An MDP-based Approach for Multipath Data Transmission over Wireless Networks

Vinh Bui and Weiping Zhu
The University of New South Wales, Australia
{v.bui, w.zhu}@adfa.edu.au

Alessio Botta and Antonio Pescapé
University of Napoli "Federico II", Italy
{a.botta, pescape}@unina.it

***Abstract*—Maintaining performance and reliability in wireless networks is a challenging task due to the nature of wireless channels. Multipath data transmission has been used in wired scenarios to reduce latency, improve throughput, and - when/where possible - balance the load. In this paper, we propose an approach for multipath data transmission over wireless networks. We demonstrate that the problem under study can be formulated as a *Markov Decision Process* (MDP) and we propose an algorithm called *On-line Policy Iteration* (OPI), to solve the formulated MDP in real time. We verified the proposed approach using simulations with ns-2 and data collected from real heterogeneous wired/wireless networks. The results indicate that we improve both delay and loss characteristics of end-to-end wireless communications outperforming the classical multi-path schemes including *Round Robin* and *Join the Shortest Queue*.**

I. Introduction

Wireless networks are noisy and unreliable communication environments. Unlike in wired networks where data losses are mainly caused by traffic congestions, losses in wireless networks are due to both traffic congestions and transmission errors. If traffic congestions are somewhat predictable, transmission errors are subject to the variation of the wireless propagation environment, and therefore much harder to predict and control. As a result, providing Quality of Service (QoS) in wireless networks is more challenging.

Recently, multipath data transmission has been proposed as a promising solution to overcome the bandwidth limitation, end-to-end delay fluctuations, and consecutive losses in the wired Internet environment [3], [6], [9], [11], [14], [21], [23], [24], [25]. Nonetheless, the proposed approaches revealed a number of limitations: (i) protocol specific; (ii) excessive computation; (iii) requirements of explicit knowledge of the network characteristics; (iv) optimal only for specific network configurations. In addition, the significant difference in data transmission characteristics of wired and wireless networks suggests the necessity of further investigation.

Therefore, in this paper we present an analytical approach to optimally transfer data over multiple wireless network paths. In particular, our contributions are the following. First, thanks to the Imbedded Markov Chain technique [15], we demonstrate that the system under study can be modeled as a Markov chain. Second, by using such results we formulate the multipath data transmission problem as a Markov Decision Process (MDP) [20], a powerful mathematical framework for making decisions in control environments exhibiting dynamic behaviors e.g. transmission errors in wireless networks. Third, we introduce a simple path state monitoring mechanism. Fourth, an algorithm called OPI (*On-line Policy Iteration*), which has a low computational complexity, is proposed to select transmission paths on the fly by using the MPD. Fifth, through simulations and real data, we show the feasibility of our approach as well as its capability to obtain better performance than classical approaches such as *Round Robin* and *Join-the-Shortest-Queue*. In addition, our approach is general enough to be applied for improving other QoS metrics such as the throughput or a function of multiple QoS parameters.

Attempts to improve QoS in wireless networks by means of multipath data transmission were made by [5] and [8]. In [5], the authors proposed a multipath routing mechanism, which adaptively searches for available network paths from a source to a destination, and optimally routes traffic among the paths. However in [5], the routing problem was formulated as a static optimization problem, using the average value of some network parameters. On the contrary, we formulate the problem using a dynamic optimization framework, which is more capable of reflecting the network dynamics. In [8], a Join the Shortest Queue (JSQ)-like algorithm was used to reduce the cost of data alignment at the receiver side while spreading traffic among multiple TCP connections. Meanwhile in this paper, we focus on reducing end-to-end delay fluctuations and improving packet loss characteristics while showing that our approach performs better than the JSQ algorithm in various simulation scenarios. Furthermore, compared to [8], we are not tied to a specific transport protocol.

The rest of the paper is organized as follows. In Section II and III, the problem under study is defined and the system model is established. The formulation of the MDP is presented in Section IV while Section V details a novel algorithm to solve the formulated MDP. In Section VI, the numerical verification of the proposed approach is presented. Finally, concluding remarks are made in Section VII.

II. Problem Definition

Consider the data transmission between two wireless end points using $M, (M > 1)$ independent paths. Fig. 1 illustrates the conceptual scenario with $M = 2$. The *multipath traffic*

[0]This work is supported by University of New South Wales at Australian Defence Force Academy. This work has been partially supported by PRIN RECIPE and CONTENT NoE, OneLab and NETQOS EU projects.

distributor is an abstract module (see Section VI-B1 for a proof of concept scenario), which is implemented to optimally distribute traffic over the paths.

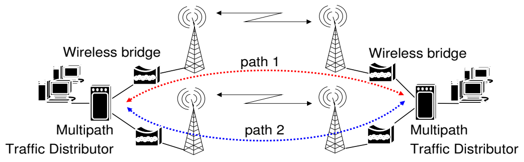

Fig. 1. Multipath data transmission with 2 independent paths.

