable data transmission has to be achieved in the face of the challenge of congestion and this gave rise to congestion control mechanisms [21].

Congestion control mechanism came to light through Transmission Control Protocol (TCP) over wired Internet in 1988 to avoid transmission failures due to congestion collapse. From that time, the mechanism has evolved through a number of varying notions [36], [12], [8] to enhance its performance over wired mediums. However, frequent transmission failures over wireless mediums, due to its non-deterministic and lossy natures, expose significantly different phenomena during data transmission and thus present the significance of retrospecting the mechanism for wireless networks [6], [5], [9]. In addition, the emergence of multi-hop WMNs provides a new dimension to rethink the mechanism [22], [19] due to utilization of dual-functioning mesh nodes. Here, the notion of dual-functionality implies the responsibility of a mesh node to route data generated from other nodes in addition to its usual data transmission. Nevertheless, the evolution of infrastructure WMNs adds yet another dimension to ponder - the mesh nodes in proximity to gateways suffer from a significantly higher level of congestion. This high extent of congestion introduces a repulsive notion to judge only in favor of the lossy nature of wireless medium in the case of failed transmissions. However, to the best of our knowledge, no research work has attempted to address all of these issues to design an end-to-end congestion control mechanism for multi-hop infrastructure WMNs, which we attempt in this paper.

The presence of two completely different causes of failed transmissions, i.e., congestion and lossy nature of wireless medium, pose the problem of disambiguating the two and developing a novel mechanism for congestion control. Therefore, we attempt to formulate the congestion control problem such that it can be mapped to a well-known decision problem called Multi-Armed Bandit (MAB) problem [32]. Consequently, we analyze its solvability and propose some policies as there exists no solution in the literature for the mapped problem. Based on this approach, our contributions in this paper are as follows:

- We formulate the congestion control problem over an infrastructure WMN and map that to a variant of MAB problem called *restless* MAB [25]. We term the mapped problem as Multi-Armed Bandit Congestion Control ($MABCC$) problem.
- We analyze solvability of $MABCC$ and find that no optimal[1] solution in the literature can be applied to it. Consequently, we propose three myopic policies to solve the problem. The term myopic refers to the fact that the solution considers only a limited amount of state built from observations on the wireless channel.
- We evaluate the efficacy of $MABCC$, with the proposed policies, in a variety of small-scale and large-scale infrastructure WMN topologies through `ns-2` simulation. Simulation results suggest that $MABCC$ can achieve up to a 40% increase in total network throughput and up to a 28% decrease in average energy consumption per bit in comparison to other available variants.
- Besides, we implement our proposed technique over a real testbed and find that $MABCC$ can increase network throughput by up to 52% and decrease average energy per bit by up to 34% compared to other variants over it.

[1]The notion of *optimality* refers to providing maximum network throughput in this paper.

II. Related Work

Different types of congestion control mechanisms have already been proposed for reliable transmissions over the Internet. TCP Tahoe [20] is the first mechanism that uses *slow start* and *congestion avoidance* in this regard. TCP Reno [36] improves the performance of TCP Tahoe by incorporating a moderate treatment to the congestion window in case of failed transmissions using the notion of duplicate acknowledgements (ACKs) in place of an aggressive shrinking of a window to 1, which is done by TCP Tahoe. TCP Newreno [12] further improves TCP Reno by continuing fast recovery as long as it acknowledges outstanding data starting from the initiation of the fast recovery. Afterwards, two successive variants modify the ACK mechanism by exploiting the notions of selective ACKs (TCP Sack [13]) and forward ACKs (TCP Fack [26]). Besides, TCP Vegas [8] enhances the performance of TCP Reno by incorporating proactive measures in its congestion control mechanism.

Subsequently, several modifications of congestion control have been proposed to deal with unreliable wireless links. Snoop [6] first attempts to improve TCP for transmissions over wireless medium by introducing caching at a base station. There is also some other work [5], [16] pertinent to such single-hop wireless connectivity. However, the architecture with single-hop clients does not conform to that of multi-hop WMNs. This architectural gap makes the direct application of single-hop based congestion control mechanisms intractable for multi-hop cases. Besides, variants of TCP for mobile clients [4], [38], [23] mainly focus on the link failure due to the mobility rather than the congestion control, which is not prevailing in the context of multi-hop WMNs either.

In addition, some other work in the literature also addresses congestion control for wireless networks. For example, TCP Westwood [9] adapts TCP Reno [36] to wireless networks. However, its assumption of constant inter-arrival time over all flows is not consistent with the multi-hop flows in our case. Nevertheless, congestion control mechanisms designed for wireless sensor networks [33] mainly focus on the resource constraints, which are not prevalent in an infrastructure WMN.

To the best of our knowledge, the first attempt [22] to design a TCP variant, solely intended for WMNs, proposes a simple approach using explicit notification about congestion from routing layer. However, the notion of the explicit notification breaches the spirit of end-to-end argument. Nonetheless, iTCP [19] proposes the only end-to-end congestion control mechanism for WMNs. However, its consideration of the WMNs is solely dedicated to an ad-hoc architecture and thus ignores the unique characteristics of infrastructure WMNs.

In summary, already proposed mechanisms are not applicable in our case due to five distinctive characteristics of a multi-hop infrastructure WMN - exposure to lossy wireless links, high extent of congestion around infrastructure, multi-hop connectivity, less probable link failures due to static nature of mesh nodes, and high capacity networking by utilizing high bandwidth radios. Considering all these aspects, it would be the most convenient approach to exploit a suitable decision problem for end-to-end congestion control. We attempt to adopt this approach utilizing only transport layer information to ensure no radical change in protocol stack. To achieve this goal, we first map the congestion control problem to a well-known decision problem called MAB so that we can exploit a solution of the decision problem, if any, in the literature. Before focusing on the mapping, we briefly present a network model for the infrastructure WMNs under consideration.

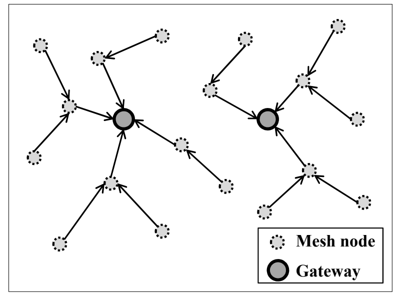


Fig. 1: Network model

III. Network Model

We consider a WMN having a number of static mesh nodes, which can act both as source and relay nodes in multi-hop transmission. In addition to such nodes, the WMN also experiences the presence of one or more gateway nodes. The traffic in the network is destined to these gateway nodes from the mesh nodes through single or multi-hop transmission. Fig. 1 depicts an instance of such network model. We can find such infrastructure WMN in several applications such as AMR [27], construction site monitoring [34], AMI [2], etc.

Now, we attempt to formulate and map the congestion control problem over such infrastructure WMNs to a MAB problem. Before presenting the formulation and mapping, we briefly focus on the MAB problem.

