Abstract—In this paper, we study the problem of optimal data transfer over multiple overlay paths. Instead of solving the problem from the single controller point of view, we adopt the game theory perspective to consider the problem from a more realistic view where multiple traffic controllers competing for the shared bandwidth. We formulate the problem as a general-sum stochastic game, and a reinforcement learning technique namely Correlated-Q Learning is implemented to derive the best-possible strategy, i.e. the strategy to play Correlated Equilibrium (CE) for each controller. Through a proof-of-concept simulation scenario with 2 overlay paths and 2 controllers, we show that by playing cooperative strategies, e.g. CE, the controllers can achieve superior performance compared to acting selfishly. The result emphasizes that considering the problem of optimal multipath data transfer from the single controller perspective is inadequate.

I. INTRODUCTION

Despite a continuous improvement in the network infrastructure, providing quality assured data transfer in the best-effort Internet remains a challenging task due to limited bandwidths, highly varying delay and losses. Recent studies have demonstrated the benefits of multipath data transfer in obtaining high available bandwidth, better loss patterns, and bounded delay in the best-effort Internet environment [6], [13], [14], [20], [23]. However, most of the previous studies considered the problem from the single controller perspective, i.e. only one “smart” controller competing with the underlying traffic for resources. Nonetheless, as the paths are shared between many users and applications, there can be multiple smart controllers, one for an application, aiming at providing the best service for their users. As a consequence, how many resources a controller can allocate depends not only on its own decisions but also on decisions of other controllers. In this case, what strategy does each controller use to obtain the best performance. Can they act selfishly or do they need to cooperate? In this paper, we show that non-cooperative controllers may suffer significant performance losses by making selfish decisions. In contrast, by cooperating with each other, the controllers can achieve a far better performance while maximizing the utilization of network resources. Thus, we emphasize that considering the problem of optimal multipath data transfer from the single controller perspective is not adequate, and further study on the problem is necessary.

In this paper, the multipath data transfer problem is studied in the overlay network environment. Due to a number of advantages, e.g. deployability, low cost, flexibility, and TCP/IP friendliness, application-level overlay networks have been recently used as a feasible yet effective architecture to establish multipath environments over the Internet [2], [3]. As we examine the problem from the multiple controllers perspective, game theory [18] is a natural framework to tackle the problem. We model each multipath traffic controller as a Markov Decision Process (MDP) [19] that makes decisions based on the paths congestion states to minimize the transmission delay and losses. (The details of developing the MDP-based multipath controllers are referred to [5]). Consequently, a general-sum stochastic game is established between the controllers competing for available bandwidths on the shared paths. To answer the question about strategies the controllers should follow, we study performance of the controllers in a [2x2] game, i.e. 2 multipath controllers competing on 2 disjoint paths. More specifically, we apply a reinforcement learning technique called Correlated-Q Learning [9] to guide the controllers to playing a cooperative game. The performance results are then compared with those obtained in the non-cooperative case.

To briefly summarize the contributions of the paper, we underline that: (i) it indicates the necessity to study the problems of optimal multipath data transfer from the multiple controllers perspective; (ii) it applies a game theoretic framework, in particular stochastic games, and reinforcement learning to address the optimal multipath data transfer problem in the overlay network environment; (iii) through a proof-of-concept simulation scenario with 2 paths and 2 controllers, it shows the failure of non-cooperative strategies and a significant performance gained by following cooperative strategies.

