ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Multi-Hop Wireless Network Optimization Solution using Automatic Distributed Joint Routing for Delay Sensitive Applications

[I]S.Akila Rajini, [II]J.Raja Shahul Hameed
[I]Research Scholar, Dept. of CSE, Manonmaniam Sundaranar University
Tirunelveli, Tamil Nadu, India
[II]M.E., Dept. of CSE, PSN College of Engineering and Technology, Melathediyoor
Tirunelveli, Tamil Nadu, India

Abstract
Multi-hop wireless network can be modeled as a dynamic network, consisting of several interconnected wireless nodes, which aim to jointly optimize the overall network utility, given the resource constraints of the wireless communication channels and also, importantly, the mutual interferences (coupling) resulting when nodes are simultaneously transmitting. The proposed system provides a distributed routing and power control algorithm that enables nodes in a multi-hop network to autonomously optimize the overall performance of delay-sensitive applications by determining their routing and transmission power to maximize the network utility, in a dynamic environment. This is achieved with the help of Markov Decision Process with distributed computation of the optimal policy, which enables individual nodes to make optimal decisions for their cross-layer strategies autonomously, by relying only on their local available information rather than acquiring global information which would cause a large delay and high communication overhead. The reinforcement-learning method is used to find the optimized policy when the dynamics are unknown. A pricing-based distributed resource allocation framework is also adopted in proposed approach. By using this cooperative approach, the video quality can be improved.

Keywords
Cross-layer strategies, Markov Decision Process, Reinforcement learning

I. Introduction

Wireless communications is the fastest growing segment of the communications industry. As such, it has captured the attention of the media and the imagination of the public. Cellular systems have experienced exponential growth over the last decade and there are currently around two billion users worldwide. Indeed, cellular phones have become a critical business tool and part of everyday life in most developed countries, and are rapidly supplanting antiquated wireline systems in many developing countries. In addition, wireless local area networks currently supplement or replace wired networks in many homes, businesses, and campuses. Many new applications, including wireless sensor networks, automated highways and factories, smart homes and appliances, and remote telemedicine, are emerging from research ideas to concrete systems. The explosive growth of wireless systems coupled with the proliferation of laptop and palmtop computers indicate a bright future for wireless networks, both as stand-alone systems and as part of the larger networking infrastructure. However, many technical challenges remain in designing robust wireless networks that deliver the performance necessary to support emerging applications. Many technical challenges must be addressed to enable the wireless applications of the future. These challenges extend across all aspects of the system design. As wireless terminals add more features, these small devices must incorporate multiple nodes of operation to support the different applications and media. Computers process voice, image, text, and video data, but breakthroughs in circuit design are required to implement the same multimode operation in a cheap, lightweight, handheld device. Since consumers don't want large batteries that frequently need recharging, transmission and signal processing in the portable terminal must consume minimal power. The signal processing required to support multimedia applications and networking functions can be power-intensive.

Multi-hop wireless networks can provide flexible network infrastructures at a low cost. In a multi-hop wireless network, a given node has a fixed amount of data to send to a destination node within a hard delay constraint [1]. The use of multiple hops to transport data has been shown to enhance network capacity and may be necessary due to cabling limitations in many environments [4].

In this paper, a communication scenario where multiple delay-sensitive video streams need to be concurrently transmitted over a multi-hop wireless network is used. At each hop, a node can optimize its relay selection, transmission power in order to support the transmission of these delay sensitive streams while explicitly considering the impact of its selected strategy on its neighboring nodes at the various OSI layers (power interference at the physical layer, network congestion at the network layer, etc.). Thus it is suitable when delay-sensitive applications need to be transmitted across the multi-hop wireless networks and able to achieve good performance in a dynamic wireless environment.

The paper is organized as follows. The next section gives an overview of the system model. Section 3 highlights the Methodologies. Section 4 highlights the Experimental Evaluation. Finally, Section 5 concludes the paper.

II. System Model

Initially the multi-hop wireless network setting at different layers is considered. Each source node needs to transmit its traffic to a destination node. Hence, each data packet in the network has a specific destination, and it is assumed that the relay nodes can extract this information from the IP header of the packet. The assumption is that each node operates in full-duplex mode, and can only communicate with the nodes in its neighborhood.