Assume that data arrives at the traffic distributor in a stream of constant-length bins (i.e packets)[1], with an average rate of λ. Since the arrival process is discrete, we approximate it using a Poisson process [12] with inter-arrival time between consecutive bins following an exponential i.i.d. random variable.[2]

The traffic distributor has to decide the path over which to send the bins according to a predefined policy. We are interested in finding a policy, which is optimal in terms of minimizing the average transmission delay and loss rate. Since delays and losses are dynamic characteristics, which exhibit randomness to some degrees, static distribution policies are not suitable. A powerful framework to cope with such stochastic optimization problems is Markov Decision Processes (MDPs) [1]. Nonetheless, before the framework can be applied, the problem under study has to be formulated as a MDP.

III. System and States

In this section, we explain what do we mean by the system states, and how do we monitor them. We also demonstrate the system under study at specific instances can be modeled as a Markov chain where the MDP framework is applicable.

A. System Model: basic idea

Previous studies on similar problems ([13], [18]) when applying decision frameworks, often assume the perfect knowledge of the network path characteristics e.g. loss, delay and throughput. To avoid this assumption, we design a path state monitoring mechanism, which reveals loss and delay conditions of the path by keeping track of data bins being transmitted. The mechanism works as follows. Before the distributor sends a bin over the selected path (i.e. it delivers the bin to the related Access Point), the bin ID and a time stamp are recorded on a list of size K. This information is kept until the bin is correctly received by the other end of the wireless link (i.e. the other Access Point). In this manner, given the data stream, the evolution of the number of bins on the list will implicitly reflect the path state in terms of Round Trip Time (RTT) and loss. In a real implementation, the information regarding the transmission success can be retrieved directly from the Access Point by using protocols e.g. CAPWAP [7]. At a first glance, this can be seen as a considerable overhead. However we have verified that, even if this information is obtained every 10/15 bins, the results presented in Section VI remain almost unchanged (here, due to the space constraints, we do not present this analysis).

From the traffic distributor viewpoint it is easy to see that by using the path state monitoring mechanism, each path is modeled as a single server queue. The inter-departure intervals between "customers" in the queue are characterized by the distribution of RTT jitter[3]. Since each path is a single server queue, the system under study is a set of M parallel queues (thanks to the monitoring mechanism the buffer can be dynamically allocated in order to avoid overflows). The system states are created by vectors of "customers" queuing in the system. Fig. 2 depicts the system model.

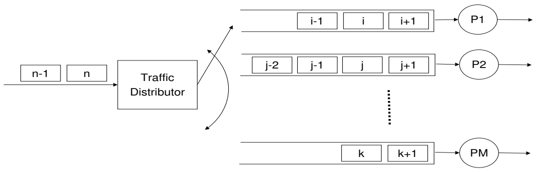

Fig. 2. The system model with M independent paths.

B. Imbedded Markov Chain

At the time instance when a data bin arrives at the traffic distributor, a decision has to be made on where to send the bin. Depending on the decision, the system may change its state in accordance. Therefore, the system behaviors observed at the arrival instances are important for the decision maker. To study these behaviors, the method of Imbedded Markov Chain [15] in combination with renewal theory [4] is applied.

Let t_i be the arrival instance of bin i at the *traffic distributor*. In accordance, let $Q_m(t_i)$ be the number of bins found in the m-th queue at t_i. Consequently, the system state at t_i is a vector $\{Q_m(t_i)\}, (m = 1 \ldots M)$. We are interested in the evolution of $\{Q_m(t_i)\}, (i = 1, 2, 3 \ldots \infty)$ under the impact of the scheduling actions taken at the traffic distributor.

Assume that, at arrival instance t_i, the bin i is sent to the $m-$th queue by the decision maker. To eliminate the possible transitive period and simplify the discussion, we assume a decision and its action can be completed instantly. As a consequence, the system will move immediately from state $\{Q_1(t_i), Q_2(t_i), \ldots, Q_m(t_i), \ldots, Q_M(t_i)\}$ to state $\{Q_1(t_i), Q_2(t_i), \ldots, [Q_m(t_i) \leftarrow Q_m(t_i) + 1], \ldots, Q_M(t_i)\}$.

From this moment, until the arrival of the next bin, the evolution of $\{Q_m(t_i)\}$ depends only on the number of bins departing from each queue during $[t_{i+1} - t_i]$ interval. In other words, if $P_{QQ'}$ denotes the probability the system is in state $\{Q'_m(t_{i+1})\}$ at the arrival instance of bin $(i+1)$, then

[1] The constant length is assumed to simplify the formulation of the model which can be, however, easily generalized for variable sized packets.

[2] In general, any discrete arrival process, e.g. packet streams, can be approximated using the Poisson arrival process (see Faddy MJ (1997a) *Extended Poisson process modelling and analysis of count data* for example). Nonetheless, our arrival process is not purely Poisson but Markov Modulated Poisson like. The reason is we consider the arrival process is Poisson(λ) only in a short period of time. After this period, λ may change to another value.

[3] RTT jitter is defined as the absolute difference between two consecutive RTT values i.e. $Jitter_i = RTT_{i+1} - RTT_i$.

$P_{QQ'}$ equals to the probability there are $Q_1(t_i) - Q'_1(t_{i+1})$ bins departing from the first queue, $Q_2(t_i) - Q'_2(t_{i+1})$ bins departing from the second queue, and so on during $[t_{i+1} - t_i]$ interval. Since these departure processes are independent, we can obtain $P_{QQ'}$ by taking product of the probabilities.