The rest of the paper is organized as follows. A review of the related work is presented in Section II. In Section III, the problem under study is defined, and the system model is established. The formulation of the problem as a general-sum stochastic game is detailed in Section IV. Section V introduces the Correlated-Q Learning algorithm to obtain a cooperation between the controllers. Simulation results and discussion comparing the controllers’ performance are provided in section VI. Finally, concluding remarks are made in Section VII.
II. RELATED WORK

Attempts to design smart traffic controllers to optimally spread data over multiple overlay paths have been made by [6], [7], [8], [22]. With the minimum packet delay objective subject to some data splitting rules, the optimal multipath data transfer problem was studied in [6]. The authors proposed an opportunistic scheduling algorithm to select the transmission path. Rather than always choosing a path with the smallest expected delay, the algorithm opportunistically selected another path based on a live-measured parameter to maintain a predefined routing ratio. On the other hand, [7] and [8] solved the multipath data transfer problem with the minimum transfer time objective. The problem was formulated under the graph theory framework. However, the graph theory approach has a number of potential problems as admitted by the authors of [7]. First, the graph formulation assumes the time required to transfer a unit of data through a particular link is fixed, which is unrealistic. Second, since the capacity of a path does not fully correspond to the delay of the path, the use of path capacities as constraints to minimize the makespan may not lead to an optimal solution. Aiming at maximizing an aggregate sending rate of a source over \( k \) \((k \geq 1)\) overlay paths, the authors of [22] have developed a multipath rate controller that randomly chooses a transmission path following a throughput proportionale selection scheme. In this paper, we show that this approach may not achieve its intended goals if there are multiple “smart” controllers competing for available bandwidths on some shared links/paths. Although different approaches have been used to address various multipath data transfer problems, none of them considered the problem from a multi-controller perspective. To the best of our knowledge, we are the first to take into account the impact of multiple competitive controllers. When taking multiple competitive controllers into account, the problem becomes more challenging. Nevertheless, it also becomes more realistic where game theory is a natural framework to tackle the problem.

As we utilize a game theoretic approach, it is worth mentioning that game theory has been used to study various communication and networking problems including routing, service provisioning, admission and flow/rate controlling while formulating them as either cooperative or non-cooperative games (see [1] and references therein). However, most of the previous work focused on analyzing the existence and uniqueness of different solution concepts, e.g. Nash, Correlated, and Stackelberg equilibria in a general setting of such games, rather than obtaining strategies to play these equilibria in a specific game. Although game theoretical frameworks are powerful in describing and analyzing competitive decision problems, obtaining solutions for many types of games goes beyond the current boundary of game theory. In particular, multiagent reinforcement learning [11] is a more practical framework to obtain different equilibria, e.g. Nash [12], Stackelberg [15], and Correlated [9] in stochastic games.

In this paper, we evaluate the benefit of considering the optimal multipath data transfer problem under the framework of general-sum stochastic games. Hence, our work focuses on formulating the game and obtaining optimal solutions for the problem. Attention is paid more to obtaining Correlated Equilibrium. Although playing Nash Equilibrium is fair, it does not guarantee to give each player the best possible reward. Indeed when players are non-cooperative, playing Nash is the best-response strategy one can apply. However, if the players are cooperative, each of them may obtain a better reward.

III. PROBLEM DEFINITION

A. Problem Setting

Consider a multiuser application-level overlay network that supports multipath data transfer; and a group of users’ applications that utilize \( L \) \((L > 1)\) “smart” traffic controllers to transfer data, i.e. packet streams of multimedia contents, between sources and destinations over multiple overlay paths. Each smart controller is modeled as a Markov Decision Process (MDP) [5], which dynamically selects transmission paths among \( M \) \((M > 1)\) available overlay paths to minimize the average end-to-end delay and loss rate. Since the network is multiuser, the paths (or some links on the paths) are shared between the applications. Fig 1 illustrates a conceptual multipath multiuser application-level overlay network.

In this context, we want to investigate the impact of multiple controllers competing for available bandwidths of the shared paths on the controllers’ performance. We are also interested in deriving the best-possible strategy for each controller to achieve its designed performance goal in this situation.

