QoS Multicast Routing Protocol Oriented to Cognitive Network Using Competitive Coevolutionary Algorithm

Xingwei Wang[a], Hui Cheng[b], Min Huang[a]

[a]*College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*

[b]*School of Computing & Mathematical Sciences, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK*

Abstract:

The human intervention in the network management and maintenance should be reduced to alleviate the ever-increasing spatial and temporal complexity. By mimicking the cognitive behaviours of human being, the cognitive network improves the scalability, self-adaptation, self-organization, and self-protection in the network. To implement the cognitive network, the cognitive behaviours for the network nodes need to be carefully designed. Quality of service (QoS) multicast is an important network problem. Therefore, it is appealing to develop an effective QoS multicast routing protocol oriented to cognitive network.

In this paper, we design the cognitive behaviours summarized in the cognitive science for the network nodes. Based on the cognitive behaviours, we propose a QoS multicast routing protocol oriented to cognitive network, named as CogMRT. It is a distributed protocol where each node only maintains local information. The routing search is in a hop by hop way. Inspired by the small-world phenomenon, the cognitive behaviours help to accumulate the experiential route information. Since the QoS multicast routing is a typical combinatorial optimization problem and it is proved to be NP-Complete, we have applied the competitive coevolutionary algorithm (CCA) for the multicast tree construction. The CCA adopts novel encoding method and genetic operations which leverage the characteristics of the problem. We implement and evaluate CogMRT and other two promising alternative protocols in NS2 platform. The results show that CogMRT has remarkable advantages over the counterpart traditional protocols by exploiting the cognitive favours.

Keywords: Cognitive network, reference model of brain, QoS multicast routing, cognitive behaviour, competitive coevolutionary algorithm

1. Introduction

With the rapid development in networking technologies, the future networks are expected to provide real-time, secure, reliable, and high-quality services to the users. The connections to the Internet should be available anytime anywhere. However, the technical advancement has also significantly increased the network complexity. The network services required by the users are far beyond the scope of the traditional data service. Since the network is not aware of its own states and requirements, the network management becomes an extremely difficult task. If the network elements can intelligently adapt to the network operations, the increased complexity will be effectively alleviated without consuming extra resources. Therefore, the future networks are expected to exhibit the following characteristics [1,2,3].

- Scalability. The network can work as normal when a large number of nodes and users join it.
- Adaptability. The network can actively adapt to the environmental changes.
- Survivability. The network can provide continuous services even when it suffers potential attacks or destruction.
- Mobility. For wireless users, it refers to the location movement. For wired users, it refers to joining or leaving the network freely.
- Diversity. The softwares and hardwares of the network equipments are compatible and cooperative.
- Self-organization. The network can manage itself and reduce the manual operations as much as possible.

In recent years, the cognition concept has been applied to various network and communication systems. Two new terms were created to reflect the technologies, i.e., cognitive radio and cognitive network. In 1999, Mitola [4] proposed the concept of software defined radio, which was the early form of cognitive radio. Its core idea is that the radio interface can actively learn from the surrounding environment by sensing and utilizing the available spectrum resources, thereby restricting and reducing the conflict. In 2005, considering the cognitive radio as an intelligent wireless communication system, the researchers proposed a new metric called interference temperature for the quantification and management of interference [5]. Three fundamental cognitive tasks were addressed as well, i.e., radio-scene analysis, channel-state estimation and predictive modeling, and transmit-power control and dynamic spectrum management.

The cognitive network was originated from the concept of knowledge plane [6]. The key idea of knowledge plane is to add a knowledge layer between the data layer and the control layer in the network. The knowledge layer contains a cognitive process which can abstract high-level objectives from the low-level network behaviours. The cognitive process can make decisions by analyzing the incomplete information. It can also optimize the future network behaviours by exploiting the experiential information. In summary, the cognitive network aims to eliminate the constraints imposed to the current network. It enables the network to sense the current conditions, and then plan, decide, and act on those conditions [7].

The current research focuses on the cognitive radio which manages spectrum resources dynamically. However, we believe that the ideas derived from cognitive science can be applied far beyond this. The future networks need more intelligence to operate with less human intervention. The network nodes can mimick the cognitive behaviours of human being to enable the network intelligence. There is lack of in-depth research to integrate and implement these ideas into the networks, especially the wired networks. In the Internet, the backbone networks and the primary infrastructure are still wired networks. It is appealing to reform the wired backbone networks into cognitive wired networks. Once the networks have cognitive capability, the network protocols also need to be adapted to the cognitive environment. The research in this paper brings new insights into the development of cognitive protocols in cognitive wired networks.

In this paper, we investigate the QoS multicast routing problem [8] in the context of cognitive wired network environment. We propose a cognitive QoS multicast routing protocol named as CogMRT, which works in a hop-by-hop style. Referring to the brain model [9], we design the cognitive behaviours for the wired network nodes to support the protocol. Each node maintains local neighbours' information instead of the unrealistic global information. Inspired by the small-world phenomenon, a few cognitive behaviours are designed for accumulating the experiential information. A competitive coevolutionary algorithm (CCA) is applied for the construction of the multicast trees. We simulate CogMRT in NS2 platform [10]. Performance evaluation shows that it has remarkable advantages over the current routing mechanisms.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 presents various models. In Section 4, we present the carefully designed cognitive behaviours for the network nodes. In Section 5, we present the proposed protocol with details. Section 6 presents simulation results and demonstrates the remarkable performance of CogMRT. Section 7 concludes this paper and presents possible future research directions.

2. Related Work

2.1. Cognitive Network

The cognitive network model is designed by exploiting the idea of knowledge plane. The model is illustrated in Fig. 1. The model can also be represented as a directed connected graph $G(V, E)$ where V is the set of nodes representing the routers in the network and E is the set of edges representing the links in the network. For each router, an additional knowledge plane is added into its protocol architecture. We utilize the cognitive behaviours derived from the cognitive cycle and the layered reference model of brain to design the knowledge plane, thereby improving the network performance.

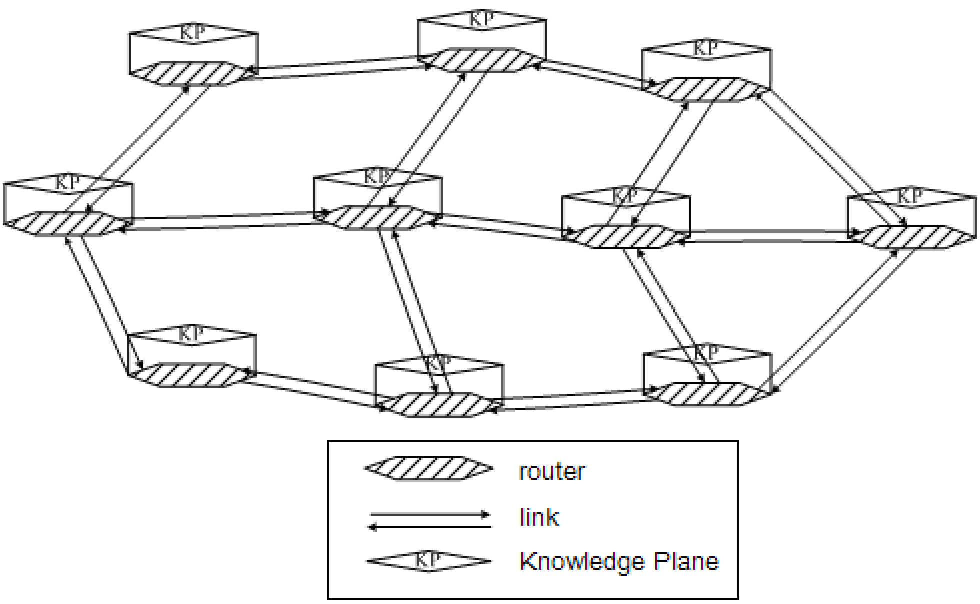

Fig. 1 The model of cognitive network.

Majority of the research work are related to the cognitive radio which deals with dynamic management of spectrum resources [5]. In [11], a new network architecture called cooperative cognitive relay network (CCRN) is proposed. CCRN combines cognitive radio and cooperative relay technologies to improve the efficiency of resource utilization. In CCRN, each secondary user can cooperate with its selected primary user to gain more spectrum access opportunities. Based on the CCRN, this paper proposes an evolutionary game model to aid the selections made by the secondary users. In [12], a new spectrum resource allocation optimization framework is developed for a single-cell multiuser cognitive radio network in the presence of primary user networks. Under the framework, a bandwidth-power product metric is used to evaluate the spectral resource consumption. The framework can significantly enhance the spectral efficiency in a cognitive radio environment compared to a classical power adaptive optimization scheme.

In [13], a cognitive network is considered to have a base station communicating with multiple primary and secondary users. Two different traffic models for the primary user have been considered. One is that the primary users can tolerate a certain average delay and the other is that the primary users do not suffer from any delay. Then a few scheduling and resource allocation algorithms are proposed to minimize the average packet delay of the secondary user and find the optimal assignment of the secondary users to the primary channels. In [14], the model assumes that secondary users can transmit if they can improve the performance of a primary user via cooperation. Two different reward strategies are studied for the secondary users, i.e., immediate reward and long-term reward. Under these strategies, different optimal opportunistic scheduling policies have been applied. The proposed scheduling policies outperform non-cooperative scheduling policies. The work is the first to consider scheduling of cooperative primary and secondary networks with multiple users sharing a common destination.

A small number of research work has investigated the architecture of cognitive network. A cognitive cycle mimicks the feedback control scheme in the biological system. The cognitive cycle has been integrated into the design of novel network architecture. In [15], the system architecture of cognitive network is designed based on the cognitive cycle. Distributed learning and reasoning is used to optimize the network operations. The island genetic algorithm (GA) is applied to optimize the channel assignment in the dynamic spectrum access. In [16], a new concept of cognitive resource manager is proposed which is a multi-purpose software entity. The manager owns a toolbox consisting of various advanced reasoning methods. It collects the information from different layers and then conducts the cross-layer optimization. In [17], a three-layer system architecture of cognitive network is developed and applied to the service assignment problem. The problem has defined four types of QoS parameters, three types of air interfaces and four types of services. Multi-objective optimization algorithm is used to assign the services to appropriate interfaces.

2.2. QoS Multicast Routing

In the wired networks, group communications become an important research topic, which is driven by the popular multimedia collaborative applications such as video conference, content distribution, and distributed games. In the group communications, a source node is required to send data to multiple destinations through a communication network. Real-time and fair delivery of data from the source to all the destinations is often required. To efficiently support QoS group communications, the most important issue that needs to be addressed is QoS multicast routing [8]. An efficient QoS multicast algorithm should construct a multicast routing tree, by which the data can be transmitted from the source to all the destinations with guaranteed QoS. Meanwhile, the QoS multicast routing should also consider the efficient utilization of the network resources. In the cognitive network environment, QoS is also a core problem and reflects the service provision performance. Only with the QoS guarantee, the potential of the cognitive network can be fully exploited.

Multicast routing trees can be classified into two types, i.e., Steiner minimum tree (SMT) [18] and shortest path tree (SPT) [19]. An SMT is also the minimum-cost multicast tree. SPT is constructed by applying the shortest path algorithm to find the shortest (e.g., minimum cost or delay) path from the source to each destination and then merging them. Inspired by SMT and SPT, some heuristic algorithms have been proposed to construct a QoS-aware multicast tree. In [20], the multicast has been used to enable the reprogramming of a subset of the sensor nodes in a wireless sensor network. By reprogramming only a group of nodes, the multicast approach has the potential to extend the network lifetime. A heuristic multicast algorithm is considered which constructs the multicast tree based on the location of group nodes. The small world concepts have been used to build a more efficient network infrastructure by creating shortcuts towards the sink. The incorporation of small world features has the desirable characteristic of reducing the average path length.