IV. Overview of Multi-Armed Bandit Problem

Multi-Armed Bandit (MAB) [32] is one of the extensively investigated stochastic problems in decision theory. It is a class of sequential resource allocation problems [7], which is effectively utilized in diversified applications such as control theory, queuing systems, sensor management, etc.

In a classical MAB problem, a gambler decides on a sequence of arms or levers of a K-slot machine, i.e., a multi-armed bandit. The gambler iteratively pulls one arm per round and each round provides a reward following a distribution corresponding to the specific arm being pulled. The ultimate target of the gambler is to maximize total rewards in a sequence of trials. There are four key features [25] of a classical MAB problem: 1) All arms are independent of each other. 2) Only one arm can be pulled at a time and the evolution of the pulled arm is uncontrolled, i.e., the reward obtained from the pulled arm is beyond the control of a gambler. 3) The arms that are not pulled remain frozen. 4) The frozen arms contribute *zero* reward in each round.

There are a number of variants of the classical MAB problem in the literature. For example, a MAB problem enabling simultaneous pulling of k number of arms with $k \geq 1$ is called a *multiple plays bandit* [3], a *multiple plays bandit* having evolving rewards from each arm in each round is called a *restless bandit* [41], etc.

A. Problem Formulation of the Classical MAB

We start our formulation from a basic problem - single-armed bandit process. The basic problem contains only one arm and is described using a pair of random sequences - $(\{S(0), S(1), ..., S(n), ...\}, \{R(0), R(1), ..., R(n), ...\})$. Here, $S(n)$ indicates the state of the arm after the completion of n^{th} round and $R(n)$ indicates the reward obtained from the arm during the n^{th} round. Both $S(n)$ and $R(n)$ are random variables with $S(n) \in \Re$ and $R(n) \in \Re_+$. The state of an arm changes depending on its previous states as well as on an independent random variable, $V(n)$ as:

$$S(n) = f_{n-1}\left(S(0), S(1), ..., S(n-1), V(n-1)\right) \tag{1}$$

where $f_{n-1}(\cdot)$ is a function operating during $(n-1)^{th}$ round. The involvement of $V(n-1)$ in the definition of $S(n)$ indicates that the single-armed bandit problem is *not* necessarily a Markov process [37].

The multi-armed bandit process, an extension of the single-armed bandit process, operates on a collection of arms. Let the number of arms be n_a. The classical MAB problem contains a multi-armed bandit process and a controller that operates on only 1 arm at a time keeping other $n_a - 1$ arms frozen. Here, each arm operates in a similar way as we have already described for the single-arm case. For our convenience, we introduce an additional term $n_i(r)$, which indicates the number of times i^{th} arm is being pulled until r^{th} round. We utilize $n_i(r)$ to describe the i^{th} arm as $(\{S_i(n_i(r)), R_i(S(n_i(r)))\})$. Here, $i = 1, 2, ..., n_a$, $r = 0, 1, 2, ...$, and $n_i(r) = 0, 1, 2, ..., r$.

Now, let $A(r)$ be the action of the controller at r^{th} round. Here, $A(r) \in s(i) : 1 \leq i \leq n_a$, where $s(i)$ denotes the states of all arms during a round in which only the i^{th} arm is being pulled. Therefore, $s(i)$ is a unit n_a-vector with only one 1 at the i^{th} position and 0 elsewhere. 1 in $s(i)$ indicates the corresponding arm being pulled whereas 0 indicates a frozen state. Conditioning on $A(r)$, we define the evolution of a classical MAB as follows:

$$S_i(n_i(r+1)) = \begin{cases} f_{n_i(r)}\left(S_i(0), ..., S_i(n_i(r), V_i(n_i(r)))\right) & \text{if } A_i(r) = 1 \\ S_i(n_i(r)) & \text{if } A_i(r) = 0 \end{cases} \tag{2}$$

where $V_i(r)$ denotes an independent random variable. Besides, $n_i(r)$ evolves as follows:

$$n_i(r+1) = \begin{cases} n_i(r) + 1 & \text{if } A_i(r) = 1 \\ n_i(r) & \text{if } A_i(r) = 0 \end{cases}$$

During each round of play, we obtain some reward as we pull any of the arms. We define the reward $(R_i(r))$ obtained by pulling i^{th} arm at r^{th} round as follows:

$$R_i(r) = R_i(S_i(n_i(r)), A_i(r)) \tag{3}$$

$$= \begin{cases} R_i(S_i(n_i(r))) & \text{if } A_i(r) = 1 \\ 0 & \text{if } A_i(r) = 0 \end{cases}$$

A sequence of decisions, i.e., a scheduling policy, $\sigma = (\sigma_1, \sigma_2, ...)$ determines the actions $(A(r))$ in a MAB problem. Here, $A(r)$ depends on all states from the beginning and all actions prior to the current action as follows:

$$A(r) = \sigma_r\left(\Psi_1(r), ..., \Psi_{n_a}(r), A(0), ..., A(r-1)\right) \tag{4}$$

where $\Psi_i(r)$ denotes all states of i^{th} arm from the beginning up to r^{th} round, i.e., $\Psi_i(r) = [S_i(0), S_i(1), ..., S_i(n_i(r))]$.

The ultimate goal in a MAB problem is to maximize the long run performance subject to all the previously mentioned constraints. There are two types of performance measures [24] that are commonly used in the maximization. The first one is *expected total discounted reward*, which is defined as follows:

$$E^{\sigma}\left[\sum_{r=0}^{\infty} \beta^r \sum_{i=1}^{n_a} R_i(S_i(n_i(r)), A_i(r)) \mid \Psi(0)\right] \tag{5}$$

where $0 \leq \beta < 1$ is the discount factor and $\Psi(0) = (\Psi_1(0), ..., \Psi_{n_a}(0))$. The involvement of β in the equation indicates that the expected total discounted reward is suitable for the performance measure of the systems where value of a reward evolves over time, for example, delay-sensitive communication systems.

The second performance measure is *expected average reward*, which is defined as follows:

$$E^{\sigma}\left[\lim_{\mathbf{r} \to \infty} \frac{1}{\mathbf{r}} \sum_{r=0}^{\mathbf{r}} \sum_{i=1}^{n_a} R_i(S_i(n_i(r)), A_i(r)) \mid \Psi(0)\right] \tag{6}$$

The expected average reward emphasizes all rewards in a similar way and thus is suitable for throughput measure in a communication system.

Now, we can find the optimal solution of a classical MAB problem using *dynamic allocation index* or *Gittins index* [14]. The *Gittins index* assigns the highest index to the optimal decision in each round. Therefore, the controller merely operates on arms following the associated indices.

Next, we utilize the formulation, presented so far, as a basis of our further investigation on end-to-end congestion control in infrastructure WMNs. First, we attempt to map the congestion control problem as an instance of the MAB problem.