B. System Model

Previous work on similar problems, (e.g. on streaming video over multiple paths [13] and on scheduling packets over multiple paths [6]), when applying decision frameworks often assumes the perfect knowledge of the path loss and delay characteristics. This information is usually difficult to obtain in practice. To avoid this assumption, we design a simple path state monitoring mechanism, which reveals loss and delay conditions of the path by keeping track of data packets being transmitted. The mechanism works as follows. Before the transmission of each packet, the packet ID and its time of transmission are recorded on a list of size \( K \). This information is kept until an acknowledgment of the successful
transmission of the packet is received or until timeout. In this manner, given the data stream, the evolution of the number of packets on the list implicitly reflects the path states in terms of Round Trip Time (RTT) delay and losses. The path state monitoring mechanism is illustrated in Fig. 2.

From the viewpoint of an individual traffic controller, by using the path state monitoring mechanism, each path is seen as a single server queue. The inter-departure intervals between “customers” in the queue are characterized by the distribution of RTT jitters\(^1\). Since each path is a single server queue, the system under study is a set of \(M\) parallel queues\(^2\). The system states are created by vectors of “customers” queueing in the system. Fig. 3 depicts the system model.

Since there are \(L_i\) (\(L > 1\)) traffic controllers competing on the overlay paths, \(L\) competitive Markov Decision Processes are formed. Each process independently makes its decisions using states of the system model observed by its controller. However, rewards received by each process depends not only on its decisions but also on decisions of other processes. As objective functions of the controllers are independent, a general-sum stochastic game is established in this scenario.

### IV. Stochastic Game Formulation

A stochastic game is defined by a tuple \(\{I, S, A, P, R\}\) where \(I\) is a set \(n\) of players; \(S = \{s\}\) is a countable state space; \(A = \{A_1(s) \times A_2(s) \ldots \times A_n(s)\}_{s \in S}\) is a countable joint action space, in which \(A_i(s)\) is a set of admissible actions in state \(s \in S\) for the \(i\)-th player; \(P\) is a matrix of conditional probabilities \(P(s'|s, a)\), which is the probability of moving from state \(s\) to state \(s'\) if joint action vector \(a \in A(s)\) is taken; \(R = \{R_i(s, a)\}_{i \in I}\) are the players’ reward functions where \(R_i(s, a)\) returns the reward value given to the \(i\)-th player if joint action vector \(a\) is taken in state \(s \in S\).

In context of the problem under study, each controller corresponds to a player, and the players’ objectives are to minimize the average transmission delay and loss rates. Subsequently, the tuple \(\{I, S, A, P, R\}\) of the stochastic game between the traffic controllers is defined as follows:

- \(I\) is the group \(L\) traffic controllers, each one is a player.
- \(S\) is the state space of the defined system model consisted of \(M\) (\(M \geq 1\)) overlay paths. Each state is a vector of the number of packets queuing on each path.
- \(A = \{A_1 \times A_2 \ldots \times A_L\} \) where \(A_i = \{a_1, a_2 \ldots a_M\}\) is the set of \(M\) admissible actions made available for all players. Each of them corresponds to an action of sending a packet (or a group of packets) to one of \(M\) paths .
- \(P = \{P(s'|s, a)\}_{s,a \in S}\) is the state transition probability matrix of the defined system model. Since \(P\) is a function of \(a\), a direct computation of \(P\) is very challenging. Therefore, a reinforcement learning technique, specifically Correlated-Q Learning, is utilized to find a solution for the game without the necessity of computing \(P\).
- \(R\) is a set of the players’ immediate reward functions, which should be able to reflect the minimum average transmission delay and loss rate objectives. According to Little’s theorem [16], the average delay is proportional to the average path queue length. Therefore, we define the immediate reward function for the \(i\)-th player as follows:

\[
R_i(s, a) = (\mu^i_a s + d_a)(1 - p_{s, a}^{loss})
\]

where \(\mu^i_a\) is the average inter-departure interval of the path chosen by action \(a_i\); \(s\) is the state of the path observed by the \(i\)-th player; \(d_a\) is the path propagation delay; and \(p_{s, a}^{loss}\) 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 \(p_{s, a}^{loss}\) can be estimated from network measurements using techniques e.g. [4].