In [21], a cognitive multi-channel multi-radio multicast protocol, CoCast, is proposed for vehicular ad hoc networks. It extends a popular protocol in mobile ad hoc network, that is, On-Demand Multicast Routing Protocol (ODMRP). CoCast has borrowed the concept of cognitive radio techniques to overcome the scalability and interference problems in ODMRP. The nodes' cognitive capability is utilized to sense the channel and select a least congested channel from primary and secondary nodes. In [22], the multi-stream multi-source multicast routing problem has been investigated. It determines multiple multicast trees on a given network for delivering one or more data streams. A heuristic algorithm is provided to find a multicast forest which can achieve near-optimal residual bandwidth. The heuristic algorithm is developed on the modification of Dijkstra's Algorithm.

In [23], two methods are proposed to find a multicast tree with the minimum bandwidth consumption for a QoS multicast request in cognitive radio ad hoc networks. The first method has two phases. It first constructs a multicast tree and then assigns timeslots to the tree links. The second method integrates them together. Both methods significantly outperform a SPT-based two-phase method. In [24], a novel multicast scheme is proposed for mobile social networks. This scheme is inspired by the homophily of social networks that friends are usually similar in characteristics. The nodes in frequent contact with the destinations will form destination clouds. The multicast runs in two phases: pre-cloud and inside-cloud. In [25], a QoS-guaranteed multicast routing protocol (QGMPR) was proposed. In QGMPR, if a receiver node intends to join the multicast communication, it will search a QoS routing path to the source node by running any unicast routing protocol. Once all the receiver nodes have joined, the multicast tree is formed.

2.3. Competitive Coevolutionary Algorithm

In this paper, the competitive coevolutionary algorithm [26] is used to search the multicast tree in the cognitive network. The CCA mimicks the predator-prey model in the biological evolutionary process, that is, the predator and the prey compete with each other for survival. The progress made by one party threatens the survival of the other party. One party cannot decide its survival capability by itself because the capability is also severely affected by the other party. In the CCA, normally there are two interacting populations. Individuals are rewarded at the expense of those with which they interact. In our design, the two populations are named as the learner and the evaluator, respectively. The two populations compete with each other and exchange their roles alternatively. The fitness of the learner reflects the result of its competition with the evaluator.

After the crossover and mutation operations, the selection of next generation learner population is by the competition fitness of all the individuals. When the update of the learner popution is finished, it will exchange its role with the evaluator population. The competition process is repeated between the two new populations. The good individuals in both populations are kept and the optimal ones are updated. Thus, both populations are pushed forward to generate high-quality offsprings for competition. The reciprocal forces will drive the coevolutionary algorithm to generate individuals with ever-increasing performance. It also overcomes the premature convergence problem in the standard GA. We denote the learner population as L_{GA} and the evaluator population as E_{GA}. The competitive fitness of the *i*th individual in the learner population is formulated as below.

$$\forall j \in E_{GA} \qquad N_j = \sum_{\substack{k \ defeat \ j \\ k \in L_{GA}}} 1 \tag{1}$$

$$\forall i \in L_{GA} \qquad CF_i = \sum_{\substack{i \ defeat \ j \\ j \in E_{GA}}} \frac{1}{N_j} \tag{2}$$

CF_i reflects the reward that the learner individual has attained by defeating the evaluator individuals. The stronger the defeated evaluator, the larger the reward attained by the learner.

Coevolutionary strategy has been exploited to design new evolutionary algorithms. In [27], a novel coevolutionary technique named multiple populations for multiple objectives (MPMO) is proposed for solving multiple objective optimization problems. Each population is responsible for one objective and an external shared archive is used for different populations to exchange search information. In [28], the concept of the preference-inspired coevolutionary algorithm and its realization, PICEA-g, are systematically investigated for solving many-objective problems. The idea is to coevolve a family of preferences simultaneously with the population of candidate solutions.

Coevolutionary algorithms have also been widely applied to solve theoretical and practical problems. In [29], CCA is used to calculate the suppliers' optimal strategies in a deregulated electricity market. CCA calculates the Nash Equilibrium strategies ensuring the best outcome for each agent. In [30], an effective coevolutionary differential evolution with harmony search algorithm (CDEHS) is proposed to solve the reliability-redundancy optimization problem. In CDEHS, two populations evolve simultaneously and cooperatively for two different parts of the problem. In [31], a Co-evolutionary Improved Genetic Algorithm (CIGA) is proposed for global path planning of multiple mobile robots. The co-evolution scheme relies on the cooperation between populations to avoid collision between mobile robots and obtain optimal or near-optimal collision-free path. In [32], an algorithm framework is developed to make use of co-evolutionary genetic programming for the problem of multi-robot motion planning. Each robot uses a grammar based genetic programming for figuring the optimal path while a master evolutionary algorithm is in charge of the overall path optimality. In [33], a Blockwise Coevolutionary Genetic Algorithm (BCGA) is proposed for high dimensional intelligent watermarking optimization of embedding parameters of high resolution images. The cooperative coevolution is performed between different candidate solutions at the pixel block.

2.4. Evolutionary Algorithms for QoS Multicast Routing

The QoS multicast routing problem has been an attractive and challenging research topic for long time due to its intractability and comprehensive application backgrounds. There are no polynomial algorithms that can solve routing problems that consider more than one QoS-constraint metric [34]. In many cases, the QoS multicast routing has been formulated into a NP-Complete problem. Population-based meta-heuristics are a type of promising techniques to solve combinatorial optimization problems including the SMT problem. Therefore, evolutionary algorithms have been largely investigated for solving the problem of QoS multicast routing.

In [34], a QoS multicast routing protocol, i.e., the core-based tree based on GAs, is proposed over a high-altitude platform (HAP)-satellite platform. Since it has considered three QoS metrics, i.e., cost, bandwidth, and delay, the algorithm is called hybrid cost-bandwidth-delay GA. The protocol performs the multicast tree search that executes the GA. In [35], three immigrants enhanced genetic algorithms are proposed to solve the dynamic QoS multicast routing problem in mobile ad hoc networks. In [36], the network coding based multicast routing problem has been investigated with two optimization objectives, i.e., the cost and the delay. For this problem, the Elitist Nondominated Sorting Genetic Algorithm (NSGA-II) has been adapted by introducing two adjustments, namely the initialization scheme and the individual delegate scheme. These two adjustments help to diversify the population thus contribute to an effective evolution towards the Pareto Front. In [37], an energy-efficient genetic algorithm is used to study the delay-constrained source-based multicast routing problem in mobile ad hoc networks. Heuristic mutation technique is developed to reduce the total energy consumption of a multicast tree.

Evolutionary algorithms have also been used to solve other types of routing and network optimization problems. In [38], a genetic algorithm is proposed for shortest path (SP) routing problems. It has analyzed the algorithms which can solve the shortest path problems in polynomial time. It then pointed out that they would be effective in fixed infrastructure networks, but, they exhibit unacceptably high computational complexity for real-time communications involving rapidly changing network topologies. In [39], an elitist multiobjective evolutionary algorithm based on the nondominated sorting genetic algorithm is proposed for the dynamic multiobjective SP routing problem in computer networks. In [40], a set of dynamic genetic algorithms are proposed to solve the dynamic delay-constrained SP problem in mobile ad hoc networks. Genetic algorithm and its variants have also been applied to the clustering problem [41], joint QoS multicast routing and channel assignment problem

[42], and QoS routing and wavelength assignment problem [43].

2.5. Comparison of Our Work to Related Work

In the above four subsections, we have introduced the latest relevant literature under four aspects. In the following, we summarize the differences between our work and the related work. We give a clear discussion on our contributions compared to those in related work. First, this paper does not investigate cognitive radio network in which the cognitive concepts have been applied to optimize the spectral efficiency or maximize the throughput [11,12,13,14]. Instead, we have designed a cognitive wired network architecture from another angle. As for the cognitive network architecture, compared to [15], we have considered more cognitive behaviours and real-world interconnection networks. In [16], it uses a cognitive resource manager which is a centralized entity. However, our network resource is managed in a distributed way. In [17], three additional layers are presented which bring difficulties for integrating into the current network architecture. Our work focuses on designing cognitive behaviours for the nodes. So it is easy to implement our methods in the current networks.

Second, to the best of our knowledge, this is the first work to utilize the cognitive science techniques and apply them to design cognitive protocols in cognitive wired network environment. In [20], it does not use any cognitive science concept and the nodes have no cognitive capabilities. Since it assumes that the source node knows the locations of all the destination nodes, it is actually a centralized algorithm. In our work, we have equipped the nodes with cognitive capabilities and our algorithms work in distributed way. The small-world concept has been applied throughout our cognitive multicast protocol. In [21], it does not design its own multicast protocol and it runs over ODMRP. It works only in wireless networks, vehicular networks and Wi-Fi networks. Its utilization of cognitive capabilities is confined to the spectrum sensing in cognitive radio. In [22], it proposes a heuristic based on the classical Dijkstra’s Algorithm and the proposed algorithm can be applied to general wired network only. It does not learn any knowledge from cognitive science. The nework and nodes have no intelligence at all and the proposed protocol can not be applied to cognitive wired network.

In [23], it is based on cognitive ad hoc network which is also a kind of wireless network. The cognitive capabilities of the nodes are limited to spectrum sensing and timeslots assignment. The discovery of multicast tree is based on the traditional spanning tree algorithm. In our work, the nodes use their cognitive capabilities to find good routes. Then we use CCA to construct multicast trees. We have also utilized the small-world phenomenon in social network to improve the efficiency of route search. In [24], it is based on mobile social network. The infrastructure is a combination of wired network and wireless network. Its primary contribution is to form destination cloud through learning from social network. The multicast protocol works at the application layer. Our work focuses on wired network with cognitive capabilities and develops cognitive multicast protocol which runs at the network layer.

Third, we have designed a problem specific CCA for the cognitive multicast protocol. The QoS multicast tree construction in cognitive wired network is still NP-Complete as in traditional networks. The problem cannot be solved exactly in polynomial time. We propose to use CCA to solve it. The general procedure of CCA has been followed. However, we have designed the encoding, fitness function, competitive fitness, crossover and mutation based on the problem characteristics. Last, our work is also the first to apply CCA to the multicast problem in cognitive wired network. In [34], a genetic algorithm is used to solve the multicast problem in satellite network whilst in [35], genetic algorithms are used for multicast in mobile ad hoc networks. In [36], it considers the nework coding based multicast and cannot be extended to backbone networks. It considers two classical optimization objectives, i.e., delay and cost. However, in our CCA, the fitness function evaluates the multicast tree by considering both the user utility and the network service provider utility. It is novel and makes an important contribution by incorporating the utilities into the algorithm. In [37], it focuses on reducing the energy consumption of multicast trees in mobile ad hoc networks. However, in wired backbone network, there is stable energy supply. It uses a single population GA to construct the multicast trees and no node has the cognitive capability.

3. Models

3.1. User QoS Requirements Model

To address QoS routing comprehensively, we consider as many QoS parameters as possible in our model. For each link, we consider its total bandwidth, available bandwidth, delay, and error rate. For each node, we consider its delay, delay jitter, error rate, and stability degree. To simplify the problem, a node's delay, delay jitter, and error rate are combined with the related QoS parameters on its adjacent links. In the search of QoS routing paths, we should consider the current load status of the nodes. The stability degree *st* is a novel QoS parameter to represent it. If the load of one node is too heavy, the routing path should bypass it. The stability degree of the node is defined as below.

$$st = \min\{\frac{ACPU}{TCPU}, \frac{AMEM}{TMEM}\} \tag{3}$$

Where *ACPU* is the available CPU cycles of the node, *TCPU* is the toal CPU cycles, AMEM is the available memory, and TMEM is the toal memory. The parameter *st* reflects the bottleneck value among CPU and memory. The bottleneck value determines the current load status and the data processing capability of the node. Large values of *st* are expected.