V. Mapping: Congestion Control to MAB

In this section, we map an end-to-end congestion control problem in infrastructure WMNs to a suitable variant of MAB. We start by considering the following correspondences:

- In each round of congestion control mechanism, we can find several fixed types of changes in the congestion window. For example, *slow start* changes the window by doubling it, *congestion avoidance* changes the window by incrementing it, etc. We consider each type of such possible changes as an arm of a bandit in a MAB.
- We adopt posteriori probability of each possible change in the congestion window as a state of an arm in MAB.

A posteriori probability provides the extent to which the associated change in a congestion window would be the optimal one at an ongoing round.

- We define the notion of reward for each arm in terms of contribution to total network throughput after applying a change in a congestion window.
- In an end-to-end congestion control mechanism, all mesh nodes in a WMN independently change their congestion windows as a mesh node changes its congestion window based only on the knowledge extracted from its *own* experience. However, the underlying essence of the extracted knowledge continually changes even during the decision making process of a mesh node as other mesh nodes in the network also simultaneously take their own independent decisions. This parallel decision making paradigm in a WMN introduces a notion of independent variability in overall network performance. This independent variability suggests independent evolution of the posteriori probability, which in turn confirms association of an independent variability in the evolution of the state of an arm. Therefore, we can guarantee the existence of $V_i(r)$, an independent random variable in the change of a state, as in Eq. 2. Moreover, other dynamically changing phenomena such as different environmental effects also contribute to the existence of $V_i(r)$.

It is worth mentioning that the correspondences are also applicable to congestion control mechanisms over any type of network. However, we specifically focus on the infrastructure WMN as the congestion control over such network is yet to be addressed in the literature, which we have mentioned in Section II. Now, considering all the above mentioned correspondences, we start our mapping with a simplified version of congestion control problem to effectively map that to the classical MAB problem. In the simplified version, we assume that only one change in the congestion window is optimal throughout the lifetime of a data flow. In practice, the optimal decision may vary after each transmission attempt. Therefore, we release the assumption later in this section after our mapping to the classical MAB problem.

Our assumption reduces the congestion control problem to a simple search problem in which our job is to find the optimal choice to change a congestion window. Due to the presence of dynamically varying phenomena in wireless environments, a mesh node may mistakenly miss the optimal choice in some cases. To model this scenario, we formulate conditional probability of finding the optimal choice as follows:

$$P\left(Cur_i \mid Opt_j\right) = \delta_{ij} q_j \tag{7}$$

where Cur_i is an event that currently assumed optimal decision is the i^{th} change and Opt_j is an event that the actual optimal decision is the j^{th} change. Besides, δ_{ij} denotes *Kronecker's delta* that is defined as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

In addition, q_j denotes the probability of the j^{th} change in a congestion window to be the optimal one. We utilize q_j to define the reward achieved by applying the i^{th} change in the r^{th} round of congestion control mechanism as follows:

$$R_i(S_i(n_i(r)), A_i(r)) = \begin{cases} N_{b\_opt}(r) \times p_i(r) \times q_i & \text{if } A_i(r) = 1 \\ 0 & \text{if } A_i(r) = 0 \end{cases}$$

where $N_{b\_opt}(r)$ is the number of successfully transmitted bits or contribution to the total network throughput that can be achieved using the optimal change in the r^{th} round and $p_i(r)$ is the posteriori probability of assuming that the i^{th} change is the optimal one at the end of the r^{th} round. We define the posteriori probability, $p_i(r)$ as

$$p_i(r) = \frac{N_{b\_i}(r)}{N_{b\_opt}(r)} \tag{8}$$

where $N_{b\_i}(r)$ is the number of successfully transmitted bits using the i^{th} change in the r^{th} round.

Now, we can deduce an expression of expected total network throughput for the optimal solution of a sequence of changes, σ in a WMN as follows:

$$\begin{aligned} & E^{\sigma}\left[\lim_{\mathbf{r}\to\infty} \frac{1}{\mathbf{r}} \sum_{r=0}^{\mathbf{r}} \sum_{i=1}^{n_a} N_{b\_i}(r) \times q_i\right] \qquad (9) \\ & = E^{\sigma}\left[\lim_{\mathbf{r}\to\infty} \frac{1}{\mathbf{r}} \sum_{r=0}^{\mathbf{r}} \sum_{i=1}^{n_a} N_{b\_opt}(r) \times p_i(r) \times q_i\right] \\ & = E^{\sigma}\left[\lim_{\mathbf{r}\to\infty} \frac{1}{\mathbf{r}} \sum_{r=0}^{\mathbf{r}} \sum_{i=1}^{n_a} R_i(S_i(n_i(r)), A_i(r)) \mid \Psi(0)\right] \end{aligned}$$

where $\Psi(0)$ denotes the starting state of the network when data transmission in the network is yet to be started.

It is worth mentioning that we consider the notion of network throughput in terms of total number of bits successfully transmitted *within a round*. We can exploit *Renewal Reward Process* [18], a variant of stochastic counting process, if we want to consider the network throughput in terms of the total number of bits successfully transmitted *in one second*.

In a *Renewal Reward Process*, events occur with independent and identically distributed (*iid*) inter-arrival times and each inter-arrival time provides a reward, which is also *iid*. The expected rate of reward in the process is $\frac{E(R)}{E(X)}$, where $E(R)$ and $E(X)$ are expected reward within an inter-arrival time and expected value of an inter-arrival time respectively. In the case of a congestion control mechanism, we can consider the duration of a round as an inter-arrival time. Here, the duration is *iid* due to its dependence on round trip time, which depends on independently distributed network conditions such as fading, multi-path effect, etc. Besides, the number of successfully transmitted bits emulates the notion of reward assuming its identical behavior over a long period of time along with its independent nature, which we have already suggested at the beginning of this section while discussing the fourth correspondence in our mapping. Therefore, Eq. 9 provides the expected reward ($E(R)$) in disguise of a renewal reward process.

Besides, we assume that impacts of network environment and relative distance to a gateway are dominant factors in our case, which influence round trip time in an infrastructure WMN. Consequently, the expected value of an inter-arrival time ($E(X)$) becomes independent of the congestion control mechanism and thus provides a constant value in the WMN. Therefore, maximization of the first notion of throughput (with respect to *one round*), as stated in Eq. 9, will also end up with similar optimal results in the case of the maximization of the second notion of throughput (with respect to *one second*). Thus, our notion of throughput leads to a mapping, which is invariant of the unit of throughput.

Now, the final step of Eq. 9 perfectly matches the expression of expected average reward formulated in Section IV-A (Eq. 6). Therefore, the simplified form of congestion control mechanism apparently maps to the classical MAB problem that can provide optimal solution using *Gittins index* [14] to maximize total network throughput. However, we can find through careful investigation that the second feature of the classical MAB problem (mentioned in Section IV), to hold the unconsidered arms frozen, is absent in our formulation. The reason behind this missing feature is - if we apply the i^{th} change in the r^{th} round and find that the i^{th} change is not the optimal one at the end of that round, then the posteriori probability ($p_i(r)$) of that change, i.e., the state of that arm, is evolved as

$$p_j(r+1) = \begin{cases} \frac{p_j(r)\times(1-q_j)}{d} & \text{if } j = i \\ \frac{p_j(r)}{d} & \text{if } j \neq i \end{cases}$$

where $d = 1 - p_i(r) \times q_i$. Here, the posteriori probability of the i^{th} change decreases, whereas the posteriori probability of all other changes increase. Therefore, the states of unpicked changes do not remain frozen.