To compute the probability there are $Q_m(t_i) - Q'_m(t_{i+1})$ bins departing from the m-th queue, we utilize some results from renewal theory. Let τ_k and τ_{k+1} subsequently denote the departure instances of bins k and $k+1$ from the m-th queue. Assuming that, the inter-departure intervals $[\tau_{k+1} - \tau_k]$ are i.i.d. random variables[4] with a general distribution $g_m(x) = P(\tau_{k+1} - \tau_k \leq x)$. As a consequence, the sequence of departure points $\{\tau_k\}$ fully forms an *ordinary* renewal process.

Suppose the queuing process has attained a steady state. Subsequently, the sequence of departure points $\{\tau_k\}$, started from the arrival instance t_i, forms the *equilibrium* renewal process of the above ordinary renewal process [4] (Fig. 3).

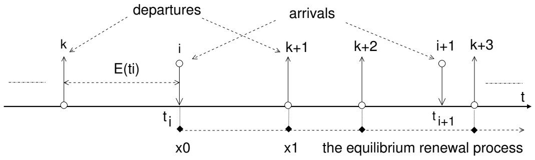

Fig. 3. The queuing process of the m-th queue.

According to renewal theory, for the *equilibrium* renewal process, the number of bins departing from the queue in $[t_{i+1} - t_i]$ interval is proportional to the length of the interval and does not depend on t_i. Hence, $Q_m(t_{i+1})$ is fully determined by $Q_m(t_i)$ and does not depend on $E(t_i)$, which is the elapsed time since the last departure observed at t_i. Thus, the random variable $Q_m(t_i)$ forms a discrete-state Markov chain imbedded in the queuing process. The state space of the chain is comprised of all possible numbers of bins in the queue.

Let β_j denote the probability of j bins, $j = 0 \ldots K$, departing from the queue in $[t_{i+1} - t_i]$ interval. Since by assumption $[t_{i+1} - t_i]$ are exponentially distributed, we have:

$$\beta_j = \int_0^\infty P[N(t) = j] \lambda e^{-\lambda t} dt \qquad (1)$$

Recall that the inter-departure intervals $[\tau_{k+1} - \tau_k]$ are i.i.d. random variables with a general PDF $g_m(x)$. Let $G_m(x)$ be the corresponding cumulative distribution function of the inter-departure intervals, and $G_m^{(n)}(x)$ be the n-fold convolution of $G_m(x)$. Let $\mu_m \equiv E[\tau_{k+1} - \tau_k]$ be the mean value of the inter-departure times. Subsequently, as shown in [4], the probability of having j renewals during interval $(0, t)$ of the equilibrium renewal process $P[N(t) = j]$ is:

$$P[N(t) = j] = \frac{1}{\mu} \int_0^t (P[N^o(u) = j-1] - P[N^o(u) = j]) du$$

$$= \frac{1}{\mu} \int_0^t [(G_m^{(j-1)}(u) - G_m^{(j)}(u)) - (G_m^{(j)}(u) - G_m^{(j+1)}(u))] du$$

where $P[N^o(u) = r] = G_m^{(r)}(u) - G_m^{(r+1)}(u)$ is the probability of having r renewals during interval $(0, u)$ of the corresponding *ordinary* renewal process. Substituting received $P[N(t) = j]$ to (1), we can compute β_j.

As mentioned, the random variable $Q_m(t_i)$ forms a discrete-state Markov chain in which the state transitions occur only at the arrival instances. As a result, vectors $\{Q_m(t_i)\}, m = 1 \ldots M$, which are the system states also form a discrete-state discrete-time Markov chain. The chain is illustrated in Fig. 4.

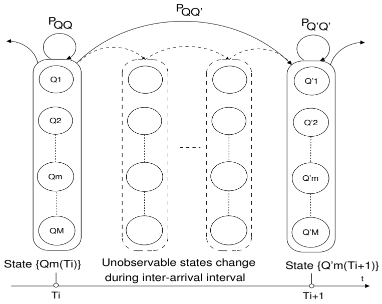

Fig. 4. The imbedded Markov chain.

C. Transition Probability Matrix Computational Issues

Although we are able to obtain the transition probabilities matrix of the combined Markov chain in the general form, it is challenging to compute the matrix with inter-departure intervals following a general distribution $g(x)$. Several approaches are available to cope with the transition matrix computational difficulty including simulation (e.g Monte-Carlo), reinforcement learning (e.g. Q-learning), and function approximation. Due to the space constraint, here we present only the function approximation approach. In particular, we approximate $g(x)$ by the a-stage Erlang distribution with rate ρ. In Section VI-A, we will show that the approximation is acceptable by studying RTT jitter of real heterogeneous networks.