The user QoS requirements refer to the QoS parameters specified by the user. We classify the network applications into different categories based on the DiffServ model [44]. Each application category is supported by a certain set of QoS parameters. The mapping relationship is formulated by ITU-T G.1010 [45]. Instead of specifying the QoS parameters directly, each user determines which category his/her request falls into. Since the requirement over any QoS parameter could not be always a fixed value, we represent them by intervals. We denote the set of application types as $APT = \{AP_1, AP_2, \cdots, AP_{|APT|}\}$. Each application type is associated with a set of QoS requirements. For example, for the application type AP_i, its QoS requirements set is $APR_i = (\Delta_{bw}^i, \Delta_{dl}^i, \Delta_{jt}^i, \Delta_{ls}^i)$. Among APR_i, the bandwidth requirement is represented by an interval $\Delta_{bw}^i = [bw\_r_L^i, bw\_r_H^i]$, the delay requirement is represented by an interval $\Delta_{dl}^i = [dl\_r_L^i, dl\_r_H^i]$, the delay jitter requirement is represented by an interval $\Delta^i_{jt} = [jt\_r^i_L, jt\_r^i_H]$, and the error rate requirement is represented by an interval $\Delta^i_{ls} = [ls\_r^i_L, ls\_r^i_H]$. For each application type, different service levels can be provided. In this paper, four service levels are provided for the same application type. They are named as diamond level, gold level, platinum level, and bronze level. The details of each level are shown in Table 1.

Table 1 Service levels and QoS requirements.

Level	Bandwidth	Delay	Delay Jitter	Error Rate	Extra Cost
Diamond	$[bw\_r^{i1}_L, bw\_r^{i1}_H]$	$[dl\_r^{i1}_L, dl\_r^{i1}_H]$	$[jt\_r^{i1}_L, jt\_r^{i1}_H]$	$[ls\_r^{i1}_L, ls\_r^{i1}_H]$	Apr^1_i
Gold	$[bw\_r^{i2}_L, bw\_r^{i2}_H]$	$[dl\_r^{i2}_L, dl\_r^{i2}_H]$	$[jt\_r^{i2}_L, jt\_r^{i2}_H]$	$[ls\_r^{i2}_L, ls\_r^{i2}_H]$	Apr^2_i
Platinum	$[bw\_r^{i3}_L, bw\_r^{i3}_H]$	$[dl\_r^{i3}_L, dl\_r^{i3}_H]$	$[jt\_r^{i3}_L, jt\_r^{i3}_H]$	$[ls\_r^{i3}_L, ls\_r^{i3}_H]$	Apr^3_i
Bronze	$[bw\_r^{i4}_L, bw\_r^{i4}_H]$	$[dl\_r^{i4}_L, dl\_r^{i4}_H]$	$[jt\_r^{i4}_L, jt\_r^{i4}_H]$	$[ls\_r^{i4}_L, ls\_r^{i4}_H]$	Apr^4_i

In a multicast routing request, each multicast group member has its own end-to-end QoS requirements. We denote the multicast group as *G*, and the QoS routing request of the group member *m* ($m \in G$) as $R(v_s, v^m_d, AP_i, SL^m_i, Pay^m)$. $v_s \in V$ is the source node, v^m_d is the node where the group member *m* attaches. $AP_i \in APT$ represents the application type of the multicast group and APR_i represents the QoS requirements of this application type. SL^m_i represents the service level requested by *m*. Pay^m represents the upper limit cost that *m* is willing to pay. The QoS multicast routing request aims to find a multicast tree T_{sG} from v_s to all the v^m_d. On the tree, the path to each v^m_d should support QoS at level SL^m_i of AP_i in terms of all the QoS metrics. Moreover, the path price should not be greater than Pay^m.

3.2. User's QoS Satisfaction Degree Model

In our model, the QoS requirements are represented by interval values instead of a single value. However, the actual QoS values experienced by the users may fall into the interval or not. By mapping the actual value of one QoS parameter to its interval, we can calculate the user's QoS satisfaction degree over that parameter. By the psychology, the user's QoS satisfaction degree should follow the S-shaped trend over the interval. It means that when the value of the QoS parameter approaches the lowest end or the highest end, there will have slight changes reflected in the user's QoS satisfaction degree. However, when the value varies in the middle of the interval, there will have remarkable changes.

(1) Bandwidth Satisfaction Degree Function

In terms of the bandwidth, the user always expects to get the largest value. We denote the bandwidth requirement interval as $[bw\_r^i_L, bw\_r^i_H]$. When the actual bandwidth of a routing path is bw_p, the user's bandwidth satisfaction degree function is defined as in Formula 4.

$$Sat(bw_p) = \begin{cases} -\Omega & bw_p < bw\_r_L^i \\ \varepsilon & bw_p = bw\_r_L^i \\ [\dfrac{bw_p - bw\_r_L^i}{bw\_r_H^i - bw\_r_L^i}]^{\alpha} & bw\_r_L^i < bw_p < \dfrac{1}{2}(bw\_r_L^i + bw\_r_H^i) \\ [\dfrac{bw_p - bw\_r_L^i}{bw\_r_H^i - bw\_r_L^i}]^{\beta} & \dfrac{1}{2}(bw\_r_L^i + bw\_r_H^i) \le bw_p < bw\_r_H^i \\ 1 & bw_p \ge bw\_r_H^i \end{cases} \quad (4)$$

Where $\alpha > 1$, $0 < \beta < 1$, ε is a very small positive integer. Ω is a penalty value, which will be applied only when the user's QoS request cannot be satisfied even at the lower end of the interval. With the increase of bw_p, the user's satisfaction degree also gradually increases. The bandwidth function is illustrated in Fig. 2.

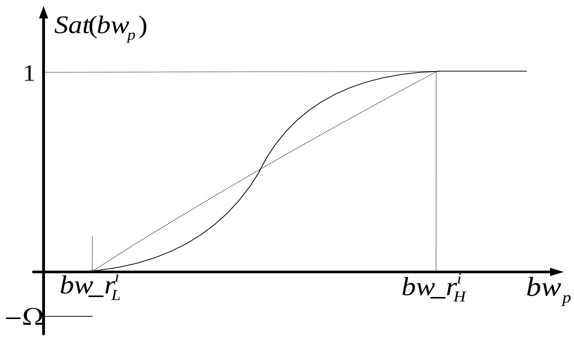

Fig. 2 Diagram of bandwidth satisfaction degree.

(2) Delay Satisfaction Degree Function

In terms of the delay, the user always expects to get the least value. We denote the delay requirement interval as $[dl\_r_L^i, dl\_r_H^i]$. When the actual delay of the routing path is dl_p, the user's delay satisfaction degree function is defined as in Formula 5.

$$Sat(dl_p) = \begin{cases} -\Omega & dl_p > dl\_r_H^i \\ \varepsilon & dl_p = dl\_r_H^i \\ [\dfrac{dl\_r_H^i - dl_p}{dl\_r_H^i - dl\_r_L^i}]^{\alpha} & \dfrac{1}{2}(dl\_r_L^i + dl\_r_H^i) \le dl_p < dl\_r_H^i \\ [\dfrac{dl\_r_H^i - dl_p}{dl\_r_H^i - dl\_r_L^i}]^{\beta} & dl\_r_L^i < dl_p < \dfrac{1}{2}(dl\_r_L^i + dl\_r_H^i) \\ 1 & dl_p \le dl\_r_L^i \end{cases} \quad (5)$$

Where $\alpha > 1$, $0 < \beta < 1$, ε and Ω have the same meanings as above. With the increase of dl_p, the user's satisfaction degree gradually decreases. Similar as in Fig. 2, the value of the delay satisfaction degree changes slowly at both ends of the interval, but changes significantly in the middle. The delay function is illustrated in Fig. 3.

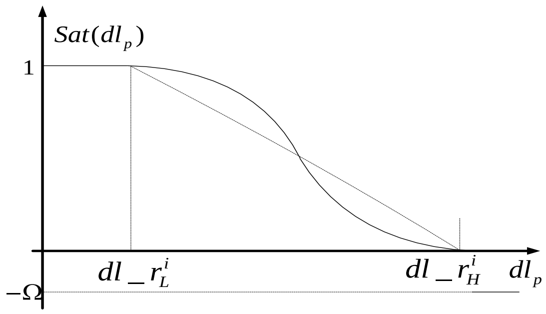

Fig. 3 Diagram of delay satisfaction degree.

Similarly as for the delay, we can design the delay jitter satisfaction degree function $Sat(jt_p)$ and the error rate satisfaction degree function $Sat(ls_p)$. By integrating the satisfaction degrees of the above four QoS parameters, we get the path general QoS satisfaction degree, $Psat$, which is calculated as in Formula 6 and Formula 7.

$$Isat = \alpha_{bw} Sat(bw_p) + \alpha_{dl} Sat(dl_p) + \alpha_{jt} Sat(jt_p) + \alpha_{ls} Sat(ls_p) \tag{6}$$

$$Psat = \begin{cases} Isat & Sat(*) > 0 \\ \sigma & o.w. \end{cases} \tag{7}$$

Where α_{bw}, α_{dl}, α_{jt}, and α_{ls} represent the weights of bandwidth, delay, delay jitter, and error rate in the general QoS satisfaction degree. $0 < \alpha_{bw}, \alpha_{dl}, \alpha_{jt}, \alpha_{ls} < 1$, and $\alpha_{bw} + \alpha_{dl} + \alpha_{jt} + \alpha_{ls} = 1$. Their values are determined according to the application types. σ is a very small positive number. Only when all the QoS requirements are satisfied, $Psat$ will achieve a meaningful value.

3.3. Evaluation Model

(1) Cost and Pricing

The cost refers to the resources spent by the network service provider on the service provision. It is a relatively stable value which can be calculated easily. The bandwidth cost is not only associated with the amount of bandwidth occupied but also related to the number of links occupied. Therefore, to further save resources, the network service provider prefers selecting the routing path with fewer links. We denote per unit time per unit bandwidth cost as c_b. The actual bandwidth assigned to a link is denoted as *ubw*. The total bandwidth cost of the multicast tree is calculated as shown in Formula 8.

$$Cost = \sum_{i \in Set_l} c_b \cdot ubw_i \tag{8}$$

Where Set_l represents the set of links belonging to the multicast tree.

Pricing refers to the procedure of setting charge rules for the usage. Pricing is a relatively complicated process. First, to fully utilize the network resources and encourage the users to use the network at off-peak time, the pricing should consider the time factor. Second, the service with higher QoS requirement should be charged at a higher level. By considering the aforementioned time and application factors, we propose the following pricing strategy as shown in Table 2.

Table 2 Pricing strategies.

	AP_1	AP_2	……	$AP_{\|APT\|}$
TZ1(21:00-07:00)	p_{11}	p_{12}	……	$p_{1\|APT\|}$
TZ2(07:00-17:00)	p_{21}	p_{22}	……	$p_{2\|APT\|}$
TZ3(17:00-21:00)	p_{31}	p_{32}	……	$p_{3\|APT\|}$

The users do not care about the number of links actually in use. They only care about the QoS satisfaction degree and the price. Therefore, we define the price of level *j* service in type *i* application within time slot *t* as shown in Formula 9.

$$price = p_{ti} \cdot (1 + Apr_i^{\,j}) \cdot ubw \tag{9}$$

(2) Multicast Tree Evaluation

In the multicast routing, since a group of users are involved in the communication, there is a high probability that multiple users share the same link. The resource sharing can reduce the price paid by each single user. Therefore, we need to recalculate the price for the multicast communication in a different way. For group member *m* requesting level *j* service of type *i* application within time period *t*, the price is recalculated by Formula 10 as below.

$$price^m = p_{ti} \cdot \frac{L_T}{\sum_{m \in G} L_{P^m}} \cdot ubw^m \cdot (1 + Apr_i^{mj}) \tag{10}$$

Where ubw^m is the actual bandwidth experienced by the user, L_T is the total number of links in the multicast tree, L_{P^m} is the number of links on the path from the source to *m*. We propose two utility formulas, i.e., the user utility as shown in Formula 11 and the network service provider utility as shown in Formula 12.

$$Uu_T = \sum_{m \in G} \frac{Pay^m - price^m}{Pay^m} \cdot Psat^m \tag{11}$$

$$Un_T = \frac{\sum_{m \in G} price^m - Cost}{\sum_{m \in G} price^m} \tag{12}$$

The standard to evaluate a multicast tree is to maximize both utilities. Therefore, we propose the following multicast tree evaluation metric as shown in Formula 13.

$$Ev_T = \frac{1}{\dfrac{\alpha_{ut}}{Uu_T} + \dfrac{\alpha_{nt}}{Un_T}} \tag{13}$$

Where α_{ut}, α_{nt} represent the weights of the user utility and the network service provider utility to the metric, respectively. $0 < \alpha_{ut}, \alpha_{nt} < 1$, $\alpha_{ut} + \alpha_{nt} = 1$. Large values of Ev_T are expected.