To alleviate the inconsistency in the mapping, we modify the notion of state in our formulation. We redefine the state of the i^{th} arm at the r^{th} round as an unnormalized probability p'_i with $p'_i(0) = p_i(0)$. Here, $p'_i(r)$ evolves as follows:

$$p'_i(r+1) = \begin{cases} p'_i(r)(1-q_i) & \text{if } A_i(r)=1 \\ p'_i(r) & \text{if } A_i(r)=0 \end{cases}$$

We also have to modify the notion of reward using the modified form of a state as follows:

$$R'_i(S_i(n_i(r)), A_i(r)) = \begin{cases} N_{b\_opt}(r)\times p'_i \times q_i & \text{if } A_i(r)=1 \\ 0 & \text{if } A_i(r)=0 \end{cases}$$

We also adjust the objective function to accommodate the modified form of the reward as follows:

$$E^{\sigma}\left[\lim_{\mathbf{r}\to\infty} \frac{1}{\mathbf{r}} \sum_{r=0}^{\mathbf{r}} \sum_{i=1}^{n_a} R'_i(S_i(n_i(r)), A_i(r)) \mid \Psi(0)\right]$$

This adjustment absorbs the variability in the states of frozen arms. Therefore, the adjustment perfectly maps the congestion control problem to the classical MAB problem.

We have started our mapping with a simplified form of the congestion control mechanism. The simplified form assumes only one optimal choice for changing a congestion window. In practice, this assumption does not hold and the optimal choice varies with progression of rounds. Therefore, to reflect this variation in evolution of the optimal choice, we redefine the evolution of the state of an arm as follows:

$$\begin{aligned} &S_i(n_i(r+1)) \qquad (10) \\ &= \begin{cases} f_{n_i(r)}\left(S_i(0), ..., S_i(n_i(r), V_i(n_i(r), A_i(r))\right) & \text{if } A_i(r) = 1 \\ S_i(n_i(r)) & \text{if } A_i(r) = 0 \end{cases} \end{aligned}$$

The only difference between Eq. 2 and Eq. 10 is the additional involvement of $A_i(r)$, i.e., the change in r^{th} round, in Eq. 10. This form of evolution of a state indicates a coherence to one of the variants of the classical MAB problem called the *restless* MAB [25] problem, which adopts a dynamic variation in the optimal choice of an arm rather than retaining the choice fixed all over the operational rounds. Therefore, the accommodation of the variation along with only *one* change in a congestion window per round effectively maps the congestion control problem to a *single processor restless* MAB problem. Following this mapping to a variant of classical MAB problem, we term the congestion control problem as *Multi-Armed Bandit Congestion Control* ($MABCC$). Next, we analyze solvability of the mapped problem.

VI. Solvability of $MABCC$

We find that the underlying trend of $MABCC$ follows a notion of the restless bandit problem as the optimal change in a congestion window varies in different rounds. The variation comes into play due to a dynamic evolution of rewards associated with the changes. The dynamically varying rewards render the Gittins index policy as suboptimal [24] in congestion control mechanism. Besides, finding the optimal solution of a general restless bandit problem is known to be PSPACE-hard [29] and little progress has been made so far [15].

Whittle et al., [41] attempt to address the problem and propose a heuristic index policy for the restless bandit problem. However, according to its definition of indexability, the states of passive or inactive arms must monotonically increase with an increase in *subsidies* of the arms [40]. Here, the *subsidy* suggests an associated value to each arm whose increase intensifies the optimality of the decision to consider the corresponding arm as passive or inactive. We can get the essence of *subsidy* in $MABCC$ as *the increase in network congestion level*, as the probability of inactivating a change in a congestion window intensifies if the increase in network congestion level corresponding to the change increases. However, according to our formulation in Section V, the state of each change in a congestion window monotonically decreases with an increase in *subsidy*. This happens as the state of an arm is a monotonically increasing function of its posteriori probability to be the optimal choice and the posteriori probability is monotonically decreasing function of *subsidy*. Therefore, the change in state with an increase in *subsidy* violates the requirement of Whittle indexability.

In summary, to the best of our knowledge, the formulation of $MABCC$ is not suitable for any indexing policies that are proposed in the literature so far. However, if any general indexing policy for the restless MAB would be found in future, then we could exploit that for optimal solution of $MABCC$. Nevertheless, suitable *myopic policy* [15], [25] can provide a near-optimal solution in this regard. Therefore, we attempt to propose some myopic policies for $MABCC$.

VII. Myopic Policies for $MABCC$

We propose three myopic policies to overcome the unavailability of a suitable index policy for the formulation of $MABCC$. While proposing the policies, we adopt a conservative approach in the presence of failed transmissions considering the high extent of congestion around a limited number of gateways. Besides, the proposed policies also keep a provision to aggressively increase the transmission rate in favorable conditions to cope with transmission failures due to lossy wireless medium. We attempt to adopt these opposing notions to be close to the optimal sequence (σ in Eq. 9) that maximizes network throughput in the WMN. To the best of our understanding, the optimal sequence should experience recurring occurrences of both conservative and aggressive approaches due to simultaneous presences of lossy link and the high extent of congestion around the gateways in the infrastructure WMN. Here, the simultaneous presences frequently change q_i and thus, in turn, p'_i that necessitates the recurring occurrences.

Now, to accumulate the two opposing notions, our policies exploit the latest available experience from an ongoing data transmission while updating a congestion window. We utilize three measures to get an essence of the latest available experience - round trip time (RTT), number of consecutive duplicate acknowledgements (DupACK), and number of consecutive retransmission timeouts (RTO). We utilize these measures in distinguished manner considering their extents of impact on congestion in an infrastructure WMN.