By approximation:

$$g(x) \approx \frac{\rho^a x^{a-1} e^{-\rho x}}{(a-1)!}$$

Since:

$$\int_0^t \frac{\rho^{m+1} u^m e^{-\rho u}}{m!} du = 1 - \sum_{n=0}^{m} \frac{(\rho t)^n e^{-\rho t}}{n!}$$

We can compute $P[N(t) = j]$ directly:

$$P[N(t) = j] = \frac{\rho}{a} \int_0^t \sum_{m=ja-a}^{ja-1} - \sum_{m=ja}^{ja+a-1} \frac{(\rho u)^m e^{-\rho u}}{m!} du$$

$$= \frac{1}{a} \sum_{n=ja}^{ja+a-1} (ja + a - n) \frac{(\rho t)^n e^{-\rho t}}{n!}$$

$$+ \frac{1}{a} \sum_{n=ja-a}^{ja-1} (n - ja + a) \frac{(\rho t)^n e^{-\rho t}}{n!}$$

Substituting received $P[N(t) = j]$ to (1), we get:

[4] we only assume i.i.d in a period of time since Internet Path Properties e.g. delays, throughputs are stationary in scale of minutes [27].

$$\beta_j = \int_0^\infty P[N(t) = j]\lambda e^{-\lambda t} dt$$
$$= \frac{1}{a} \sum_{n=ja}^{ja+a-1} \frac{(ja + a - n)(\frac{\rho}{\lambda})^n}{(1 + \frac{\rho}{\lambda})^{n+1}} + \sum_{n=ja-a}^{ja-1} \frac{(n - ja + a)(\frac{\rho}{\lambda})^n}{(1 + \frac{\rho}{\lambda})^{n+1}}$$

Given ρ/λ ratio[5], we can be compute β_j directly by using the above formula. Obtaining β_j for each queue, we can compute the transition probability matrix of the Markov chain.

IV. MDP Formulation for Multipath Transmissions

In previous section we demonstrated that the system under study can be modeled as a Markov chain. Now, we show how such a problem is formulated as a MDP.

Consider a time homogeneous MDP with a countable state space S, a finite action space A where $A(s) \in A, s \in S$ is a set of admissible actions in state s, a non-negative immediate reward function $R : S \times A(S) \rightarrow \Re_+$, and a set of conditional probabilities $P(s'|s, a)$, which is the probability of moving from state s to state s' if action $a, a \in A(s)$ is taken. In context of the problem under study, the MDP objective is to minimize the average reward, which is a function of the path delay/loss states over a period of time. The four-component tuple $\{S, A, P, R\}$ of the MDP is subsequently made of:

- ***S*** *which is the state space of the defined system consisting of* $M, (M \geq 1)$ *paths. It is clear that* S *is the state space of the combined Markov chain.*
- ***A*** $= \{a_1, a_2 \ldots a_M\}$ *is the set of actions each of which corresponds to the action of scheduling a data bin to one of* M *paths.*
- ***P*** *is the state transition probability matrix of the combined Markov chain, which can be calculated from the transition probabilities of each path in the system.*
- ***R*** *is the immediate reward function, which should be able to reflect the minimum average transmission loss/delay objective. Thanks to the path monitoring mechanism, the number of queuing bins on each path is a proportional function of the path delay/loss states [17]. Hence, we define the immediate reward function* $R(s, a) = (\mu_a s + d_a)(1 - p^{loss}_{s,a})$ *where* μ_a *is the common inter-departure interval of the path chosen by action* a*;* $s \in \{1, 2 \ldots K\}$ *is the state of the path;* d_a *is the path propagation delay; and* $p^{loss}_{s,a}$ *is the probability the packet is lost if the path is in state* s*. In practice,* d_a *can be estimated as a half of the minimum RTT delay, and* $p^{loss}_{s,a}$ *can be estimated from the historical data using the frequency count technique.*

A Markov policy is a description of behaviors, which specifies the action to be taken in correspondence to each system state and time step. If a policy is *stationary*, it specifies only the action to be taken in each state independently of the time steps. Given policy π and initial state s of the system, one can quantitatively evaluate π based on the expected long-term average reward, which is defined as follows:

$$v^\pi(s) \equiv \limsup_{N\rightarrow\infty} \frac{1}{N} \sum_{t=1}^{N} E^\pi_s \quad R(S(t), a_\pi(S(t))) \qquad (2)$$

where $R(S(t), a_\pi(S(t))$ denotes the immediate expected reward received by the decision maker at time t, when taking action $a_\pi(S(t))$ in accordance with the policy π, while the system is in state $S(t)$. An optimal policy is the one that minimizes the average reward v^π. Note that, it is sufficient to find an optimal policy in the Markov policy space since, for any history-dependent policy there is a Markov policy that yields the same average reward [20]. In unconstrained MDP formulation, it is possible to find a stationary optimal policy.

V. Optimal Data Distribution Algorithm

Optimal data distribution can be achieved by solving the formulated MDP. To obtain the MDP optimal policies, different approaches can be used e.g. dynamic programming and linear programming. However, we pay more attention on the solving time since the problem under study is time sensitive and we are interested, as next step of this work, in building a real implementation. Therefore, we implement an algorithm called *On-line Policy Iteration (OPI)* to gradually approach the optimal policy of the formulated MDP on the fly.