3.4. Problem Model

The QoS multicast routing problem can be informally described as follows. Given a source node *s*, a set of destination nodes *R*, a set of QoS constraints *C* and the optimization metrics, find the optimal routing tree which spans *s* and *R* and satisfies *C*. The mathematical model of the QoS multicast routing problem is described as below.

$$Uu_T \to \max\{Uu_T\} \tag{14}$$

$$Un_T \to \max\{Un_T\} \tag{15}$$

$$Uu_T + Un_T \to \max\{Uu_T + Un_T\} \tag{16}$$

$$s.t. \quad \min_{l \in P_{sm}}\{abw_l\} \geq bw\_r_L^{i_m} \tag{17}$$

$$\sum_{l \in P_{sm}} dl_l \leq dl\_r_H^{i_m} \tag{18}$$

$$\sum_{l \in P_{sm}} jt_l \leq jt\_r_H^{i_m} \tag{19}$$

$$1 - \prod_{l \in P_{sm}} (1 - ls_l) \leq ls\_r_H^{i_m} \tag{20}$$

$$price^m \leq Pay^m \tag{21}$$

Where i_m represents the application type that group member *m* has requested, P_{sm} represents the path from the source node v_s to *m* on the tree.

4. Cognitive Behaviours for Nodes

In this section, we describe the detailed design of cognitive behaviours for the network nodes. All these behaviours serve the QoS routing protocol aiming to optimize the routing process and improve the routing efficiency.

(1) Sensation

Through the sensation behaviour, each node maintains two tables, i.e., Table 3 and Table 4. The information in the tables is utilized by the routing protocol.

Table 3 The table of neighbour information.

Neighbour	CPU Utilization Ratio	Memory Utilization Ratio	Standby
Nid_1	$ACPU_1$	$AMEM_1$	NO
Nid_2	$ACPU_2$	$AMEM_2$	YES
……	……	……	……
Nid_n	$ACPU_n$	$AMEM_n$	NO

Table 4 The table of link information.

Reachable Node	Total Bandwidth	Available Bandwidth	Delay	Delay Jitter	Error Rate
Nid_1	tbw_1	abw_1	dl_1	jt_1	ls_1
Nid_2	tbw_2	abw_2	dl_2	jt_2	ls_2
…...	……	……	……	……	……
Nid_n	tbw_n	abw_n	dl_n	jt_n	ls_n

(2) Sense of Spatiality

To adapt to the dynamic changes of the network, the routing protocol should work in a distributed way. During the protocol running, the path probing procedure exchanges the control messages which contain partial topology information. The sense of spatiality is a procedure mainly for collecting useful topology information from the probing packets. The collected topology information is stored by the format shown in Table 5. Nid_i is the ID of node i. $lklp_i$ is a pointer pointing to the linked list which is used to store the useful routing information associated with node i. $laddr_i$ is the address of the linked list of node i. $linklist_i$ is the linked list storing the reachable nodes from node i. Since these information is dynamically updated during the path probing procedure, we use linked lists to store them.

Table 5 The table of topological information.

Node ID	Linked List Pointer	Linked List Address	Linked List of Reachable Nodes
Nid_1	$lklp_1$	$laddr_1$	$linklist_1$
Nid_2	$lklp_2$	$laddr_2$	$linklist_2$
…...	……	……	……
Nid_n	$lklp_n$	$laddr_n$	$linklist_n$

(3) Memorization

The memorization behaviour adds the experiential route information into the memory. Once a path satisfying the user's QoS requirements has been found, the control packets will travel back to the source node along the discovered path. Each visited intermediate node memorizes the indicated experiential route from it to the destination. The experiential route memorized by the source node is actually the complete path discovered between the source and the destination. The rationale is that these paths can be directly used when the same routing requests arrive next time. The format of experiential routes is shown in Table 6. The table will be updated when new information arrives.

Table 6 Information of experiential route section.

Upstream Node	Experiential Route	Bandwidth Interval	Delay Interval	Delay Jitter Interval	Error Rate Interval
Ph_1	$Path_1$	$[bw_L^1, bw_H^1]$	$[dl_L^1, dl_H^1]$	$[jt_L^1, jt_H^1]$	$[ls_L^1, ls_H^1]$
Ph_2	$Path_2$	$[bw_L^2, bw_H^2]$	$[dl_L^2, dl_H^2]$	$[jt_L^2, jt_H^2]$	$[ls_L^2, ls_H^2]$
…...	……	……	……	……	……
Ph_n	$Path_n$	$[bw_L^n, bw_H^n]$	$[dl_L^n, dl_H^n]$	$[jt_L^n, jt_H^n]$	$[ls_L^n, ls_H^n]$

(4) Learning

Learning refers to the maintenance of the memory. When links or nodes become invalid, the maintenance procedure is triggered to update the topology. In our design, the learning focuses on the maintenance of the experiential route information. For the experiential routes, the variation intervals of their QoS parameter values are estimated and provided as reference for their future use. We use the interval estimation method in the standard normal distribution to estimate the variation intervals of bandwidth, delay, delay jitter, and error rate.

Before an experiential route is used, a probing procedure is triggered to obtain its actual QoS parameter values, whic help estimate the variation interval of the corresponding QoS parameters. Formula 22 shows the calculation method.

$$[\bar{\varepsilon} - \frac{S \cdot t_{1-\alpha/2}(n-1)}{\sqrt{n-1}}, \quad \bar{\varepsilon} + \frac{S \cdot t_{1-\alpha/2}(n-1)}{\sqrt{n-1}}] \tag{22}$$

Where, $\bar{\varepsilon}$ is the expectation of the parameter value, S is the standard deviation of the parameter value, and $t_{1-\alpha/2}$ is the value of *t* distribution with confidence level α. The above estimation method applies to the bandwidth, delay, delay jitter, and error rate. Due to the dynamic changes in the network topology, only a few most recent records are kept and used for the interval estimation, avoiding the occurrence of outdated information.

(5) Reasoning

In our cognitive network, reasoning is used to perform two tasks. First, statistically summarize the usage of all the network links. The statistical results will suggest how to reconstruct and optimize the network topology. Second, speculate on the possible causes why the exceptions occur. The two tasks are directly related to the routing protocol. Each node senses the usage of its adjacent links and experiential routes. The links, which have not been in use for a long time, are deleted. Direct links are established to replace the frequently used experiential routes. To implement the topology reasoning, we create two statistical counters, i.e., NF_{sm} and NS_{sm}. NF_{sm} records the times that the neighbour interfaces have been used. Each time the routing protocol updates the routing table, for each used neighbour interface, NF_{sm} is increased by one. Every a long time period, if a certain interface has not been used, its link will be deleted and NF_{sm} is set to 0. NS_{sm} records the times that the experiential routes have been used. When the routing protocol is running, a procedure is triggered to probe the QoS information of the experiential route. The returned probing result will decide whether to use the experiential route or not. If it can be used, NS_{sm} is increased by one.

(6) Emotion

The emotion behaviour supports the routing protocol in exploiting the history information to aid the next hop selection. We use statistical method to calculate the probability that a certain neighbour node is selected as the next hop by a certain application type arriving at a certain destination node within a certain time period. The probability of selecting a certain neighbour may be often high. This information can be exploited to optimize the routing path search. The probability of selecting neighbour $Neib_{bk}$ as the next hop by application type AP_{bk} arriving at destination node v_{bk} within time period TZ_{bk} is denoted as $P(\frac{Neib_{bk}}{TZ_{bk} \wedge v_{bk} \wedge AP_{bk}})$. By the probability theory, we get the following equation as shown in Formula 23.

$$P(\frac{Neib_{bk}}{TZ_{bk} \wedge v_{bk} \wedge AP_{bk}}) = \frac{P(Neib_{bk} \wedge TZ_{bk} \wedge v_{bk} \wedge AP_{bk})}{P(TZ_{bk} \wedge v_{bk} \wedge AP_{bk})} \quad (23)$$

For each node, all its successful routing requests are stored in the format <destination, application type, next hop, time segement>, i.e., $< v_d, AP_{bk}, NH_{bk}, TZ_{bk} >$. The accumulation of history information may result in large size of data. Therefore, it is not suitable to store the history data in the memory. Instead, we store them on the hard disc and restrict the number of history data items for each node. In the protocol, if the selection probability of a neighbour exceeds a specified threshold value, the neighbour can be directly selected without any further calculation and judgement.

From the above description, we can see that sensation, sense of spatiality, and memorization are not typical of cognitive nodes. They are also usually implemented in the nodes of non-cognitive networks. However, learning, reasoning, and emotion are typical of cognitive nodes since they involve operations with higher level of cognitive capabilities. Together, the six types of cognitive behaviours form a comprehensive framework for the cognitive network nodes.

5. Design of CogMRT

Based on the cognitive behaviours designed in Section 4, we develop CogMRT, the QoS multicast routing protocol oriented to cognitive network. CogMRT works on the classic Bellman-Ford algorithm [46]. In the protocol, we have two types of probing packets, i.e., short distance probing packet and long distance probing packet. The data structures of the probing packets are shown in Table 7. G denotes the set of multicast members which include all the destination nodes $v_d^m, m \in G$.

Table 7 Structures of the probing packets

Information Carried by the Probing Packet			
Type	$PType_{mr}$	Type of Requested Application	AP_{mr}
Path Stack	$RStack_{mr}$	Set of Requested Service Levels	SL_{mr}
Maximum Hops	TTL_{mr}	Set of Link Available Bandwidth	$CurBw_{mr}$
Current Node	$CurNode_{mr}$	Set of Link Delay	$CurDl_{mr}$
Multicast Group	G	Set of Link Delay Jitter	$CurJt_{mr}$
Group Member Tag	GT	Set of Link Error Rate	$CurLs_{mr}$

5.1. QoS Routing Path Search Procedure

The following is the QoS routing path search procedure. Through it, we get a few routing paths leading to each group member.

Step 1: Start from the source node v_s, probing packets are sent to the network to search routing paths to all the group members.

Step 1.1: Initialization of each probing packet. The source node v_s is put at the bottom of the Path Stack. Set $CurBw_{mr} = \phi$, $CurDl_{mr} = \phi$, $CurJt_{mr} = \phi$, and $CurLs_{mr} = \phi$. Set the maximum hop count TTL_{mr}. Set $CurNode_{mr}$ to be v_s. Mark all the group members unreached.

Step 1.2: If $CurNode_{mr}$ has a neighbour node which belongs to the same subnet as an unreached group member v_d^m and satisfies the QoS constraints at service level SL_{mr}^m of application type AP_{mr}, a short distance probing packet is directly sent to that neighbour node, then go to Step 1.4.

Step 1.3: Look up the local experiential route information, if there exists a path to v_d^m and the path QoS satisfies the constraints, a long distance probing packet is directly sent to v_d^m through a directly connected active node. Otherwise, a short distance probing packet is sent to an active neighbour node if it meets the following conditions: not present in the Path Stack, QoS constraints being satisfied, stability degree above T_{st}, and not present in the long distance probing packet. Then go to Step 1.4. Otherwise, there is no available next hop. If all the group members have been checked, go to Step 3.