1. Policy with RTT: We exploit relative change in a newly computed RTT compared to the last observed value to extract status of current congestion level in the network. In the presence of low extent of congestion, it is likely that the RTT will get decreased in successive transmissions. Therefore, we increase the size of congestion window when we experience a decrease in RTT. Here, we enable a combination of additive and multiplicative increase. We consider a threshold value to distinguish between the additive and multiplicative increase. We perform a multiplicative increase on the window up to the threshold and then carry out an additive increase onward as

$$C_{new} = \begin{cases} \lceil C_{old} \times (1 + \gamma_{inc}) \rceil & \text{if } R_o > R_n \text{ and } C_{old} < C_{th} \\ C_{old} + \delta_{inc} & \text{if } R_o > R_n \text{ and } C_{old} \geq C_{th} \end{cases}$$

where C_{new} is the size of newly computed congestion window, C_{old} is the size of last computed congestion window, δ_{inc} is the step in additive increase, γ_{inc} is a multiplier in multiplicative increase, R_n is the RTT of last successful transmission, R_o is the RTT of prior to the last successful transmission, and C_{th} is the threshold on the congestion window. Here, γ_{inc} evolves with successive successful transmissions as $\gamma_{inc} = \gamma_{inc} + \frac{1}{\alpha_{inc}}$, where α_{inc} provides a step in increase of γ_{inc} per successful

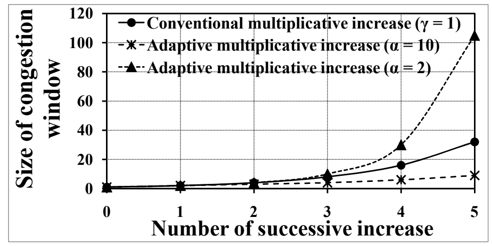


Fig. 2: Conventional versus adaptive multiplicative increase

transmission. α_{inc} enables an adaptive nature in multiplicative increase compared to a conventional multiplicative approach (Fig. 2).

On the other hand, an increase in RTT may indicate a rising extent of congestion in a WMN. However, the increase in RTT may also result due to various other reasons such as increase in intermediate queuing delay, backoff delay, etc. Nevertheless, successive increases in RTT emphasize more to the rising extent of congestion than these factors. Therefore, we focus on the number of consecutive increases in RTT in our mechanism. We perform an additive decrease on the congestion window for the first increase and then carry on a multiplicative decrease from the second one onward as

$$C_{new} = \begin{cases} C_{old} - \delta_{dec} & \text{if } R_o < R_n \text{ and } N_{i\_R} = 1 \\ \left\lfloor \frac{C_{old}}{(1+\gamma_{dec})} \right\rfloor & \text{if } R_o < R_n \text{ and } N_{i\_R} > 1 \end{cases}$$

where δ_{dec} is the step in additive decrease, γ_{dec} is a multiplier in multiplicative decrease, and N_{i_R} is the number of successive increases in RTT. Here, we consider four intervals on N_{i_R} while determining the value of γ_{dec}. The four intervals are - $((2,3),(4,5),(6,7),(8,\infty))$. The corresponding values of γ_{dec} are - $(0.5, 1.0, 1.5, 2.0)$.

2. Policy with DupACK: Presence of a duplicate ACK indicates intermediate transmission failure or incomplete transmission that may result from a rising extent of congestion level as well as from various other reasons such as fading, noise, increase in queuing delay, etc. Nevertheless, similar to the number of successive increases in RTT in the previous policy, the increase in number of successive DupACKs emphasizes more on the rising extent of congestion compared to other factors. Therefore, we focus on the number of consecutive DupACKs in our mechanism. We perform an additive decrease in the case of the first DupACK and then a multiplicative decrease from the second consecutive DupACK onward in a similar fashion to that which we have elaborated in the previous policy. However, the presence of a DupACK emphasizes more on the rising extent of a congestion level than that emphasized by an increase in RTT. Therefore, we exploit double values for δ_{dec} and γ_{dec} for DupACK compared to that in previous policy.

3. Policy with RTO: The presence of an RTO implies a failed or incomplete transmission. It signifies the highest probability of a rising extent of congestion level among three measures that we consider. We address the highest probability by performing multiplicative decrease from the very beginning of sensing RTOs. We follow a similar equation as illustrated in the case of the first policy, except we consider the ranges on number of successive RTOs as $((1, 2), (3, 4), (5, 6), (7, \infty))$ with the corresponding values of γ_{dec} as $(1.0, 2.0, 3.0, 4.0)$. Moreover, we cut the value of C_{th} to half of the size of the congestion window prior to the expiration of the retransmission timer to emphasize RTO as the most probable cause of congestion.

Next, we evaluate effectiveness of a combined application of our proposed policies in infrastructure WMNs. We perform both simulation and testbed experiment for the evaluation.

VIII. Simulation Evaluation of $MABCC$

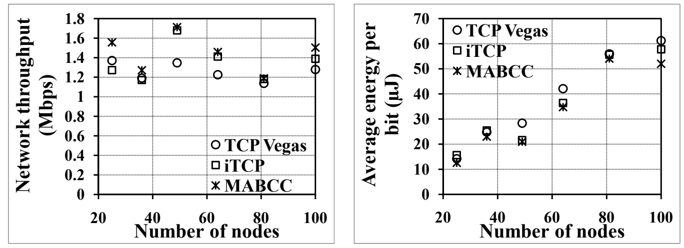
We evaluate the performance of $MABCC$ against that of two other congestion control mechanisms available in the literature - iTCP [19] and TCP Vegas [8]. To the best of our knowledge, iTCP adopts the only known end-to-end congestion control mechanism for optimized network throughput in a WMN. Besides, TCP Vegas is known to perform the best in WMNs [19] among other end-to-end variants available in the literature. Nonetheless, other standard TCPs (for example, TCP Tahoe [31], TCP Newreno [12], etc.) do not perform well over multi-hop transmission due to their incompetency for being limited to small-sized congestion windows [19]. TCP Vegas also experiences a similar small-sized congestion window. Therefore, we perform our evaluation against TCP Vegas as a representative of the standard TCPs and iTCP as a variant for WMNs. Here, we adopt two metrics - total network throughput and average energy consumption per bit.

In our evaluation, we simulate two different scales of WMNs - small-scale and large-scale, using `ns-2` with varying types of data flow. Simulation settings in both the scales share some common system parameters. Therefore, we first present the simulation settings with these parameters and then elaborate the experimental results.

A. Simulation Settings and Parameters

We consider DSDV [30] as network layer protocol in each topology. We adopt Two-Ray Ground reflection model [1] as the radio wave propagation model in our experiment. Besides, we utilize a 54Mbps 802.11g radio in each mesh node. Transmission and sensing ranges of the radio are 140m and 280m accordingly. We consider omnidirectional antenna and droptail priority queue having a capacity of 1000 packets at each mesh node. Finally, we adopt the measurements provided in [35] to estimate the power consumptions of 802.11g radio in different modes. With these considerations, we simulate each network topology for 75 seconds with an initial interval of 50 seconds that lets the routing paths be stable.

In addition to the common system parameters, we also set parameters pertinent to $MABCC$ in our experiment. We utilize 128 as maximum congestion window size and set initial value of congestion window threshold (C_{th}) as half of that. We set the step per successful transmission (α_{inc}) as 30. Besides, we utilize 1 as both steps in additive increase (δ_{inc}) and additive decrease (δ_{dec}). We empirically set all



(a) Total network throughput

(b) Average energy per bit

Fig. 3: Performances of TCP Vegas, iTCP, and $MABCC$ in a small-scale grid WMN with 1 base station at a corner

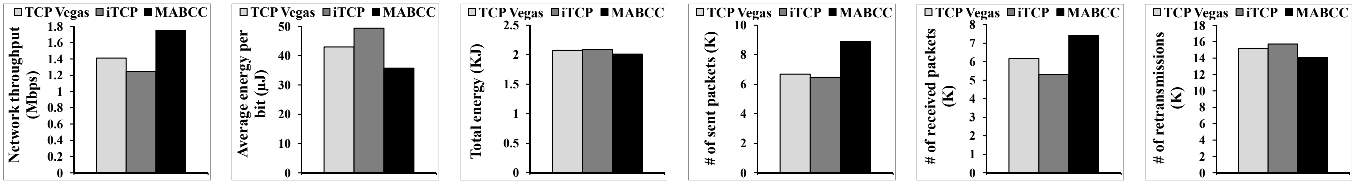
these parameters to improve performance under high extent of congestion.