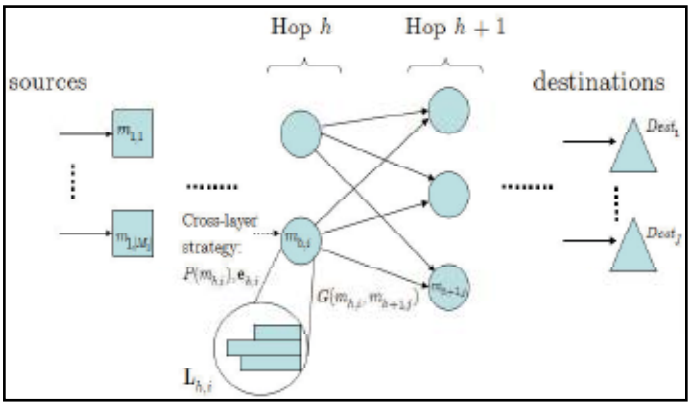

Figure 1: A multi-hop wireless network with time-varying channels and delay-sensitive source packets.

A. Layer Based Communication

Each data stream corresponds to a source-destination pair. The network consists of H hops with the first hop being source nodes, and we define $Mh = \{m_{h,1}, m_{h,2}, \ldots, m_{h,|M_h|}\}$ to be the set of nodes at the h th hop ($1 \le h \le H$). The destination nodes are $\{Dest_1, Dest_2, \ldots, Dest_J\}$. We investigate the performance of transmitting V delay sensitive data streams over a multi-hop wireless network. Delay-sensitive data packets with different delay deadlines are generated by multiple sources in the first hop, and relayed hop-by-hop by the wireless nodes in the multi-hop network until the destinations at the H-th hop receive the packets. Again the assumption is that the system is time-slotted and wireless nodes determine their cross-layer transmission strategies at the beginning of each time slot. Every node maintains several transmission queues for its received packets with different remaining lifetimes (i.e. time until delay deadline expires).

Physical Layer Model

It is defined with the transmission powers from the h-th hop to the (h+1)-th hop at time t, with its i-th entry being the transmission power of node $m_{h,i}$, i.e. $P^t_h(i) = P^t(m_{h,i})$. And it is also defined a channel-state matrix G^t_h from the h-th hop to the (h+1)-th hop at time t as $G^t_h = G^t(m_{h,i}, m_{h+1,j})$, where $G^t(m_{h,i}, m_{h+1,j})$ is the propagation gain of the channel from node $m_{h,i}$ to node $m_{h+1,j}$ at time t.

Network Layer Model

The model used $e^t_{h,i}$ to represent node mh,i's routing decision at time t , i.e. $e^t_{h,i}(j) = 1$ if node mh,i selects $m_{h+1,j}$ as its relay at time t, and $e^t_h,i(j) = 0$ otherwise. The packets at the nodes are transmitted according to the routing decisions, and received with the corresponding probability, i.e. $1 - p^t e(m_{h,i}, m_{h+1,j})$ for the erasure channel from node $m_{h,i}$ to node $m_{h+1,j}$. For every time slot, each node can only transmit one packet at a time from its transmission queue. Once a packet is transmitted, it leaves the transmission queue, and it is assumed that there is no retransmission for the lost packets.

B. Dynamic Decision Making

MDP has been used to solve various wireless networking problems, where the decision maker needs to capture the network dynamics, and take into account the effect of its current action on the future performance. The method also uses MDP to model and capture the network dynamics (time-varying channels and data sources). Then a distributed computation of the optimal policy based on the factorization of MDP is considered. The proposed method enables wireless nodes to make decisions autonomously, based on their local available information exchanged with their neighboring nodes.

C. Finding Optimal Policy

There are different approaches to compute the optimal policy, such as value iteration and policy iteration. In value iteration, a state value function V (s) is defined as the expected accumulated discounted reward when starting with state s [2].

$$V^{t+1}(s) = \max_{a \in A} \left\{ R(s,a) + \gamma \sum_{s' \in s} P(s'|s,a) V^t(s') \right\} \quad \text{Eq 1}$$

The hop-by-hop structure of the network can be applied to derive a more compact representation of the state transition probability by factorizing it into local state transition probabilities at different nodes, which can further lead to an efficient distributed computation of the optimal policy. Moreover, this method developed a distributed algorithm to approach the optimal policy, which can avoid the large delay and high communication overhead caused by the acquisition of global information [5].

D. Value Function Update with Less Information Exchange

This method can reduce the information exchange overhead by reducing NI. The centralized solution without MDP formulation does not consider the network dynamics and only takes myopic actions. Therefore, it requires a much lower complexity than the centralized MDP approach [6].