Step 1.4: When node v receives the probing packet, it first performs the sense of spatiality behaviour to get the Path Stack information. The useful topology information is extracted and stored in the memory, and TTL_{mr}=TTL_{mr}-1. If TTL_{mr}=0, discard it. Update $CurBw_{mr}$, $CurDl_{mr}$, $CurJt_{mr}$, and $CurLs_{mr}$. Update $CurNode_{mr}$. If node *v* is the group member v_d^m, then go to Step 1.6.

Step 1.5: Check $PType_{mr}$. If it is a long distance probing packet, forward the packet as required. Then go to Step 1.4. If it is a short distance probing packet, insert node *v* into the Path Stack, then go to Step 1.2.

Step 1.6: When a probing packet arrives at a group member node v_d^m, $PType_{mr}$ is checked. If it is a long distance probing packet, the actual QoS parameter values will be returned to the intermediate nodes. If it is a short distance probing packet, add v_d^m into the Path Stack and then send the packet back to the source node v_s along the reverse path in the Path Stack. All the intermediate nodes from v_d^m to v_s perform learning behaviour, i.e., memorizing or updating their experiential route information.

Step 1.7: v_d^m is marked as reached. If the current QoS parameter values still satisfy the QoS requirements of all the unreached group members, then go to Step 1.2.

Step 2: The source node v_s records all the paths returned within time interval TS_{mr} and sorts out all the paths to the reached group members.

Step 3: For each of the unreached group members, run the Bellman-Ford unicast routing algorithm to find a path. If it is successful, the search succeeds; otherwise, the search fails.

In the above procedure, Table 5, which is used to store the topology information, is updated at Step 1.4 when a node performs the sense of spatiality behaviour. Table 6, which is used to store the experiential route information, is updated at Step 1.6 when the intermediate nodes perform learning behaviour. To help understand the above procedure, we use two flowcharts to explain its two key elements, respectively. Fig. 4 shows how one node decides whether to send a long probing packet or a short probing packet. Fig. 5 shows how one node takes action when it receives a probing packet.

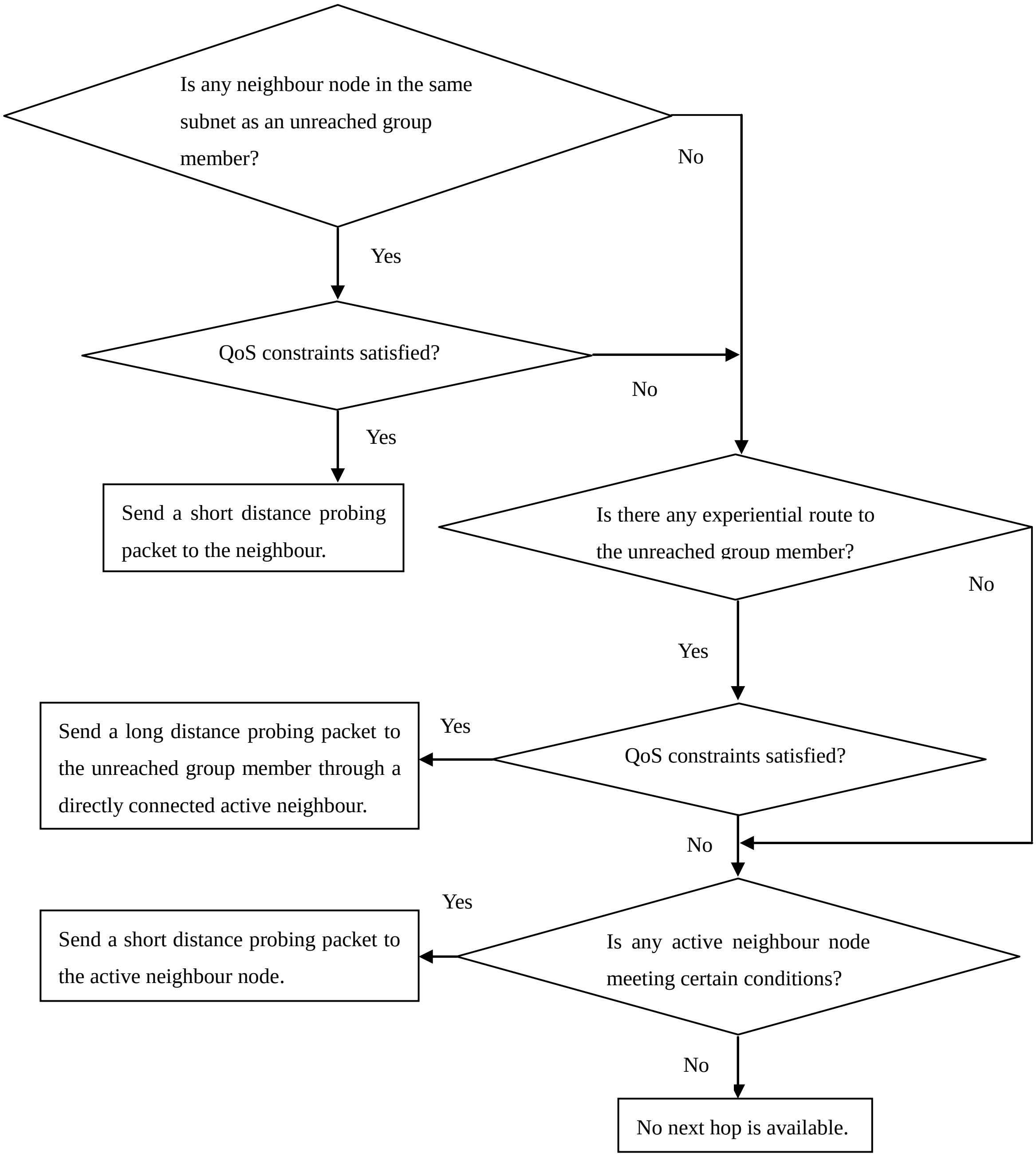

Fig. 4 The procedure for one node to decide whether to send a long probing packet or a short probing packet.

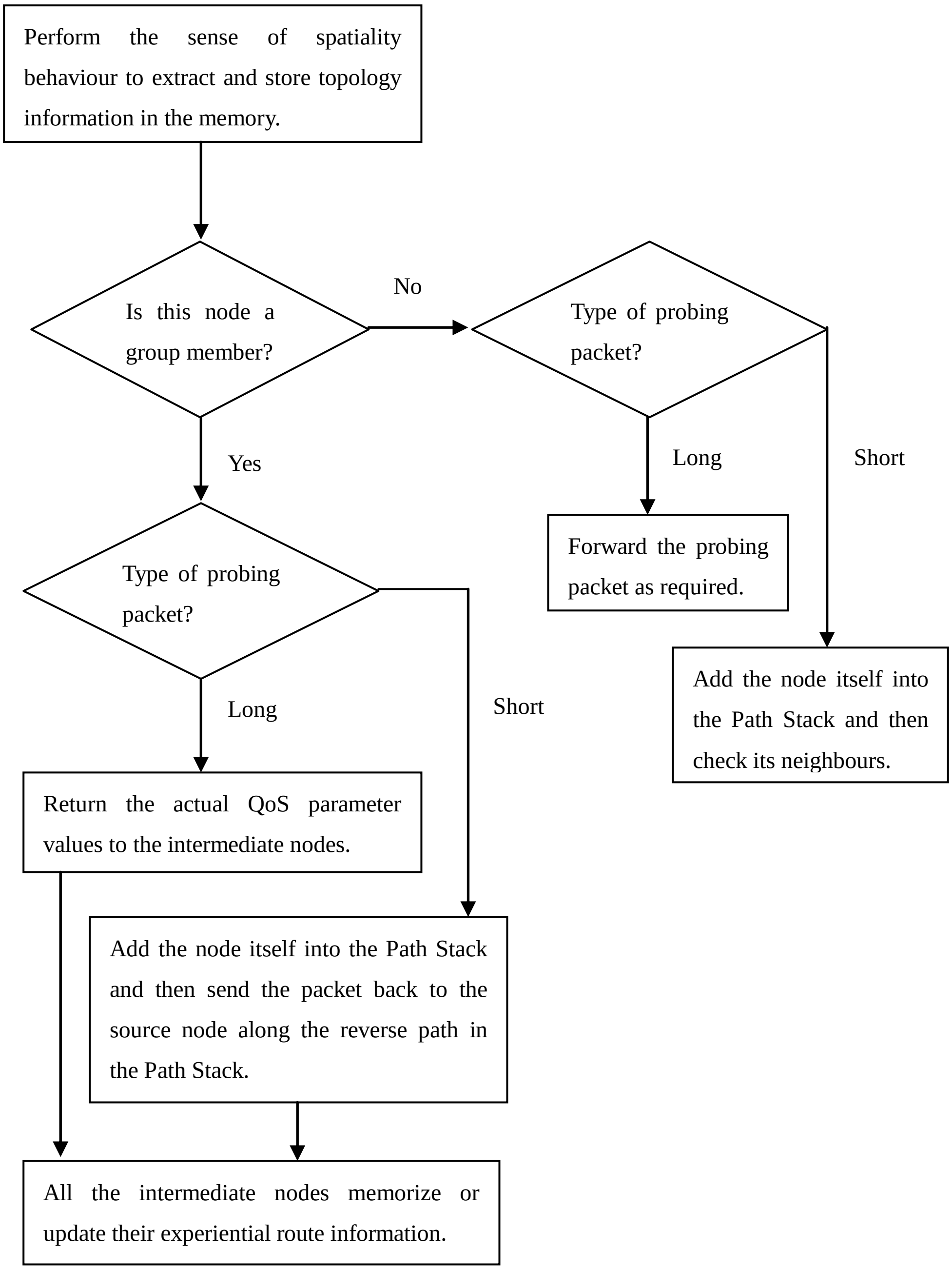

Fig. 5 The procedure for one node to deal with a received probing packet.

5.2. Multicast Tree Construction

For the multicast routing, we need to construct a multicast tree spanning both the source node and all the group members. It is a complicated process to construct the multicast tree through selecting and combing different routing paths. The reasons are three-fold. First, multiple paths may have been found to the same group member. Second, due to possible loops resulting from the combination of routing paths, the new paths after the loop removal may breach the user QoS requirements. Third, also due to possible loop removals, different sequences of the routing paths being added to the multicast tree will lead to different multicast trees. Therefore, when the size of the multicast group exceeds a certain threshold, there are huge amount of possible combinations to be considered.

We wish to find the multicast tree which can produce the best value for the multicast tree evaluation metric Ev_T specified by Formula 13. As shown in Section 3.3, the multicast tree evaluation metric has considered a number of parameters and factors such as QoS satisfaction degree, price, and cost. This problem is the same as Steiner minimum tree (SMT) problem where an SMT is also the minimum-cost multicast tree. This is a typical combinatorial optimization problem and it has been proved to be NP-Complete [34]. The problem cannot be solved exactly in polynomial time. We seek the help from the competitive coevolutionary algorithm.

(1) Encoding

We encode each solution as a dual-chromosome mode $X_{rt} \oplus Y_{ord}$, where $X_{rt} = \{x_1, x_2, \cdots, x_{|G|}\}$ and $Y_{ord} = \{y_1, y_2, \cdots, y_{|G|}\}$. They represent the selected routing paths and the joining sequence of selected routing paths, respectively. Both chromosomes have the same size equal to the number of group members. X_{rt} uses the interger coding where x_i represents the x_i th path of the i th group member. Y_{ord} uses the sequence coding where y_i represents the joining sequence of the routing path selected by the i th group member in the multicast tree.

(2) Fitness Function

X_{rt} specifies all the routing paths used to construct the multicast tree and Y_{ord} specifies the sequences of adding these paths to the tree. We use the method shown in Section 3.3 to evaluate the multicast tree. Formula 13 is used as the fitness function and Ev_T is the fitness value of $X_{rt} \oplus Y_{ord}$.

(3) Competitive Fitness

For $\forall i \in L_{GA}, \forall j \in E_{GA}$, if $Ev_i > Ev_j$, we say i beats j. Formulas 1 and 2 are used to calculate the competitive fitness CF_i of the ith learner.