B. Small-scale Evaluation

We start our experiment in small-scale WMN topologies. First, we evaluate our policies in grid mesh networks. We consider 6 grid meshes having 5^2, 6^2, ..., 10^2 nodes in $500{\times}500\text{m}^2$ coverage area. We place a gateway at a corner of the coverage area (at (500m, 500m)). Each node transmits 200 packets per second (pps) towards the gateway with 1KB application payload in each packet. Fig. 3 depicts performances of three congestion control mechanisms over these flows.

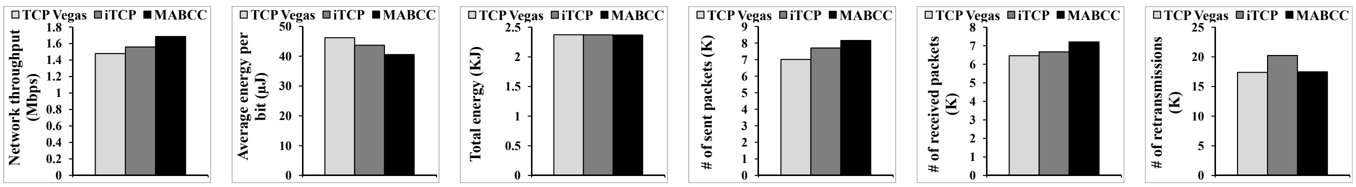
In Fig. 3a, our policies for $MABCC$ always exhibit better total network throughput than that using TCP Vegas and iTCP. We achieve 15% and 7% average improvement in total network throughput with $MABCC$ in comparison to TCP Vegas and iTCP accordingly (90% confidence intervals are 0.13Mbps, 0.06Mbps, and 0.13Mbps for MABCC, TCP Vegas, and iTCP respectively). However, we find that $MABCC$ consumes comparable total energy in comparison to TCP Vegas and iTCP. Therefore, the combined effect of the enhanced total network throughput and comparable total energy consumption results in 13% and 7% decreases in average energy consumption per transmitted bit for $MABCC$ in comparison to TCP Vegas and iTCP (90% confidence intervals are 11.51μJ, 12.42μJ, and 11.98μJ respectively), which is depicted in Fig. 3b.

Next, we consider a coverage area of 1Km^2 with 100 randomly placed mesh nodes and a gateway. We simulate the network for two different positions of the gateway. First, we place the gateway at a corner of the coverage area (at (1Km, 1Km)). Each node transmits to the gateway at a rate of 100 pps with 1KB application payload in each packet. Fig. 4 shows the performances of three congestion control mechanisms in this WMN setting. Here, $MABCC$ achieves 24% and 40% increased total network throughput than that using TCP Vegas and iTCP accordingly (90% confidence intervals are 0.16Mbps, 0.06Mbps, and 0.08Mbps respectively). We analyze reasons behind the increase in network throughput in Fig. 4d and Fig. 4e. These two figures depict that the congestion control mechanism of $MABCC$ enables mesh nodes to send and successfully receive an increased number of packets compared to TCP Vegas and iTCP. However, the



(a) Throughput (b) Energy per bit (c) Total energy (d) Total sent packets (e) Total received packets (f) Total retransmissions

Fig. 4: Performances of TCP Vegas, iTCP, and $MABCC$ in a small-scale random WMN with 1 base station at corner



(a) Throughput (b) Energy per bit (c) Total energy (d) Total sent packets (e) Total received packets (f) Total retransmissions

Fig. 5: Performances of TCP Vegas, iTCP, and $MABCC$ in a small-scale random WMN with 1 base station at middle

number of retransmission attempts in $MABCC$ is less than that using TCP Vegas and iTCP (Fig. 4f). The combined effect of these three factors results in comparable total energy consumption for $MABCC$ compared to the other two variants (Fig. 4c). Finally, the increased total network throughput and comparable total energy consumption of $MABCC$ allow it to achieve 17% and 28% decreased average energy consumption per bit (90% confidence intervals are 3.78μJ, 2.45μJ, and 2.95μJ respectively) than that using TCP Vegas and iTCP.

Besides, we simulate another WMN topology by changing the position of the gateway from the corner to the center of the coverage area, retaining all other parameters same as the previous setting. This topology experiences a lower extent of congestion than the previous one as the placement of the gateway at the middle increases the number of mesh nodes in proximity to the gateway and thus moderately distributes the extent of congestion over them. Fig. 5 shows the performances of three congestion control mechanisms in the presence of such lower extent of congestion. Here, $MABCC$ achieves 14% and 7% improvement in total network throughput (90% confidence intervals are 0.06Mbps, 0.04Mbps, and 0.04Mbps respectively), and 12% and 7% improvement in average energy per bit (90% confidence intervals are 1.36μJ, 1.35μJ, and 1.0μJ respectively) in comparison to TCP Vegas and iTCP accordingly. These improvements exhibit lower values in comparison to that with the gateway at a corner. The reason behind such outcomes is the conservative approach adopted in our proposed myopic policies to guard against a high extent of congestion in infrastructure WMNs.

C. Large-scale Evaluation

We also evaluate $MABCC$ in a large-scale WMN topology containing 1000 randomly placed mesh nodes within a coverage area of 10km^2 having randomly placed gateways. We consider two topologies having 40 and 20 gateways to analyze the impact of varying extents of congestion. We utilize 500 flows each having 200 pps data rate with 1KB application data in each packet. Here, 250 flows operate from random sources to the nearest base stations and the remaining 250 flows operate to random destinations from the nearest base stations.

The extent of congestion intensifies with a decrease in the number of gateways. Therefore, the topology with 20 gateways experiences a higher extent of congestion than that with 40 gateways. Consequently, following the findings of the previous subsection, $MABCC$ should achieve higher performance improvement with 20 gateways than that with 40 gateways, which we indeed find in our analysis.

$MABCC$ achieves 13% and 19% increased total network throughput, and 11% and 15% decreased average energy consumption per bit in comparison to TCP Vegas and iTCP accordingly with 20 gateways. On the other hand, $MABCC$ achieves 11% and 15% higher total network throughput, and 10% and 12% lower average energy consumption per bit with 40 gateways. Therefore, the improvement with 40 gateways exhibit smaller values in comparison to that with 20 gateways, which validates our initial claim. We skip detail of these results due to space limitation.