We are interested in finding a solution that gives each player the best-possible average reward. In game theory, Correlated Equilibrium [18] is a solution concept that satisfies our requirement. Therefore, to guide players to playing Correlated Equilibrium, Correlated-Q Learning technique is applied.

### V. Correlated-Q Learning

Q-Learning [21] is an effective reinforcement learning technique for solving Markov Decision Processes (MDPs) in the situation when the state transition probability matrices are not known. The basic idea of Q-Learning is to directly learn MDPs’ value functions by interacting with control environments without the need of obtaining the state transition probability matrices. In a Markov Decision Process, the value function is a function of states (or of state-action pairs) that estimates the value in terms of future rewards, or to be precise, in terms of expected return the decision maker can receive by being in a given state (or by performing a given action in a given state). Accordingly, if the value function is learned, the optimal policy is simply the set of actions that maximizes (or

---

\(^1\)RTT jitter is defined as the difference between two consecutive RTT values i.e. \(\text{Jitter}_i = \text{RTT}_{i+1} - \text{RTT}_i\).

\(^2\)Thanks to the monitoring mechanism the buffer can be dynamically allocated in order to avoid overflows.
minimizes) the function at each state. Q-Learning provides the following update rule to successively approximate the value function \( Q(s, a) \) (henceforth, referred to as the Q-function):

\[
Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha[r_t + \gamma V^*(s_{t+1})]
\]

where \( \alpha \in [0, 1) \) is the learning rate; \( \gamma \in [0, 1) \) is the discount factor; \( r_t \) is the reward received at time step \( t \); and \( V^*(s_{t+1}) \) is the value function that maximizes the Q-function at state \( s_{t+1} \) over all actions \( a \). Subsequently:

\[
V^*(s_{t+1}) = \max_a Q_t(s_{t+1}, a)
\]

Theoretically, the idea of Q-Learning can be applied to solve stochastic games in a straightforward manner. However, the major difference when applying Q-Learning on stochastic games is that the value function \( V^* \) now depends on a joint action vector \( \vec{a} \) instead of a single action \( a \). This difference makes (1) become invalid since one player cannot maximize its reward based on the actions taken by other players. To remedy this problem, Greenwald et al. [9] propose a new value function, which is defined as follows:

\[
V^*_i(s_{t+1}) = \text{CE}_i(Q^1_t(s_{t+1}), Q^2_t(s_{t+1}) \ldots Q^l_t(s_{t+1}))
\]

where \( \text{CE}_i(X_1, X_2 \ldots X_n) \) is the reward value received by the \( i \)-th player according to some Correlated Equilibrium (CE) in a one-shot game specified by reward matrices \( X_1, X_2 \ldots X_n \) [10]. In this case, the one-shot game is determined by the matrix of the players’ Q-function values in state \( s_{t+1} \) for all admissible joint action vector \( \vec{a} \in A(s_{t+1}) \). Since our objective is to maximize the maximum reward of each player, if multiple Correlated Equilibria exist, for the \( i \)-th player, we select a strategy to play the equilibrium satisfying:

\[
\sigma^i \in \arg\max_{\sigma \in \text{CE}} \sum_{\vec{a}} \sigma(\vec{a})Q_t(s, \vec{a})
\]

where \( \sigma^i \) is the strategy for the \( i \)-th player, or in other words, the probability of taking an action to play the chosen equilibrium assigned to the \( i \)-th player; and \( \sigma = \prod_i \sigma^i \) is the product of other players’ equilibrium strategies.

Unlike Nash, Correlated Equilibrium can be computed easily using Linear Programming techniques. Nonetheless, the implementation of the Correlated-Q Learning algorithm still requires the sharing of Q-functions among the players. Although a theoretical proof of conversion for Correlated-Q Learning algorithm has not been found, a practical proof of conversion for the algorithm has been seen in a wide range of problems [10], including the one under study.