It can also reduce the information exchange overhead by increasing TI. An important limitation of the learning method is that it requires information feedback at the same frequency as the decision making, i.e. the approximate value functions from other nodes need to be updated every time slot. It computes the $n(t)$-step temporal difference ($n(t) = (M + 1)TI - t$) as:

$$\delta^t_{H,a}(\mathbf{s}^t_{\mathcal{F}(H)}) = (1-\lambda)\sum_{t'=t}^{(M+1)TJ} \lambda^{(M+1)TJ-t'} \mathbf{R}_H(\mathbf{s}^{t'}_{\mathcal{F}(H)}) + \gamma V^t_{H,a}(\mathbf{s}^{(M+1)TJ}_{\mathcal{F}(H)}) - V^t_{H,a}(\mathbf{s}^t_{\mathcal{F}(H)}) \quad \text{Eq 2}$$

III. Methodologies

A. Reinforcement Learning

Reinforcement learning is the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. There are two main strategies for solving reinforcement-learning problems. The first is to search in the space of behaviors in order to find one that performs well in the environment. The second is to use statistical techniques and dynamic programming methods to estimate the utility of taking actions in states of the world.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

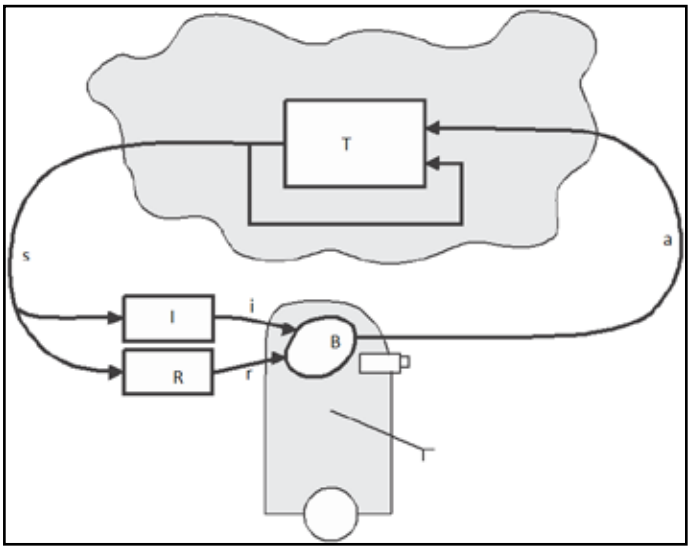

Figure 2 The Standard Reinforcement-Learning Model

In the standard reinforcement-learning model, an agent is connected to its environment via perception and action, as depicted in Figure. On each step of interaction the agent receives as input, i, some indication of the current state, s, of the environment; the agent then chooses an action, a, to generate as output. The action changes the state of the environment, and the value of this state transition is communicated to the agent through a scalar reinforcement signal, r. The agent's behavior, B, should choose actions that tend to increase the long-run sum of values of the reinforcement signal. It can learn to do this over time by systematic trial and error. Using reinforcement learning, the policy can be updated more quickly and using only local information.

B. Online and Actor-Critic Learning

In real wireless environment, the global state transition probability is usually not known by the distributed nodes. Moreover, if a centralized controller is implemented to collect all the information, it will bring both high communication overhead and large delay, and cannot support delay-critical applications well. Hence, a learning method is required to update the value-function online, and adapt the cross-layer transmission strategies on-the-fly [5]. During an online adaptation process, the learning algorithm first chooses its action according to the current state and value-function, and then the network transits to the next state and receives the immediate reward. The AC learning separates the value-function update and policy update. The value-function (i.e. the critic), is used to strengthen or weaken the tendency of choosing a certain action. The policy structure, or the actor, is a function of state-action pair, i.e. ρ(s, a), which indicates the tendency of choosing action a at state s [6].

C. Markov Decision Process

Markov decision processes (MDPs) provide a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker. MDPs are useful for studying a wide range of optimization problems solved via dynamic programming and reinforcement learning. More precisely, a Markov Decision Process is a discrete time stochastic control process. At each time step, the process is in some state s, and the decision maker may choose any action 'a' that is available in state 's'. The process responds at the next time step by randomly moving into new state 's', and giving the decision maker a corresponding reward Ra(s, s').

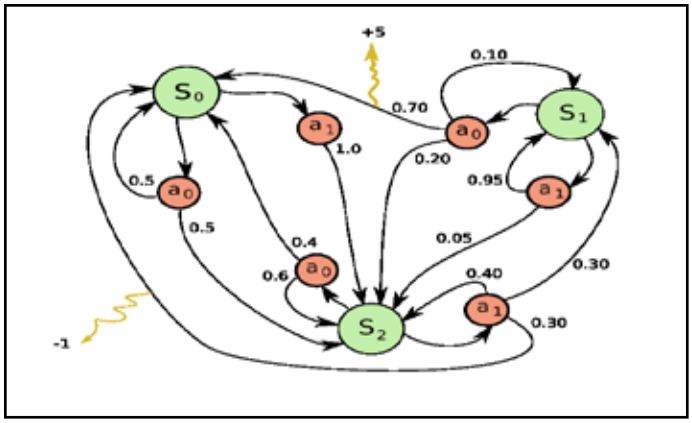

Figure 3 Example of a Simple MDP with 3 States and 2 Actions

The probability that the process moves into its new state s' is influenced by the chosen action. Specifically, it is given by the state transition function Pa(s, s'). Thus, the next state s' depends on the current state s and the decision maker's action a. But given s and a, it is conditionally independent of all previous states and actions.