IX. Testbed Evaluation of $MABCC$

In addition to the simulation evaluation, we also evaluate $MABCC$ over a real testbed. Our testbed consists of ten nodes placed in 2^{nd} floor of the EE building at Purdue University. In the testbed, we utilize a Beagleboard [10] having a 54Mbps Trendnet (TEW-648UB) 802.11 dongle [28] as a node. We show node placements in the testbed in Fig. 6. Here, rectangles denote source nodes (node-1, node-2, and node-10), circles denote intermediate nodes, and the triangle denotes a base station (node-6).

In the testbed evaluation, we exploit two custom-written C programs to send and receive data in a similar way of a file transfer. Using these programs, we enable data transmissions having a rate of 75pps from node-1 and node-2, and having a rate of 150pps from node-10. We take five iterations of data transmission with 10 seconds of delay in between two successive iterations. In each iteration, node-1 and node-2

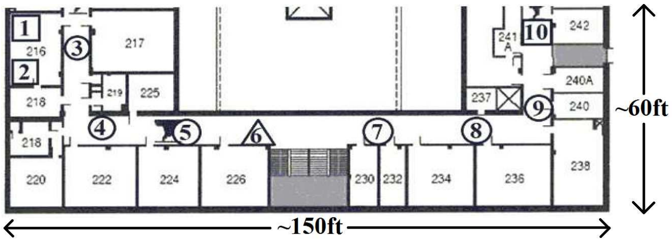


Fig. 6: Node placements in real testbed evaluation

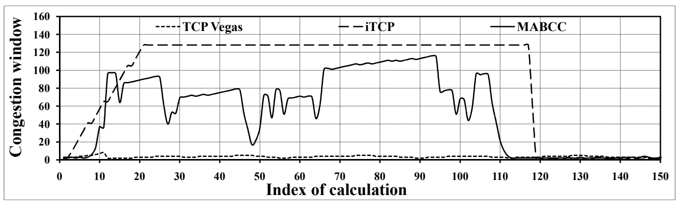


Fig. 7: Comparison of congestion windows

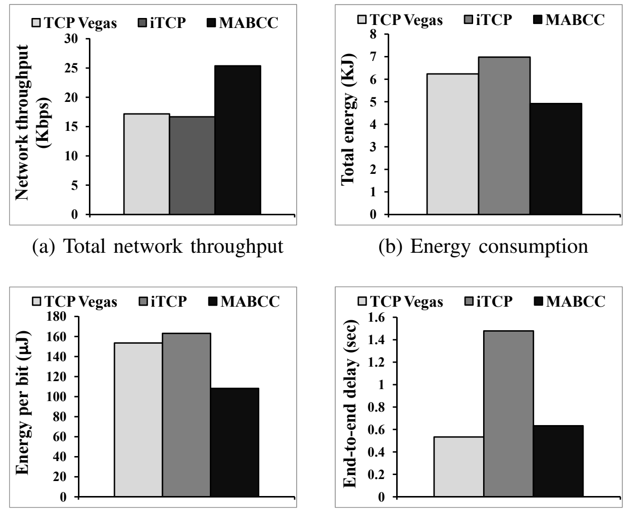
transmit 750 packets and node-3 transmits 1500 packets having 1KB payload in each packet. We present our testbed results by taking average over these iterations.

We implement $MABCC$ as a loadable Ubuntu kernel module. We separately enable data transmission in our testbed using the kernel modules of TCP Vegas, iTCP, and $MABCC$. The $MABCC$ module utilizes similar parameters that we have presented in Section VIII-A, except we set α_{inc} to 3 due to the smaller size of our testbed in comparison to that of networks considered in simulation. Besides, we enable Olsrd [39] in each node to enable multi-hop transmissions over our testbed. Finally, we estimate the energy consumption of a node following the equations presented in [17].

A. Performance Evaluation

We start our testbed data analysis through evaluating network performances of TCP Vegas, iTCP, and $MABCC$. First, we evaluate overall network throughput. Fig. 8a depicts the network throughput. $MABCC$ achieves 48% and 52% increased network throughput in comparison to TCP Vegas and iTCP respectively. The increase is achieved due to the suitable adjustment of the congestion window in $MABCC$, which we present in Fig. 7. This figure shows 150 consecutive changes in the congestion window of node-10. It reveals that TCP Vegas follows a highly conservative approach and iTCP follows a highly aggressive approach of data transmission. However, our proposed $MABCC$ adopts in between both of them and thus achieves a combination of both conservative and aggressive approaches that we have mentioned in Section VII. Besides, $MABCC$ allows quick decreases in the congestion window in response to transmission failures. The notion of quick response enables $MABCC$ to achieve better fairness compared to other variants. We analyze the fairness through standard deviation over throughput per flow. The standard deviation of $MABCC$ is 2% and 40% smaller than that using TCP Vegas and iTCP. Therefore, $MABCC$ ensures better fairness in comparison to other two variants.

Next, we compare total energy consumption over the network. Fig. 8b shows total energy consumptions over all iterations for three variants. It reveals that $MABCC$ consumes 21% and 30% less energy in comparison to TCP Vegas and iTCP though it achieves increased throughput. Consequently, $MABCC$ requires 30% and 34% lower average energy per transmitted bit than that using TCP Vegas and iTCP (Fig. 8c).



(a) Total network throughput

(b) Energy consumption

(c) Average energy per bit

(d) Average end-to-end delay

Fig. 8: Comparison of testbed performances

In addition, we also measure end-to-end delays for all of the three variants (Fig. 8d). End-to-end delays of both $MABCC$ and TCP Vegas are small whereas the end-to-end delay of iTCP is significantly large. Here, $MABCC$ achieves a 57% smaller end-to-end delay compared to that using iTCP.

B. System Overhead

Now, we analyze system overhead of $MABCC$. First, we compare CPU usages of three variants. The simplified approach of $MABCC$ enables it to operate with significantly low CPU time - 32% and 18% decreased compared to TCP Vegas and iTCP. In addition, we also compare memory usages of the three variants. Here, we focus on both buffer and cache memory usages. The buffer memory is used as a virtual disk while data is being *read from* the disk, whereas the cache memory is used as a virtual disk while data is being *written to* the disk. Our testbed results suggest that $MABCC$ requires only 4% and 2% higher memory (sum of buffer and cache memories) compared to other variants. Therefore, $MABCC$ saves significant CPU cycles incurring a marginal increase in memory occupancy compared to other variants.

C. Network Analysis and Overhead

Next, we analyze network conditions and overheads in our testbed experiment. We analyze the network condition to show that our results are collected in similar network conditions for all variants. We analyze network conditions through *link quality* and *signal strength* measured from the base station in our testbed. The measured values show that link qualities are similar for all variants ($\sim$ 100%). Besides, signal strengths also exhibit close values ($\sim 80\%$). In addition, we also measure noise levels, which are found to be 0 for all variants.