(4) Crossover and Mutation

In the learner population, by the competitive fitness of each individual, the roulette wheel selection is used to select individuals from the parent generation for crossover. For two selected chromosomes, single point crossover is performed with a random cross point. Since the chromosome X_{rt} is encoded by integers, the individuals generated by the crossover operations are still feasible solutions. However, Y_{ord} is encoded by sequences and the individuals generated by crossover are possibly infeasible and need to be repaired.

Once an infeasible solution is generated, we use the partially mapped crossover to repair it. First, select a crossover point randomly and swap the gene segments before the crossover point. As shown in Fig. 6, the segment 6 1 3 is swapped with another segment 4 2 7. Then record the gene pair relationship between the two chromosomes after the crossover point. In Fig. 6, the gene pair relationship tables are shown in the text boxes. For each repeated gene in each child chromosome, replace it with the paired one in the relationship table. For example, 3 is a repeated gene and it is replaced by 7. After all the repeated genes are replaced, we get two feasible child chromosomes.

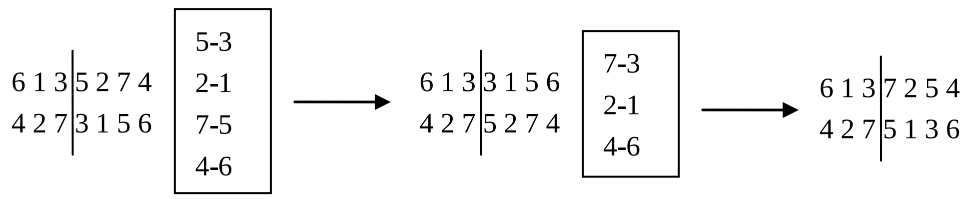

Fig. 6 Partially mapped crossover.

The mutation probability is P_{mut}. We also need to guarantee that the new chromosome generated by the mutation is feasible. The mutation to chromosome X_{rt} is to randomly select a gene, and then randomly select a different routing path among all the paths represented by this gene to replace the current one. The mutation to chromosome Y_{ord} is to randomly select a gene and move it to the first position in the chromosome. Accordingly, the other genes move backward in turn.

(5) Selection

Selection plays an important role in improving the average quality of the population by passing the high quality individuals to the next generation. The selection of individuals is based on their fitness values. Assuming that the size of the learner population is denoted as N_L, perform $N_L/2$ times of crossover and mutation operations. Then, we adopt the scheme of pair-wise tournament selection without replacement [47] as it is simple and effective. The tournament size is 2.

The procedure of CCA is described as below.

Step 1: Initialize both the learner population and the evaluator population.

Step 2: If the termination condition is met, go to Step 8.

Step 3: Calculate the competitive fitness of all the individuals in the learner population.

Step 4: Perform crossover and mutation operations over the learner population.

Step 5: Perform selection operation over the learner population.

Step 6: Update the optimal solution.

Step 7: Exchange the roles of the learner population and the evaluator population. Go to Step 2.

Step 8: The algorithm ends.

5.3. CogMRT

CogMRT consists of two steps. At Step 1, the source node looks up all the experiential routes to all the group members at itself and its neighbours. Then the source node sends probing packets to check if these route are still valid in terms of the QoS requirements. All the valid routes will be recorded for use. At Step 2, if there are no sufficient valid routes to any group member, the QoS routing path search procedure (detailed in Section 5.1) will be triggered to find more routes. For each group member, a certain number of valid routes are selected by their QoS performance. Then all the routes to all the group members will be provided to the CCA for constructing the best QoS multicast tree. The detailed procedure of CogMRT is described below.

Input: A multicast routing request for group *G* where the routing request for group member v_d^m is denoted as $R(v_s, v_d^m, AP_{mr}, SL_{mr}^m, Pay^m)$. For each group member, set the lower and upper limit for the number of candidate routing paths as N_{DT}^G and N_{UT}^G, respectively.

Step 1: Upon receiving the routing request, for each group member v_d^m:

Step 1.1: First, the source node v_s looks up the experiential routes in the local memory. If there are paths to v_d^m satisfying the QoS requirements at service level SL_{mr}^m of application type AP_{mr}^m, record the paths.

Step 1.2: Ask the neighbour nodes (in the high to low order of probabilities regulated by the emotion behaviour) to look up the experiential routes in their memories. If there are paths to v_d^m which satisfy the QoS requirements, add v_s as the source node to form new paths, and then record the paths.

Step 1.3: The source node v_s sends the probing packets to v_d^m along the paths discovered in Step 1.1 and Step 1.2. Once arriving at v_d^m, the probing packets return the acknowledgements which contain the actual QoS information of each path (learning behaviour). If the probed paths can satisfy the QoS requirements of AP_{mr}^m, record the paths at the source node and delete v_d^m from group *G*.

Step 2: For each group member v_d^m:

Step 2.1: If the number of its candidate routing paths is less than N_{DT}^G, then go to

Step 3;

Step 2.2: If the number is larger than N_{UT}^{G}, select the first N_{UT}^{G} routing paths with the best QoS performance, then go to Step 4.

Step 3: Trigger the QoS routing path search procedure to find more routing paths to v_d^m. Then go to Step 2.

Step 4: Perform the competitive coevolutionary algorithm to get the best multicast tree T_{best}. If the utilities for the network service provider and all the group members are all positive, the routing succeeds, then go to Step 5; otherwise, the routing fails.

Step 5: Reserve the resources on the final multicast tree based on the QoS requirements and set up the routing entries.

In Section 4, we have defined six types of cognitive behaviours for nodes, i.e., sensation, sense of spatiality, memorization, learning, reasoning, and emotion. The sensation is used regularly by each node to maintain the neighbour information and adjacent link information. The sense of spatiality is used at Step 1.4 in the QoS routing path search procedure. Both memorization and learning are used at Step 1.6 in the QoS routing path search procedure. The learning is also used at Step 1.3 in CogMRT. The reasoning is used for topology maintenance. Therefore, it has not been directly reflected in the QoS multicast protocol. The emotion is used at Step 1.2 in CogMRT.

6. Experimental Study

6.1. Experimental Environment Configuration

According to the generic service QoS specified in the ITU-T G.1010 standard [45], we use the following generic service classes in our simulation experiments. They are listed in Table 8 and represent four typical types of applications, i.e., telemedicine, high-quality audio, video on demand, and file transfer. In the experiments, we further classify each application type into four levels and each level corresponds to one service level of the user QoS requirements. The user request includes both the application type and the service level. By the mapping, we can get the user's detailed QoS parameter requirements. The other parameters in the routing protocol are set as in Table 9.

Table 8 Generic service class.

Application Type	Bandwidth	Delay	Delay Jitter	Error Rate	Duration Time	Application Example
App1	>3Mbps	<120ms	<10ms	0	10-90min	Telemedicine
App2	384Kbps-1.44Mbps	<250ms	<10ms	<0.01	1-30min	High-quality audio
App3	1Mbps-6Mbps	<250ms	<250ms.	<0.01	5-180min	Video on demand
App4	10Kbps-10Mbps	250ms-1s	N.A.	0	0.5-20min	File transfer

Table 9 Parameters values.

Parameters	Values
Ω, ε, α, β, σ (used in the satisfaction degree function)	1, 10^{-6}, 2, 0.5, 10^{-6}
c_b (cost), RT_{bw} (residual bandwidth)	1.5, 1000Kbps
α_{up} , α_{np} (the utility weights)	0.5, 0.5
TTL_{mr}	8
N_{DT}^{G} , N_{UT}^{G} (the lower and upper limits of the routing path to each group member)	1, 5
P_{mut} (mutation probability)	5

To evaluate the protocol performance, we must select appropriate topology examples which can reflect the practical networks. In the following experiments, we have used three practical network topologies and one random topology generated by the Waxman random graph model [48]. The three practical ones are China Education and Research Network (CERNET), CERNET2, and USA NSFNET. CERNET2 consists of 20 nodes and 22 links. NSFNET consists of 18 nodes and 27 links. CERNET consists of 36 nodes and 46 links, as shown in Fig. 7. The random topology consists of 50 nodes and 80 links, as shown in Fig. 8. They represent different sizes of networks.

A QoS-guaranteed multicast routing protocol (QGMRP) [25] was developed to solve the multi-constraints QoS multicast routing problem. QGMRP is based on the basic unicast routing protocol. It can search multiple feasible tree branches (i.e., routing paths), and select the optimal or near-optimal branch for connecting the new group member to the multicast tree. It has shown high routing success ratio. In CogMRT, the source node searches a set of candidate QoS routing paths to each group member and then use the competitive coevolutionary algorithm to select the best routing tree derived from these routing paths. For comparison purpose, we implement and evaluate CogMRT, QGMRP, and the shortest path tree (SPT) protocol in NS2.

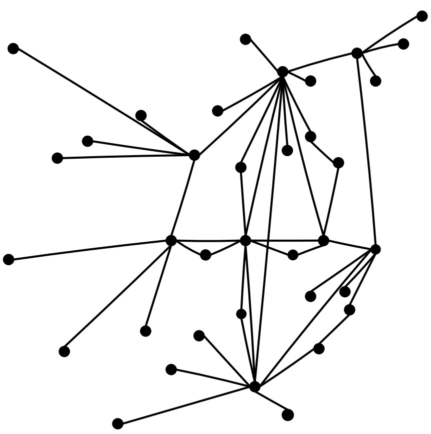

Fig. 7 Topology of CERNET.

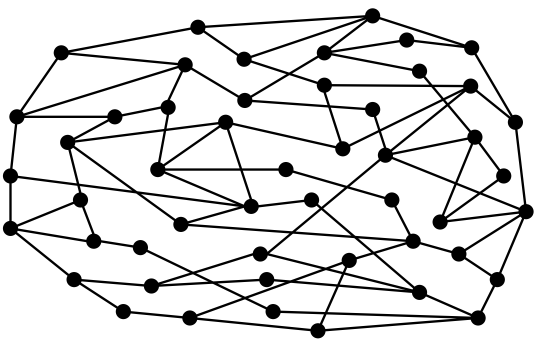

Fig. 8 Random topology based on Waxman's

6.2. Experimental Results

Since the performance of the multicast routing is directly related to the size of the multicast group, we choose five types of multicast groups with different ratios of the group size to the network size. In the experiments, we evaluate the protocol performance under the five different ratios: 10%, 20%, 30%, 40% and 50%. For each run both the source node and the multicast group members are randomly selected.

(1) Multicast Routing Success Rate

For each multicast group, we generate 100 random multicast routing requests. If the routing protocol can successfully find a multicast tree which satisfies the request, it means that the multicast routing is successful. We calculate the multicast routing success ratio. The results are shown in Figs. 9-12.

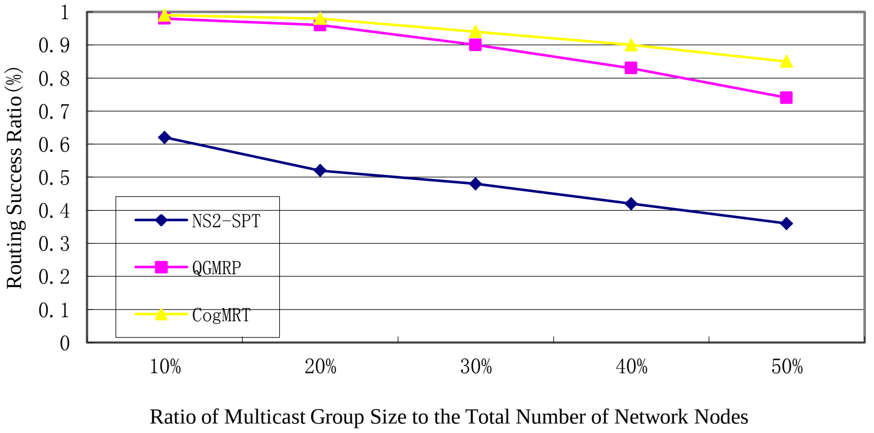

Fig. 9 Comparison of multicast routing success rates over CERNET2 Topology.