VI. NUMERICAL STUDY

We assess the impact of having multiple smart traffic controllers by comparing the controllers’ performance in two simulation scenarios. In the first scenario, we have only one controller competing with background traffic to transfer data over 2 wired disjoint overlay paths. The controller implements 3 algorithms, which subsequently are Join the Shortest Queue (JSQ), Weighted Random Round Robin (WRRR), and Markov Decision Process (MDP) to select transmission paths. While the JSQ algorithm always chooses a path with the shortest queue for transmission, the WRRR algorithm opportunistically selects a path using a throughput proportional selection scheme in which, at every control epoch, the average throughput of each path is estimated and normalized into the selection probability of the path. In the second scenario, there is another controller of the same type simultaneously transferring data over the paths. Apart from JSQ, WRRR, and MDP, Correlated-Q Learning algorithm is implemented to derive the best-possible strategy for each controller in this 2x2 game. Performance of the controllers is evaluated and compared in terms of average transmission delay and loss rates.

A. Simulation Setup

A simulation network that provides 2 wired disjoint overlay paths is set up using Ns-2 [17]. Background traffic on each path is generated using FTP, CBR and Pareto ON/OFF traffic sources, the parameters of which e.g. file size, sending rate and on/off time are randomly chosen (see TABLE I) to guarantee the network dynamics. The path capacity and one-way propagation delay are subsequently set to 1.0Mbps and 40ms for the first path and 0.5Mbps and 50ms for the second. Packets arrive at the traffic controllers following the Poisson distribution with the average arrival rate of 768Kbps in the first, and 384Kbps in the second scenario. The reason for reducing the average packet arrival rate to 384Kbps in the second scenario is to maintain the similar bandwidth requirement in both scenarios. The packet size is set to 1500 bytes. Each scenario is simulated for approximately 1000 seconds with the background traffic occupying from 30% – 70% of the path capacity. The results are averaged over 30 runs to guarantee statistical reliability.

B. Performance Comparison

TABLE II shows the average transmission delay and loss rates obtained in the first and second simulation scenarios for an individual controller. As shown, in the first scenario where there is only one “smart” traffic controller, the MDP performs significantly better than the JSQ i.e. 61.8ms vs. 89.3ms. The WRRR controller also performs relatively well, i.e. 67.7ms vs. 61.8ms of MDP and 89.3ms of JSQ, even
through its performance is quite sensitive to the selection of the control epochs. However, both MDP and WRRR controllers have lost their significant performance advantage over the JSQ in the second scenario despite that all the three suffered some performance losses in comparison to the first scenario. Meanwhile, the controllers implementing the Correlated-Q Learning algorithm (henceforth the CEQ controllers) perform exceptionally well in the second scenario although its variance is relatively high (10.4ms) compared to JSQ’ (8.7ms) and MDP’ (9.2ms) due to the effect of learning. The average transmission delay and loss rates obtained by the CEQ are almost the same as those obtained with the MDP in the first scenario i.e. 67.1ms(0.08%) vs. 61.8ms(0.10%).

Fig. 4 illustrates the bandwidth utilization of the CEQ controllers in comparison with the utilization of the MDP controllers in the second scenario. The bandwidth is estimated every 1.0 second. (The continuous line in the middle is the 10-second moving average). As illustrated, except for the starting period when the CEQ were learning, the bandwidth utilization of the CEQ is higher than that of the MDP.