Finally, we analyze network overhead due to routing, i.e., due to enabling Olsrd, using Wireshark [11]. We find 17% network overhead for Olsrd in terms of the number of packets. However, network overhead in terms of the number of bytes is only 7% and thus is not highly significant in our testbed.

X. Conclusion

Wireless packet loss in multi-hop infrastructure wireless mesh networks is caused by two factors - lossy nature of wireless medium and high extent of congestion around gateways in the network. Due to the intermingled nature of these two factors, prior work on congestion control in wireless networks does not directly apply to infrastructure wireless mesh networks. In this paper, we attempt to exploit a well-known decision problem called the Multi-Armed Bandit problem for efficient congestion control in infrastructure WMNs. We formulate the congestion control problem and map it to a variant of the decision problem, which we term as $MABCC$. We find that no optimal solution exists in the literature for $MABCC$. In the absence of an optimal solution, we propose three myopic policies for $MABCC$. The term myopic refers to the fact that the solution considers only a limited amount of state built from observations on the wireless channel. We evaluate these policies for $MABCC$ against available variants of end-to-end congestion control mechanism over varying small-scale and large-scale infrastructure WMNs through `ns-2` simulation and real testbed experiment. Our evaluation reveals that $MABCC$ exhibits up to 52% and 34% improvement in network throughput and average energy consumption per bit compared to already available TCP variants.

References

[1] Two-ray model, *http://www.isi.edu/nsnam/ns/doc/node219.html*.
[2] Advanced Metering Infrastructure. *http://www.buss.com/content/public/en/power_systems/solutions/ami.html*.
[3] R. Agrawal, M. Hegde, and D. Teneketzis. Multi-armed bandit problems with multiple plays and switching cost. *Stochastics An International Journal of Probability and Stochastic Processes*, 29(4):437–459, 1990.
[4] V. Anantharaman, S.J. Park, K. Sundaresan, and R. Sivakumar. TCP performance over mobile ad hoc networks: a quantitative study. *Wireless Communications and Mobile Computing*, 4(2):203–222, 2004.
[5] B. S. Bakshi, P. Krishna, N. H. Vaidya, and D. K. Pradhan. Improving performance of tcp over wireless networks. In *Proceedings of the 17th International Conference on Distributed Computing Systems*, page 365, May 27–30 1997.
[6] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving tcp/ip performance over wireless networks. In *Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 2–11, November 13–15 1995.
[7] D.A. Berry and B. Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman and Hall London, 1985.
[8] L. S. Brakmo and L. L. Peterson. Tcp vegas: End to end congestion avoidance on a global internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–1480, October 1995.
[9] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang. Tcp westwood: end-to-end congestion control for wired/wireless networks. *Wireless Networks*, 8(5):467–479, September 2002.
[10] G. Coley. Beagleboard system reference manual, 2009.
[11] G. Combs et al. Wireshark-network protocol analyzer. *July*, 2008.
[12] S. Floyd and T. Henderson. Rfc 2582: The newreno modification to tcp's fast recovery algorithm. April 1999.
[13] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky. Rfc 2883: An extension to the selective acknowledgement (sack) option for tcp. 2000.
[14] J.C. Gittins and D.M. Jones. A dynamic allocation index for the sequential design of experiments. *Progress in statistics*, 241:266, 1974.
[15] S. Guha, K. Munagala, and P. Shi. Approximation algorithms for restless bandit problems. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 28–37. Society for Industrial and Applied Mathematics, 2009.
[16] G. Holland and N. Vaidya. Analysis of TCP performance over mobile ad hoc networks. *Wireless Networks*, 8(2/3):275–288, 2002.
[17] A.B.M.A.A. Islam, M.S. Hossain, V. Raghunathan, and Y.C. Hu. Backpacking: Deployment of heterogeneous radios in high data rate sensor networks. In *Proceedings of ICCCN*, pages 1–8. IEEE, 2011.
[18] A.B.M.A.A. Islam, C. S. Hyder, H. Kabir, and M. Naznin. Finding the Optimal Percentage of Cluster Heads from a New and Complete Mathematical Model on LEACH. *WSN*, 2(2):129–140, February 2010.
[19] A.B.M.A.A. Islam and V. Raghunathan. End-to-End Congestion Control in Wireless Mesh Networks using a Neural Network. In *Wireless Communications And Networking Conference*, March 28–31, 2011.
[20] V. Jacobson. Congestion avoidance and control. *ACM SIGCOMM Computer Communication Review*, 18(4):314–329, August 1988.
[21] N. Li, G. Chen, and M. Zhao. Autonomic Fault Management for Wireless Mesh Networks. *Electronic Journal for E-Commence Tools and Applications (eJETA)*, 2(4), January 2009.
[22] C. Liu, F. Shen, and M.-T. Sun. A unified tcp enhancement for wireless mesh networks. In *Proceedings of the Internationall Conference Parallel Processing*, pages 71–76, September 2007.
[23] J. Liu and S. Singh. ATCP: TCP for mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 19(7):1300–1315, 2002.
[24] K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of Whittles index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2008.
[25] A. Mahajan and D. Teneketzis. Multi-armed bandit problems. *Foundations and Applications of Sensor Management*, pages 121–151, 2008.
[26] M. Mathis and J. Mahdavi. Forward acknowledgement: refining tcp congestion control. In *Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 281–291, August 28–30 1996.
[27] Mesh Networks: An Optimum Solution for AMR. *http://www.electricenergyonline.com/?page=show_article&mag=17&article=129*.
[28] Mini Wireless N USB Adapter. *http://www.trendnet.com/products/proddetail.asp?prod=190_TEW-648UB&cat=175*.
[29] C.H. Papadimitriou and J.N. Tsitsiklis. The complexity of optimal queueing network control. In *Structure in Complexity Theory Conference, Proceedings of the Ninth Annual*, pages 318–322. IEEE, 2002.
[30] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, August 1994.
[31] L. L. Peterson and B. S. Davie. *Computer networks: a systems approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2007.
[32] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.
[33] Y. Sankarasubramaniam, Ö.B. Akan, and I.F. Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 177–188. ACM, 2003.
[34] Solutions For Businesses. *www.arrowspan.com/solutions/business.html*.
[35] T. Stathopoulos, M. Lukac, D. Mclntire, J. Heidemann, D. Estrin, and W. J. Kaiser. End-to-end routing for dual-radio sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2252–2260. IEEE, 2007.
[36] W. Stevens. Rfc 2001: Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. January 1997.
[37] D.W. Stroock. *An introduction to Markov processes*. Springer, 2005.
[38] K. Sundaresan, V. Anantharaman, H.Y. Hsieh, and R. Sivakumar. ATP: A reliable transport protocol for ad hoc networks. *IEEE transactions on mobile computing*, pages 588–603, 2005.
[39] A. Tonnesen, T. Lopatic, H. Gredler, B. Petrovitsch, A. Kaplan, and SO Tcke. Olsrd: an ad hoc wireless mesh routing daemon, 2008.
[40] R.R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.
[41] P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298, 1988.