A. Scheduling with On-line Policy Iteration

On-line Policy Iteration is based on the idea of the Policy Iteration (PI) algorithm [20] and asymmetric dynamic programming. Two key factors, which potentially make OPI a preferable algorithm for time sensitive problems are the incremental on-line policy improvement and the computational efficiency. The first factor comes naturally since OPI is based on the PI algorithm, which iteratively improves a given random policy until an ϵ-optimal policy is found. The second factor however is resulted from a smart choice of the starting policy and the improvement strategy. Particularly, instead of choosing a random policy for improvement, like in the PI algorithm, we select the Join the Shortest Queue (JSQ) policy to be the starting policy for OPI. Since the JSQ policy is an optimal policy for 2 parallel symmetric queues and an acceptable policy for several other cases [16], we can save a lot of iterations by starting from it, on the way to reach an ϵ-optimal policy. The policy improvement strategy also plays an important role in saving computational efforts. It is common in practice that some states of a system are more frequently visited than the others. These states are usually more important to the decision maker as they contribute more to the final outcome. Therefore, instead of equally evaluating and improving the policy for every state, as in Policy Iteration algorithm, we could spend more time on evaluating and improving the policy for those important states. This policy improvement strategy is known as prioritized sweeping, which still guarantees an ϵ-optimal policy will be found if all system states are visited infinitely often [20]. The OPI algorithm is detailed below:

- ***Step 1****: Evaluate JSQ policy to obtain* $v^{opi}(s) = v^{jsq}(s)$.
- ***Step 2****: Obtain the system current state* s*. Take action* $a \in A$ *that maximizes* $v^{opi}(s)$ *and update* $v^{opi}(s)$ *with the maximum value:*

$$v^{opi}(s) = \max_{a\in A}\{R(s, a) + \sum_{s'\in S} p_a(s'|s)v^{opi}(s')\} \qquad (3)$$

[5] In practical implementation, ρ/λ ratio can be periodically estimated on the fly from the historical evolution of the path queue lengths.

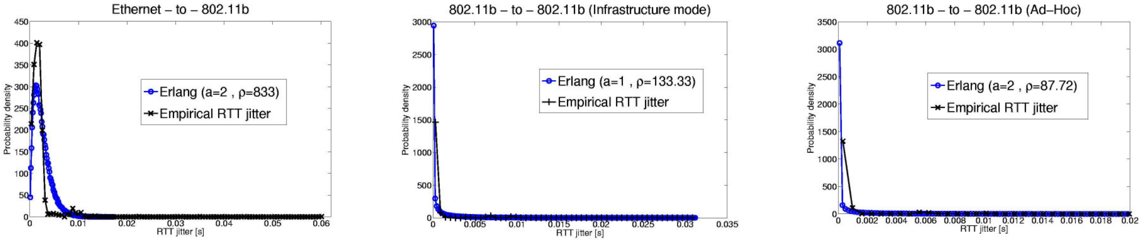

Fig. 5. RTT Jitter and Erlang distributions for each path.

where $p_a(s'|s)$ denotes the transition probability from state s to state s' if action a is taken.

- ***Step 3**: If there is a decision to be made, i.e. more data to send, return to Step 2. Otherwise, stop.*

The implementation of OPI is straightforward. The decision maker does not have to wait until the algorithm finds an ϵ-optimal policy. Instead, it can start immediately with the JSQ policy, and the JSQ policy will be improved on-line after each decision epoch. For many cases, the JSQ policy will become ϵ-optimal after few iterations, e.g. 10 to 15.

B. Complexity Analysis

Since the most computationally expensive step of OPI (the evaluation of JSQ policy) can be done offline, immediately whenever the system model is available, the OPI algorithm is lightweight and can be used for making decisions on the fly. In specific, we only have to compute Eq. (3), the complexity of which is $O(N)$, $(N = sizeof(S))$. In practice, N is usually in the range of thousands, depending on K and M.

VI. Numerical Study

In Section III we demonstrated that the problem under study can be modeled as a Markov chain. Afterward, in Section IV we have shown how to apply the MDP to such problem, and in Section V we presented the OPI algorithm to solve it on the fly. In this section, (i) we first provide experimental evidences of an assumption made in Section III-C; then, (ii) we investigate the behavior of important QoS parameters when the OPI is adopted; and finally, (iii) we compare the performance of OPI with those of the other scheduling algorithms.

A. RTT Jitter Distribution

This section analyzes the distribution of RTT jitters measured over different real wireless network paths. This analysis is meant to provide an experimental proof of the thesis, previously argued in Section III-C, for which such jitters are well approximated by an Erlang distribution. The network scenarios on which the data was collected are part of a real heterogeneous testbed (see [2] for more details). To show our claim, here we consider the following three network paths: (i) Ethernet-to-802.11b; (ii) 802.11b-to-802.11b (infrastructure mode); (iii) 802.11b-to-802.11b (ad-hoc mode). Taking into account RTT jitter related to different paths with different characteristics allows to verify our assumption over several network scenarios making it more general.

Fig. 5 shows the RTT jitter distribution for the above mentioned network scenarios. In all the cases the Erlang distribution obtains a good fitting. To provide a numerical evidence, we have evaluated a discrepancy measure (by using the λ^2 [22]) between the RTT jitter and three analytical distributions: Erlang, Normal, and Weibull. In details, in the first case the three distributions obtained 0.44, 1.24, and 0.54 respectively. In the second case, they obtained 0.20, 5.15, and 0.74 respectively, and in the third 0.25, 3.89, and 0.48. In conclusion, thanks to this preliminary analysis, we can state that, in all the three cases, the Erlang distribution provides the best approximation for the RTT jitter distribution.

B. Performance Evaluation

*1) **Simulation Setup**:* To evaluate the proposed approach, we setup a multipath wireless communication scenario in which two wireless end points exchange data over - respectively - 2, 3, and 4 paths while moving in parallel in a [1500m x 1500m] flat grid area, with speed of $10m/s$. The scenario is illustrated in Fig. 6. With respect to Figure 1, here the *traffic distributor module* is integrated into the end points.