MDP Representation:

An MDP can be defined by a tuple $(\mathcal{S},\mathcal{A}, P,R)$, with $\mathcal{S}$ as its state space, $\mathcal{A}$ as its action space, $P(s'|s, a)$ as the state transition probability and $R(s, a)$ as its reward [7].

IV. Experimental Evaluation

A. n- Step td-learning algorithm

Using -step TD-learning to reduce the information exchange overhead

if $t = kT$I then

1. Receive the information feedback, i.e. the approximate value function
2. Use the n-step TD-learning to update the approximate value function
3. Send the approximate value function and reward information to all the nodes requesting information feedback (this feedback contains the information for the previous TI slots.)

else

- Observe the current local state sFh, make the estimation of, and take the action given by the current approximate value function
- Record both the experienced state and reward for future feedback to other nodes

Temporal difference (TD) learning is an approach to learning how to predict a quantity that depends on future values of a given signal. The name TD derives from its use of changes, or differences, in predictions over successive time steps to drive the learning process. The prediction at any given time step is updated to bring it closer to the prediction of the same quantity at the next time step. It is a supervised learning process in which the training signal for a prediction is a future prediction. TD algorithms are often used in reinforcement learning to predict a measure of the total amount of reward expected over the future, but they can be used to predict other quantities as well.

An obvious approach to learning the value function is to update the estimate of the value function when the actual return is known.

This method is called the constant-a Monty Carlo method, where 'a' is a learning parameter between 0 and 1. Since the actual return is the sum of all future rewards, this algorithm must wait until the end of the episode when the expected return is known before the value function is updated. The update to the value function takes the difference of successive estimates of the value function, thus the name temporal difference.

TD learning can often be accelerated by the addition of eligibility traces. When the lookup-table TD algorithm described above receives input (y_{t+1}, x_{t+1}) it updates the table entry only for the immediately preceding signal x_t.. That is, it modifies only the immediately preceding prediction. But since y_{t+1} provides useful information for learning earlier predictions as well, one can extend TD learning so it updates a collection of many earlier predictions at each step.

Both TD and Monte Carlo methods use experience to solve the prediction problem. Given some experience following a policy ' π', both methods update their estimate V of V^{π}. If a nonterminal state S_t is visited at time t, then both methods update their estimate $V(S_t)$ based on what happens after that visit. Roughly speaking, Monte Carlo methods wait until the return following the visit is known, then use that return as a target for $V(S_t)$ A simple every-visit Monte Carlo method suitable for nonstationary environments is

$$V(S_t) \leftarrow V(S_t) + \alpha[R_t - V(S_t)] \quad \text{-- Eq 3}$$

where R_t is the actual return following time t and α is a constant step-size parameter.

Let us call this method constant α MC. Whereas Monte Carlo methods must wait until the end of the episode to determine the increment to $V(S_t)$ (only then is R_t known), TD methods need wait only until the next time step. At time t+1 they immediately form a target and make a useful update using the observed reward r_{t+1} and the estimate $V(S_{t+1})$ The simplest TD method, known asTD(0), is

$$V(S_t) \leftarrow V(S_t) + \alpha[r_{t+1} + \gamma V(S_{t+1})] \quad \text{--Eq 4}$$

In effect, the target for the Monte Carlo update is R_t , whereas the target for the TD update is $r_{t+1} + \gamma V(S_{t+1})$.

B. Analysis

Figure 4 Trace File

The various traces begin with a single character or abbreviation that indicates the type of trace, followed by a fixed or variable trace format. The tables listing the trace formats differ between fixed and variable trace formats:

1. For fixed trace formats, the table lists the event the triggers the trace under the Event heading and the characters that start the trace under the Abbreviation heading. The format is listed across the last two columns, and the the type and value for each element of the format are listed beneath under the Type and Value headings. Some events have multiple trace formats.
2. For variable trace formats, the table lists the event the triggers the trace under the Event heading and the characters that start the trace under the Abbreviation heading. The last three columns list the possible flags, types, and values for the event under the Flag, Type, and Value headings.