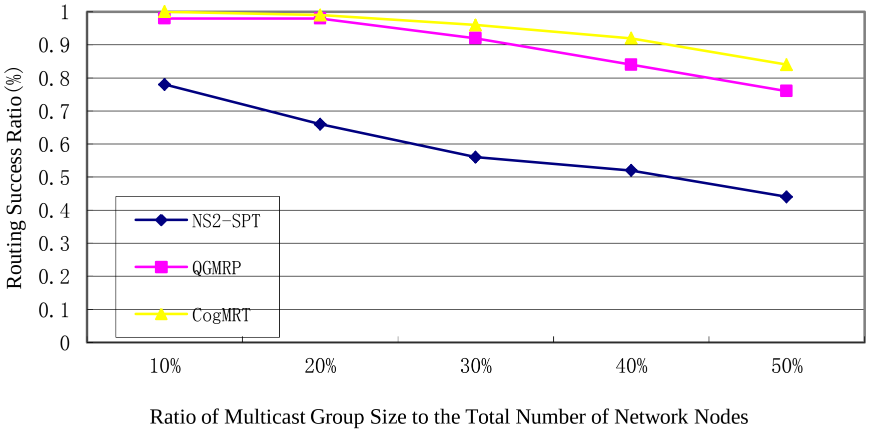

Fig. 10 Comparison of multicast routing success rates over NSFNET topology.

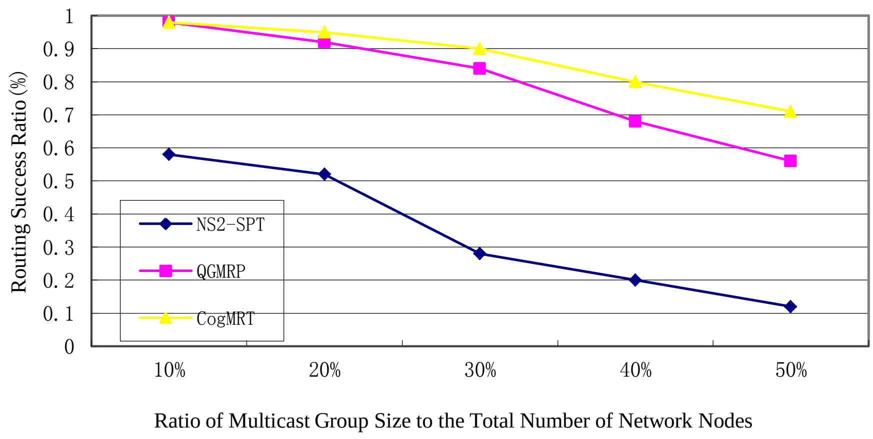

Fig. 11 Comparison of multicast routing success rates over CERNET topology.

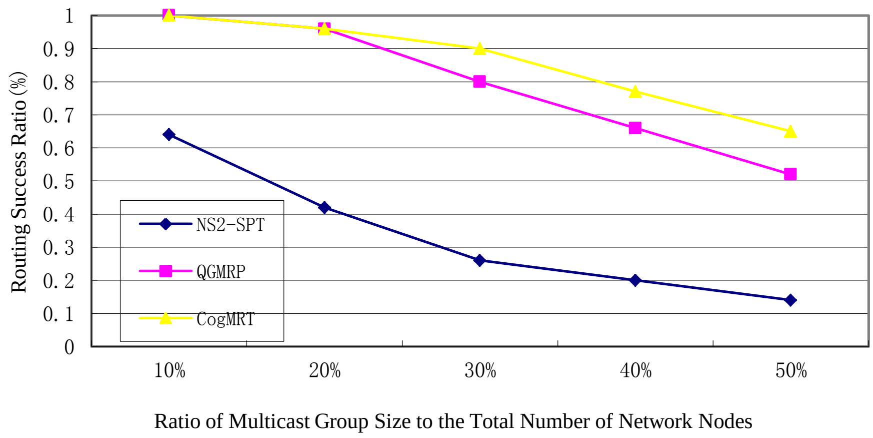

Fig. 12 Comparison of multicast routing success rates over random topology.

The results in the four networks all show that with the increase of the multicast group size, the routing success ratio decreases significantly. The primary reason is that when there are more users, there are less routing paths which can satisfy all the users' requests. Both CogMRT and QGMRP have shown similar results and their performance is much better than the SPT protocol. The reason is that both QGMRP and CogMRT have considered the users' QoS requirements but the SPT only considers the connectivity.

(2) Multicast Users' QoS Satisfaction Degree

Over the four topologies, we evaluate the users' QoS satisfaction degree for each protocol based on the results obtained in the aforementioned routing requests. Fig. 13 shows the comparison results in the four networks. We can see that CogMRT has the best performance among the three protocols in terms of the users' QoS satisfaction degree. QGMRP considers the users' QoS requests and guarantees the QoS during the running of the protocol. Therefore, it can achieve good routing success ratios. However, QGMRP does not contain any optimization procedure. It stops searching the optimal multicast tree as far as the routing paths can satisfy the user's QoS requirements. Contrarily, CogMRT has considered both the QoS requests and the multicast tree optimization.

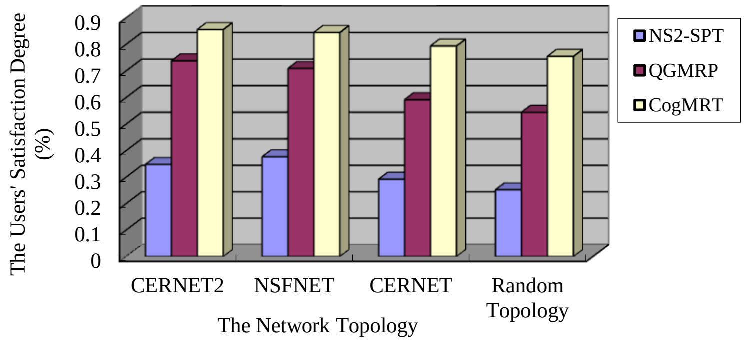

Fig. 13 The users' satisfaction degrees of multicast routing over different topologies.

(3) Utility

CogMRT has considered both the cost of the network service provider and the price paid by the user. Two utility functions are used to calculate the utilities for both parties, respectively. To evaluate the utility performance, we also calculate both utilities based on the tested routing requests. Figs. 14-16 show the results. In terms of the user utility, the network service provider utility, and the total utility, CogMRT has always shown the best performance because it has specifically considered the utilities for both parties. Contrarily, SPT has always shown the worst performance because it aims only to find the shortest paths from the source to each multicast group member. Since multiple users can share a single link in the multicast communication, the network service provider can discount the price. Therefore, the utility for the multicast users is higher than the utility for the unicast users.

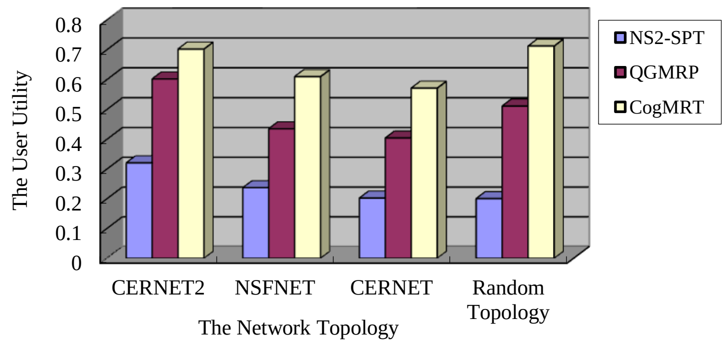

Fig. 14 Comparison of the users' utility of the multicast routing over different topologies.

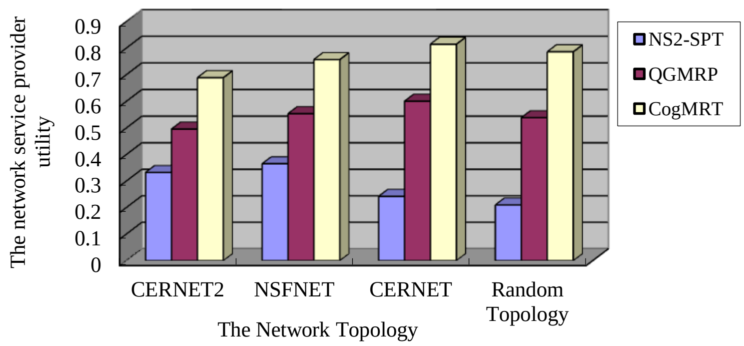

Fig. 15 Comparison of the network service provider utility of multicast routing over different topologies.

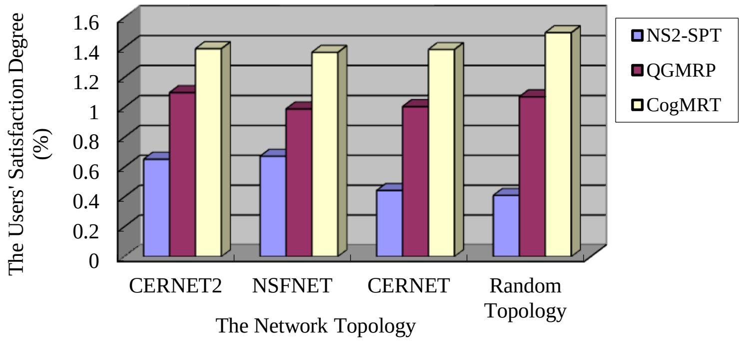

Fig. 16 Comparison of the total utility of multicast routing over different topologies.

(4) Multicast Tree Construction Time and Iteration Number

Fig. 17 shows that the time spent by CogMRT in constructing a multicast tree is much longer than the other two protocols. Because both QGMRP and SPT are based on unicast protocol, there is no path searching procedure and they only need to construct the multicast trees. Therefore, they take shorter routing time. However, CogMRT is a cognitive QoS routing protocol incorporating both the routing path searching procedure and the CCA based multicast tree construction procedure. It takes relatively longer running time. This is the price paid by CogMRT for its performance improvements in other aspects.

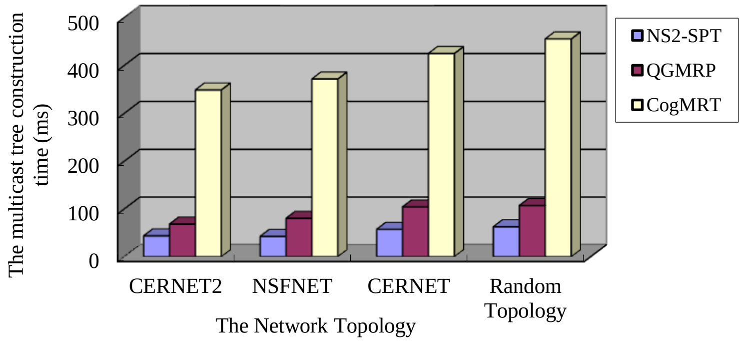

Fig. 17 Comparison of multicast tree construction time over different topologies.

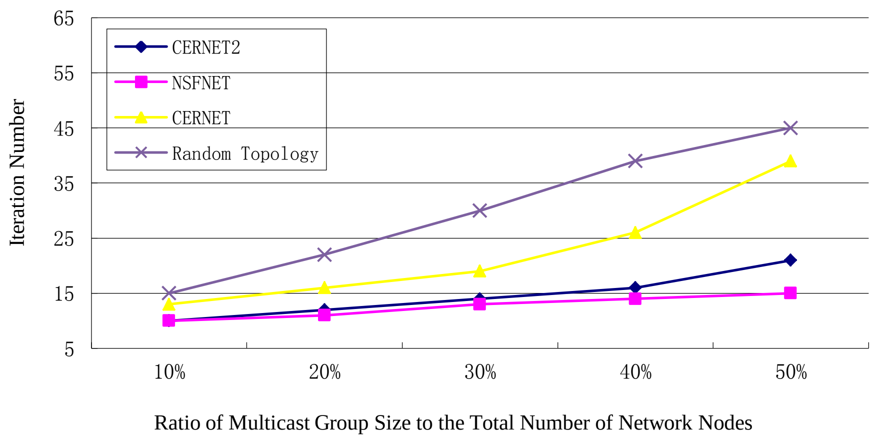

Fig. 18 Comparison of iteration number of constructing multicast trees over different topologies.