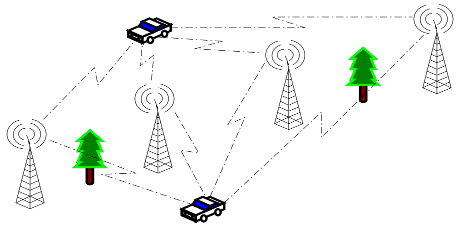

Fig. 6. Multipath wireless communication: 4-path scenario.

Each path is comprised of 3 wireless nodes (2 hosts and an Access Point), which implement Ricean radio propagation model [19], 802.11 MAC with the basic rate of 1Mbps, $600m$ of transmission range, and supporting Dynamic Source Routing Protocol [10]. To create disjoint wireless paths, different communication channels are used. Since in ns-2 the channels are orthogonal, the interference effect by default is not considered. Therefore, to simulate transmission errors on 802.11 links more appropriately, we implement a modification of the 802.11 error model proposed by Wu [26] in conjunction with the Ricean fading propagation model. The two end nodes exchange data in a 500-byte packet stream at the average rates of 2.0Mbps to 4.0Mbps depending on the number of paths in use. The stream of packets arrives at the traffic distributor like in a Poisson process. The packets are then routed to the paths for transmission using 3 algorithms: OPI, RR, and JSQ.

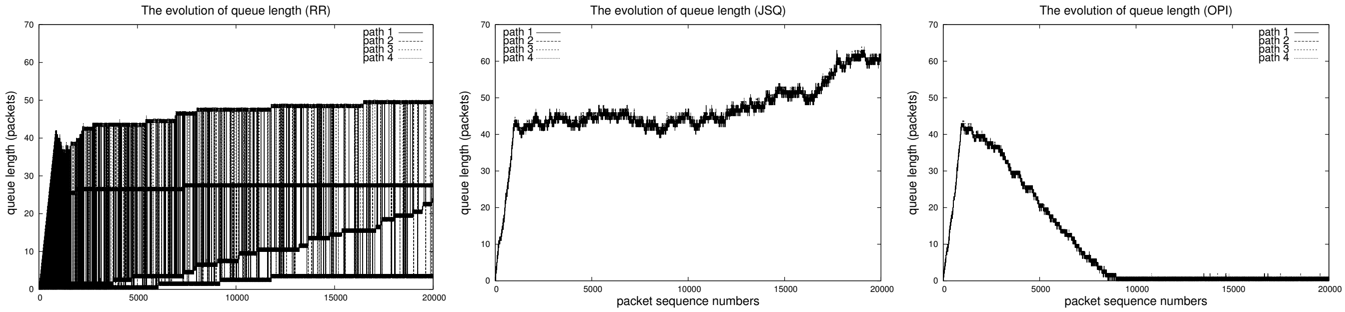

Fig. 7. The average path queue lengths obtained with *RR*, *JSQ* and *OPI* (4-path scenario).

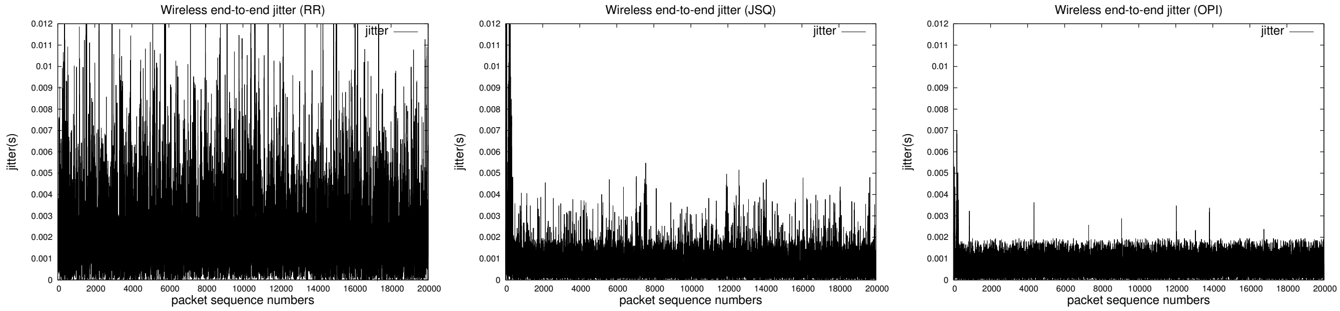

Fig. 8. Jitter obtained with *RR (mean: 2.1ms, stdev: 2.0ms)*, *JSQ (mean: 0.8ms, stdev: 0.65ms)* and *OPI (mean: 0.67ms, stdev: 0.47ms)*, over 3 paths.

2) ***Performance Comparison:*** We compare the performance of the scheduling algorithms in terms of average end-to-end delay, jitter, loss rate and distribution of consecutive losses while transferring 10MB of data between the two end nodes. We vary the distance and speed of the two end nodes to create 3 different loss scenarios, in which the average loss rates (experienced by the RR algorithm) approximately are 1.0%, 3.0% and 6.0%. Due to the space constraint, we cannot present here the simulation results of all scenarios we have considered. Hence, only the representative results of the 6.0% average loss rate scenario are presented. Table I shows the average, and variance in parenthesis, of delays and loss rates obtained by the algorithms with 2, 3 and 4 paths.