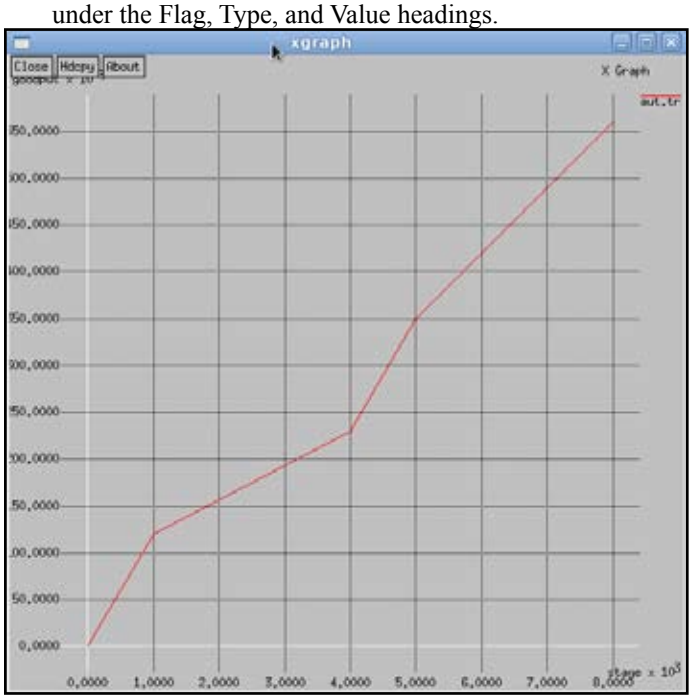

Figure 5 Comparison of Video Quality

This method also compares the performances of distributed actor-critic learning with different feedback frequencies by evaluating the quality of received video packets (measured by PSNR) after the learning process has converged. The time-slot is assumed to be 1.0ms. Performance is being compared in a 3-hop network too.

V. Conclusion and Future Work

In this paper, Markov Decision Process is applied for both joint routing and power control mechanism in wireless multi-hop networks as a solution. Based on the factorization of the state transition probability, it is derived with a distributed computation method for finding the optimal policy. In order to reduce both the communication overhead and delay incurred due to the inter-node information exchanges, it has been proposed an on-line learning method which enables the nodes to autonomously learn the optimal policy. Moreover, when the network protocol allows for block acknowledgments to be deployed, an $n(t)$-step learning method is proposed to further reduce the information exchange overhead and improve the network performance

In immediate future, an advanced system for improving the video quality will be implemented. The proposed system will maximize the long-term sum of utilities across the video terminals in a decentralized fashion, by jointly optimizing the packet scheduling, the resource allocation, and the cooperation decisions.

References

[1] *R. Agarwal and A. Goldsmith, "Joint rate allocation and routing for multihop wireless networks with delay-constrained data," Technical Report, Wireless Systems Lab., Stanford University, CA, USA, 2004.*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

[2] D. P. Bertsekas, Dynamic Programming and Optimal Control, 2nd edition. Athena Scientific, 2000.
[3] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: a reinforcement learning approach," Advances in Neural Information Processing Systems, vol. 6, 1994.
[4] R. Cruz and A. Santhanam, "Optimal routing, link scheduling, and power control in multi-hop wireless networks," in Proc. IEEE INFOCOM, Apr. 2003, pp. 702-711.
[5] L. Kaelbling, M. Littman, and A. Moore. "Reinforcement learning: a survey," J. Artificial Intelligence Research, vol. 4, pp. 237-285, 1996.
[6] R. Sutton and A. Barto, Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press, 1998.
[7] Q. Zhao and A. Swami, "A decision-theoretic framework for opportunistic spectrum access," IEEE Wireless Commun. Mag., vol. 14, no. 4, pp. 14-20, Aug. 2007

Mrs.S.Akila Rajini, Research Scholar, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli District, TamilNadu, India. She received the M.E. degree in Computer Science and Engineering from Anna University, Chennai in 2009. She is pursuing her doctoral degree in the area of Wireless Networks. Presented Papers in more than fifteen National and International conferences. Published articles in International Journals in the area of Text Mining, Image Compression, Optimized Routing in Wireless Networks.

Mr. Raja Shahul Hameed pursuing his Post Graduation in Computer Science and Engineering at PSN College of Engineering and Technology has interest in the area of Networking, Network Security. He has conducted a workshop and been a resource person for "Open Source Software" at SCAD Engineering College. And to add he has conducted a seminar on Sixth Sense technology, at TDMNS Arts College, Tirunelveli. He has presented more than 5 papers in national-level conference and 2 in international level conferences.