From the obtained performance picture we can see, the presence of the second controller significantly affects the controllers’ performance, especially the MDP controllers’. There are several reasons leading to this performance loss. First, the MDP controllers make their decisions based on the assumption that the underlying network is stationary. When the second controller is in action, this assumption no longer holds. Second, the expected reward receiving by an individual MDP controller no longer depends only on the action taken by the controller. Instead it depends on the controllers’ joint actions. As a result, decisions made by each individual MDP controller are no longer optimal. A similar problem is also experienced by the JSQ and WRRR controllers. With the presence of the second controller, decisions made by one
controller considerably affect the performance gain expected by the other controller while making its decisions. However, compared to the WRRR controller, the JSQ suffers less performance losses. From our point of view, the reason is that the JSQ controller reacts faster to the presence of the second controller since its decisions are made based on the instant path queue lengths. Meanwhile, the WRRR controllers make their decisions based on the average path throughput, which is estimated periodically only after each control interval. Furthermore, performance of the WRRR controllers is quite sensitive to the choice of control intervals. A too short interval would give a big error in throughput estimation while a too long interval would greatly affect the ability of “detecting” the presence of the second controller.

In contrast, we have observed an impressive performance of the CEQ controllers. In theory, each individual CEQ controller is a MDP agent. Hence, there should be an internal mechanism that gives the CEQ advantage over the MDP. To understand this mechanism, we examine the evolution of the path queue length of the controllers in the simulations.

Fig. 5 depicts the evolution of the queue length on the first path observed by each individual JSQ, WRRR, MDP, and CEQ controller (a similar evolution observed on the second path and therefore excluded here due to the space constraint). As depicted, the trend in the queue length evolution of the two controllers (Controller 1 and Controller 2) is clearly observed for the JSQ and MDP case. Since objectives of the two controllers, 1 and 2, are identical, this phenomenon indicates that these controllers are non-cooperative. When the first controller selects one path to transmit data then it is likely that the second controller also selects the same path. As a consequence, their decisions virtually eliminate each other leading to a high queue length for both controllers.

On the other hand, the trend is partially observed in the evolution of the WRRR controllers’ queues, and absolutely disappeared in the CEQ controllers’ queues. A closer look at the evolution of the CEQ controllers’ queues (Fig. 6), and the WRRR controllers’ queues (Fig. 7) reveals a very interesting phenomenon. In the CEQ case, every time the queue length of Controller 1 goes high, the queue length of Controller 2 goes low and vice versa. This phase inverting evolution is the internal mechanism that benefits both CEQ controllers as they manage to keep a relatively low average queue length, hence low end-to-end delay and loss rates, compared to the WRRR, the MDP and especially the JSQ controllers. This mechanism emerges as a result of the cooperation between the two controllers. In contrast, the queue length evolutions of the WRRR controllers are mainly synchronized, although the opportunistic path selection scheme does help the WRRR controllers avoid some of the “colliding” decisions. This synchronization is resulted in a higher queue length, and subsequently, higher end-to-end delay and loss rates.

We also observe that the time required for the CEQ controllers to “learn” is approximately 100 – 150 seconds in this simulation setting. Depending on the network dynamics, the learning time can be longer or shorter. However, since the network dynamics usually show some periodical behaviours, historical data can be used to train the controllers in advance to reduce the learning time and improve performance.

VII. Conclusion and Future Work

To conclude, in this paper we investigated the impact of having multiple competitive traffic controllers while solving the problem of optimal multipath data transfer in overlay networks. We formulated the problem as a stochastic game, and the Correlated-Q Learning technique was applied to guide the controllers to playing a cooperative game. Through a proof-of-concept simulation study, we showed that by being cooperative, the traffic controllers can achieve a significantly higher performance than by acting selfishly. However, since the controllers by their nature are selfish, we need a mechanism to force them to play cooperative games. As mentioned, although Correlated-Q Learning is capable of establishing a cooperative game between players, its requirement on sharing the Q-functions of the players makes the implementation of the algorithm very difficult in practice. Hence, in future work we will analyze a condition, in which cooperations can arise between the controllers, and a more practical mechanism to enforce this condition will also be investigated.
REFERENCES