TABLE I
AVERAGE AND VARIANCE (IN PARENTHESIS) OF DELAY AND LOSS RATE.

	2 paths		3 paths		4 paths	
	D(ms)	L(%)	D(ms)	L(%)	D(ms)	L(%)
RR	7.7(2.4)	6.0(0.1)	7.4(2.3)	5.8(0.1)	7.6(2.1)	5.9(0.1)
JSQ	8.3(0.9)	6.2(0.1)	8.1(0.7)	6.5(0.1)	8.2(0.7)	6.1(0.1)
OPI	7.2(0.6)	5.9(0.1)	6.7(0.5)	4.7(0.1)	5.9(0.5)	4.2(0.1)

As shown in Table I the OPI obtains the best performance. Compared to JSQ and RR, OPI achieved smaller average end-to-end delay and loss rate in all considered scenarios. A good explanation for these results is in Fig. 7, which illustrates the evolution of the path queue lengths of the three algorithms obtained with 4 paths. As shown, the fluctuations of the RR queues indicates the dynamics of the wireless channels, which reflect both congestion and transmission errors. These fluctuations are resulted in high end-to-end delay variance and jitter (shown in Fig. 8 - left). Compared to RR, the JSQ managed to maintain a stable and equal queue length for all paths. However, with this algorithm the queue lengths are high, which subsequently yields the high average end-to-end delay and loss rate. Anyway, in comparison to those obtained by RR, the JSQ jitter is lower (Fig. 8 - center). As for the evolution of the OPI queues, (Fig. 7-right) clearly shows the online policy improvement process. In the warm up period (approximately first 500Kb of data), the evolution of the OPI queues is almost the same as the evolution of the JSQ queues since the two policies are in fact the same. However, after the OPI obtained the optimal policy, its queue lengths start to decrease. As a result, the OPI gives the lowest average end-to-end delay, with the lowest variance as well as more stable jitter (Fig. 8 - right).

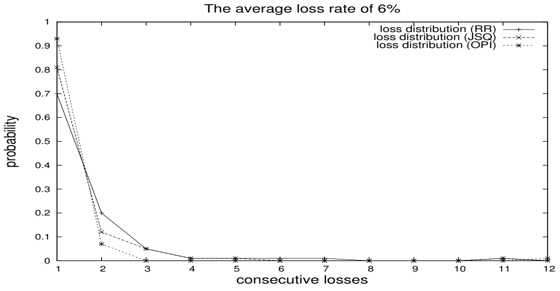

Fig. 9. Probability of consecutive losses with *RR*, *JSQ* and *OPI* (3-paths).

Besides the end-to-end delay characteristics, the simulation results also indicate that the OPI algorithm improves end-to-end loss characteristics. Fig. 9 shows the distribution of consecutive losses obtained with OPI in comparison with those obtained by RR and JSQ. As shown, the probability of suffering from 2 consecutive losses is 0.07 for OPI while this probability is subsequently 0.12 and 0.20 for JSQ and RR. The probability of suffering from more than 2 consecutive losses is 0.01 for OPI, 0.07 for JSQ and 0.10 for RR.

Although our approach is built to pursue the improvement of delay and loss characteristics, it is also interesting to look at the behavior of another relevant QoS parameter such the throughput. Fig. 10 shows the average throughput obtained by the algorithms using 2 paths with the average data rate of 2Mbps. We can see the impact of the fading effect on the JSQ algorithm, which gives a low and fluctuating throughput. Due to the fading effect, throughput of the wireless channels changes rapidly. As a result, greedy strategies like JSQ suffer heavily from these rapid variations. On the other hand, the OPI algorithm, which "learns" from the experience, obtains a significantly better throughput. The RR algorithm, which distributes packets equally to all paths also obtain a good throughput. However, it suffers more losses and high end-to-end delay variations since it does not take the channels dynamics into account. Finally, we can conclude that OPI reaches the better performance with respect all QoS parameters.

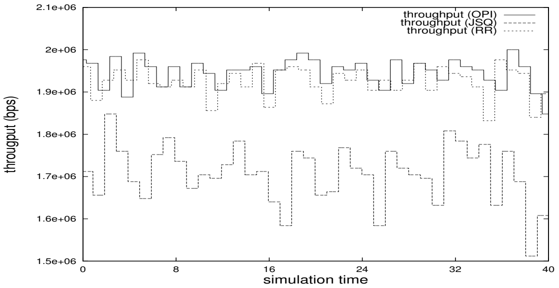

Fig. 10. *RR*, *JSQ* and *OPI* throughput computed in 1s interval (2 paths).

VII. Conclusion

In this paper we have proposed an approach to distribute data over multiple wireless paths. In details, we designed a simple path monitoring mechanism to obtain the loss and delay information on each individual path. Moreover, by exploiting the Imbedded Markov Chain, we formulated the multipath data transmission problem as a MDP and an algorithm called OPI was introduced to solve the MDP on the fly. We verified the proposed approach in a proof-of-concept network scenario, obtaining encouraging results from ns-2 simulations and using real data from operational wired/wireless networks. The preliminary evaluation has shown the feasibility of the proposed approach as well as a performance gain in terms of both end-to-end delay and loss characteristics with respect to classical schemes such as *Round Robin* and *Join the Shortest Queue*. In future works, we will deeply investigate preliminary results obtained in this study, and we will extend the approach to cover the problems with multiple QoS objectives, e.g. to minimize both loss rates and average transmission delay, while bounding transmission powers and/or load balancing.