In CogMRT, a competitive coevolutionary algorithm is used to construct a multicast tree from a set of routing paths to different multicast group members. Fig. 18 shows the comparison results of the iteration number used by the competive coevolutionary algorithm in constructing a multicast tree over different topologies. We can see that with the increase in the multicast group size, the iteration number over each topology increases in an approximately linear way. The iteration number also increases as the topology size enlarges.

7. Conclusion and Future Work

The major contributions of this paper are threefold. First, we for the first time extend the concepts and techniques of cognitive science to wired backbone network. We have designed six cognitive behaviours for each network node by considering the practical requirements of interconnection networks. These behaviours are mapped to specific functions of each node for supporting the QoS routing. Our cognitive wired network architecture can be easily integrated into the current networks without bringing any extra layer. Second, we for the first time exploit the cognitive science knowledge for designing cognitive protocols in cognitive wired network. With the nodes equipped with cognitive capabilities, they can work in a distributed way to find good routes. The small-world phenomenon in the social network is also utilized by the cognitive behaviours to accumulate the experiential route information. Third, we for the first time design and apply a problem specific CCA for the cognitive multicast protocol. We have designed the encoding, fitness function, competitive fitness, crossover and mutation based on the problem characteristics. Both the user utility and the network service provider utility are considered in the fitness evaluation to guide the search towards a win-win situation.

Our work has significant practical implications. First, the proposed cognitive wired network will greatly reduce the human intervention in the network administration. The current networks have the characteristics such as large size, heterogeneity, and dynamics. These characteritiscs bring huge challenges to the network administration and maintenance, especially when the users require higher and higher QoS. By incorporating the cognitive capabilities into the network, the nodes can perform self-adaptation, self-organization, and self-protection in the network. This is extremely beneficial to the wired backbone network because it can improve the network QoS and save a lot in terms of expenditure and energy consumption. Second, the proposed cognitive multicast protocol can effectively support group communication in cognitive wired network. In wired backbone networks, there are many scenarios which require a group of routers to work in a collaborative way. The business and entertainment applications over the Internet very often involve many groups of users. By using our protocol, the routers will experience very high multicast routing success rate. The users will experience very high QoS satisfaction degree. The benefits for both the network service provider and the users will be well balanced. Third, we have introduced the advanced artificial intelligence techniques into the cognitive network management. This opens a new research frontier for both network research and artificial intelligence research in both academia and industry.

There are four solid future research directions to be considered. First, we can learn more from cognitive science and develop more cognitive capabilities for the wired network. A cognitive framework will be developed and all the cognitive behaviours will be formulated as modules under the framework. These modules can be enabled or disabled depending on the requirements of the network and the applications. This will bring more flexibility to the self-management and self-maintenance of the network. Second, we can develop some other protocols in the cognitive wired network environment, e.g., QoS routing, transmission control. The current protocols need to be well adapted to utilize the cognitive capabilities provided by the nodes and the network. Third, we can extend the cognitive capabilities of the current cognitive wireless network, which will not be limited to spectrum sensing and dynamic management in cognitive radio. The mobile nodes will also conduct cognitive behaviours to ease the network management. Last, we will implement the protocols in a prototype system. We are developing a testbed consisting of 20 prototype routers and the topology is the same as CERNET2. The proposed CogMRT and other future protocols will be tested in it.

Acknowledgment

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant No. 61225012 and No. 71325002; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant No. 20120042130003; the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20110042110024; the Fundamental Research Funds for the Central Universities under Grant No. N110204003 and No. N120104001.

References

[1] Fortuna C, Mohorcic M. Trends in the development of communication network: Cognitive networks. *Computer Networks,* 53(9): 1354-1376, 2009.

[2] Shi Y, Hou Y, Zhou H, et al. Distributed cross-layer optimization for cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59(8): 4058-4069, 2010.

[3] Mahmoud Q. *Cognitive networks: towards self-aware networks*. West Sussex: John Wiley & Sons Ltd, 2007.

[4] Mitola III J. *Cognitive radio: An integrated agent architecture for software defined radio*. Dissertation Thesis, Royal Institute of Technology (KTH), Sweden, 2000.

[5] Haykin S. Cognitive radio: brain-empowered wireless communication. *IEEE Journal on Selected Areas in Communication,* 23(2): 201-220, 2005.

[6] Clark D D, Partridge C, Ramming J C, et al. A knowledge plane for the Internet. *Proc. ACM SIGCOMM*, Karlsruhe, Germany, 3-10, 2003.

[7] Thomas R, Friend D, DaSilva L, et al. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine*, 44(12): 51-57, 2006.

[8] Wang X, Cao J, Cheng H, et al. QoS multicast routing for multimedia group communications usingintelligent computational methods. *Computer Communications*, 29(12): 2217-2229, 2006.

[9] Wang Y X, Wang Y. Cognitive informatics models of the brain. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 36(2): 203-207, 2006.

[10] Vint P. The network simulator (ns-2). Available on the WWW: http://nsnam.isi.edu/nsnam.

[11] Zhang Z W, Zhang H L. A variable-population evolutionary game model for resource allocation in cooperative cognitive relay networks. *IEEE Communications Letters*, 17(2): 361-364, 2013.

[12] Tachwali Y, Lo B F, Akyildiz I F, et al. Multiuser resource allocation optimization using

bandwidth-power product in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 31(3): 451-463, 2013.
[13] Saleh G, El-Keyi A, Nafie M. Cross-layer minimum-delay scheduling and maximum-throughput resource allocation for multiuser cognitive networks. *IEEE Transactions on Mobile Computing*, 12(4): 761-773, 2013.
[14] Karaca M, Khalil K, Ekici E, et al. Optimal scheduling and power allocation in cooperate-to-join cognitive radio networks. *IEEE/ACM Transactions on Networking*, 21(6): 1708-1721, 2013.
[15] Friend D, ElNainay M, Shi Y, et al. Architecture and performance of an island genetic algorithm-based cognitive network. *Proc. 5th IEEE Consumer Communications & Networking Conference*, Las Vegas, NV, United States, 993-997, 2008.
[16] Mahonen P, Petrova M, Riihiharvi J, et al. Cognitive wireless networks: your network just became a teenager. *Proc. 25th IEEE International Conference on Computer Communications*, Barcelona, Spain, 2006.
[17] Fortuna C, Mohorcic M. Advanced access architecture for efficient service delivery in heterogeneous wireless networks. *Proc. 3rd International Conference on Communications and Networking in China*, Hangzhou, China, 1275-1279, 2008.
[18] Kim M, Choo H, Mutka M W, et al. On QoS multicast routing algorithms using k-minimum Steiner trees. *Information Sciences*, 238: 190-204, 2013.
[19] Qu H, Yi Z, Yang S X. Efficient shortest-path-tree computation in network routing based on pulse-coupled neural networks. *IEEE Transactions on Cybernetics*, 43(3): 995-1010, 2013.
[20] Maia G, Aquino A, Guidoni D, et al. A multicast reprogramming protocol for wireless sensor networks based on small world concepts. *Journal of Parallel and Distributed Computing*, 73(9): 1277-1291, 2013.
[21] Kim W, Gerla M. Cognitive multicast with partially overlapped channels in vehicular ad hoc networks. *Ad Hoc Networks*, 11(7): 2016–2025, 2013.
[22] Chen Y-R, Radhakrishnan S, Dhall S, et al. On multi-stream multi-source multicast routing. *Computer Networks*, 57(15): 2916-2930, 2013.
[23] Xie L, Jia X, Zhou K. QoS multicast routing in cognitive radio ad hoc networks. *International Journal of Communication Systems*, 25(1): 30-46, 2012.
[24] Wang Y, Wu J, Yang W-S. Cloud-based multicasting with feedback in mobile social networks. *IEEE Transactions on Wireless Communications*, 12(12): 6043-6053, 2013.
[25] Li L, Li C. A QoS-guaranteed multicast routing protocol. *Computer Communication*, 27(1): 59-69, 2004.
[26] Li B, Lin T, Liao L, et al. Genetic algorithm based on multipopulation competitive coevolution. *Proc. IEEE Congress on Evolutionary Computation*, Hong Kong, China, 225-228, 2008.
[27] Zhan Z-H, Li J, Cao J N, et al. Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems. *IEEE Transactions on Cybernetics*, 43(2): 445-463, 2013.
[28] Wang R, Purshouse R C, Fleming R J. Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(4): 474-494, 2013.
[29] Ladjici A A, Tiguercha A, Boudour M. Nash Equilibrium in a two-settlement electricity

market using competitive coevolutionary algorithms. *International Journal of Electrical Power and Energy Systems*, 57: 148-155, 2014.
[30] Wang L, Li L-P. A coevolutionary differential evolution with harmony search for reliability-redundancy optimization. *Expert Systems with Applications*, 39(5): 5271-5278, 2012.
[31] Qu H, Xing K, Alexander T. An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120(23): 509-517, 2013.
[32] Kala R. Multi-robot path planning using co-evolutionary genetic programming. *Expert Systems with Applications*, 39(3): 3817-3831, 2012.
[33] Rabil B S, Tliba S, Granger E, et al. Securing high resolution grayscale facial captures using a blockwise coevolutionary GA. *Expert Systems with Applications*, 40(17): 6693-6706, 2013.
[34] Rango F, Tropea M, Santamaria A, et al. Multicast QoS core-based tree routing protocol and genetic algorithm over an HAP-satellite architecture. *IEEE Transactions on Vehicular Technology*, 58(8): 4447-4461, 2009.
[35] Cheng H, Yang S. Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks. *Engineering Applications of Artificial Intelligence*, 23(5): 806-819, 2010.
[36] Xing H, Qu R. A nondominated sorting genetic algorithm for bi-objective network coding based multicast routing problems. *Information Sciences*, 233: 36-53, 2013.
[37] Lu T, Zhu J. Genetic algorithm for energy-efficient QoS multicast routing. *IEEE Communications Letters*, 17(1): 31-34, 2013.
[38] Ahn C, Ramakrishna R. A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6): 566-579, 2002.
[39] Chitra C, Subbaraj P. A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks. *Expert Systems with Applications*, 39(1): 1518-1525, 2012.
[40] Yang S, Cheng H, Wang F. Genetic algorithms with immigrants and memory schemes for dynamic shortest path problem in mobile ad hoc networks. *IEEE Transactions on Systems, Man, and Cybernetics--Part C: Applications and Reviews*, 40(1): 52-63, 2010.
[41] Cheng H, Yang S, Cao J. Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc network. *Expert Systems With Applications*, 40(4): 1381-1392, 2013.
[42] Cheng H, Yang S. Joint QoS multicast routing and channel assignment in multiradio multichannel wireless mesh networks using intelligent computational methods. *Applied Soft Computing*, 11(3): 1953-1964, 2011.
[43] Cheng H, Wang X W, Yang S, et al. A multipopulation parallel genetic simulated annealing based QoS routing and wavelength assignment integration algorithm for multicast in optical networks. *Applied Soft Computing*, 9(2): 677-684, 2009.
[44] Blake S, Black D, Carlson M, et al. An architecture for differentiated services. IETF draft, draft-ietf-diffserv-arch-01.txt, Aug. 1998.
[45] ITU-T G.1010, End-user multimedia QoS categories.
[46] Cormen T, Leiserson C, Rivest R, et al. *Introduction to Algorithm*. 2nd Edition, MIT

Press, 2001.
[47] Lee S, Soak S, Kim K, et al. Statistical properties analysis of real world tournament selection in genetic algorithms. *Applied Intelligence,* 28(2): 195-205, 2008.
[48] Haddadi H, Fay D, Jamakovic A, et al. On the importance of local connectivity for Internet topology models. *Proc. 21st International Teletraffic Congress*, Paris, France, 1-8, 2009.