References

[1] E. Altman, "Applications of markov decision processes in communication networks," in *Handbook of Markov Decision Processes: Methods and Applications*, E. A. Feinberg and A. Shwartz, Eds. Norwell, Massachusetts, USA: Kluwer, 2002.

[2] A. Botta, A. Pescapé, and G. Ventre, "Identification of network bricks in heterogeneous scenarios," in *IEEE Conference on Local Computer Networks (LCN)*, Nov. 2006, pp. 675–682.

[3] C. Cetinkaya and E. W. Knightly, "Opportunistic traffic scheduling over multiple network paths," in *Proc. IEEE INFOCOM*, vol. 3, Hong Kong, Mar. 2004, pp. 1928–1937.

[4] D. R. Cox, *Renewal Theory*. London, GB: Methuen & Co. Ltd, 1962.

[5] S. K. Das, A. Mukherjee, S. Bandyopadhyay, K. Paul, and D. Saha, "Improving quality-of-service in ad hoc wireless networks withadaptive multi-path routing," in *Proc. IEEE GLOBECOM*, San Francisco, CA, USA, Nov. 2000.

[6] L. Golubchik, J. Lui, T. Tung, A. Chow, W. Lee, G. Franceschinis, and C. Anglano, "Multi-path continuous media streaming: What are the benefits?" *Elsevier Performance Evaluation*, vol. 49, no. 1-4, pp. 429–449, Sept. 2002.

[7] S. Govindan, H. Cheng, Z. Yao, W. Zhou, and L. Yang. Objectives for control and provisioning of wireless access points (capwap). rfc4564.txt.

[8] Y. Hasegawa and et al, "Improved data distribution for multipath tcp communication," in *Proc. IEEE GLOBECOM*, vol. 1, Sant Louis, MA, USA, Dec. 2005, pp. 271–275.

[9] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using sctp multihoming over independent end-to-end paths," *IEEE/ACM Trans. Networking*, vol. 14, no. 5, pp. 951–964, Oct. 2006.

[10] D. Johnson, D. Maltz, and J. Broch, *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*.

[11] D. Jurca and P. Frossard, "Video packet selection and scheduling for multipath streaming," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 629–641, Apr. 2007.

[12] T. Karagiannis, M.Molle, M.Falautsos, and A.Broido, "A nonstationary poisson view of internet traffic;," in *Proc. of the Infocom 2004*.

[13] A. K. Karmokar and V. K. Bhargava, "Optimal packet scheduling over correlated nakagami-m fading channels with different diversity-combining techniques," in *Proc. IEEE GLOBECOM*, Sant Louis, MA, USA, Dec. 2005, pp. 1217–1222.

[14] R. Karrer and T. Gross, "Multipath streaming in best-effort networks," in *IEEE International Conference on Communications (ICC)*, vol. 2, May 2003, pp. 901–907.

[15] L. Kleinrock, *Queueing Systems Volume 1: Theory*. New York, USA: John Wiley & Son, 1975.

[16] J. Y.-T. Leung, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. USA: Chapman and Hall/CRC, 2004.

[17] J. Little, "A proof of the queueing formula $l = \lambda w$," *Operations Research*, vol. 9, pp. 383–387, 1961.

[18] X. Liu and et al., "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 2053–2064, Oct. 2001.

[19] R. J. Punnoose and al. (2007, Aug.) Additions to the ns network simulator to handle ricean and rayleigh fading. [Online]. Available: http://www.ece.cmu.edu/wireless/

[20] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA: John Wiley and Sons, 1994.

[21] K. Rojviboonchai, F. Yang, Q. Zhang, H. Aida, and W. Zhu, "Amtp: A multipath multimedia streaming protocol for mobile ad hoc networks," in *IEEE International Conference on Communications (ICC)*, vol. 2, May 2005, pp. 1246–1250.

[22] M. J. S. Pederson and, "Estimating model discrepancy," in *Technometrics, 32(3)*, Aug. 1990, pp. 305–314.

[23] S. Vutukury and J. J. Garcia-Luna-Aceves, "Mpath: A loop-free multipath routing algorithm," *Elsevier Journal of Microprocessors and Microsystems*, vol. 24, pp. 319–327, 2000.

[24] B. Wang, J. Kurose, D. Towsley, and W. Wei, "Multipath overlay data transfer," University of Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. CMPSCI-TR05-45, 2005.

[25] B. Wang, W. Wei, Z. Guo, and D. Towsley, "Multipath live streaming via tcp: Performance and benefits," University of Connecticut, Storrs, CT, USA, Tech. Rep. BECAT/CSE-TR-06-7, 2006.

[26] W. Xiuchao. (2007, Aug.) Simulate 802.11b channel within ns2. Progress. [Online]. Available: http://www.comp.nus.edu.sg/~wuxiucha/research/reactive/

[27] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of internet path properties," in *Proc. of the 1st ACM SIGCOMM Workshop on Internet Measurement*, USA, Nov. 2001, pp. 197–211.