

Quality of Service Rerouting in Cognitive Network

Y.B.Gandole¹

Abstract: The cognitive behavior based on the cognitive science model for the network nodes to benefit the Quality of Service rerouting is designed. A Quality of Service rerouting protocol oriented to cognitive network is proposed. Rerouting is a distributed protocol where the route search is in a hop by hop way. Inspired by the small-world phenomenon, the experiential route information is collected and stored at each node to benefit the future route selection. We implement and evaluate Quality of Service rerouting in NS2 platform. Its performance is compared with another two popular routing protocols. The results show that Cognitive Rerouting has achieved remarkable performance improvements over the protocols where no cognitive behaviours are exploited.

Keywords: Cognitive Network, Quality of Service routing, rerouting, Cognitive behavior

1 Introduction

Nowadays, the users have high-end demand of the network services which are far beyond the scope of the traditional data service. The spatial and temporal complexity of the network has worsened due to the everlasting expansion of the Internet. With the rapid development in networking technologies, the next generation 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. Since the network is not aware of its own states and requirements, the network management becomes an extremely difficult task. However, if the network elements can intelligently adapt to the operations, the increased complexity can be alleviated without much extra resource consumption. Reliability, security, scalability and Quality-of-Service have become key issues as we envision the future Internet. In recent years, the cognition concept has been applied to various net-

¹ Department of Electronics, Adarsha Science J.B.Arts and Birla Commerce Mahavidyalaya Dhamangaon Rly-444709 (India).
Email:ygandole@indiatimes.com

work and communication systems. Two new terms were created to reflect the new technologies, i.e., cognitive radio and cognitive network. The cognitive science has brought new features to the renovation of traditional network protocols. In the cognitive network, we should exploit these features to design meaningful cognitive behaviors for enhancing the scalability, self-adaptivity and self-protection of the network. In this paper, we investigate the Quality-of-Service routing and rerouting problem [Pragyansmita and Raghavan(2002)] in the context of cognitive network environment. Referring to the brain model [Wang(2006)], the research on the cognitive network mainly focuses on cognitive radio which deals with dynamic management of spectrum resources. The cognitive radio has been applied to the practical wireless communication system. Research interests have been drawn on the cross-layer optimization and design in cognitive radio network. In the cross-layer approach, the information is shared among multiple layers instead of only between two neighboring layers as in the traditional layered approach. Distributed learning and reasoning is used to optimize the network operations. The island genetic algorithm is applied to optimize the channel assignment in the dynamic spectrum access. The primary objective of Quality-of-Service routing [Adrian, Gerald and Bruce(2008)] is to select the transmission path which satisfies the Quality-of-Service requirements of the application. In the autonomous system level topology, the small-world phenomenon is caused by the highly variable node degrees. In the router level topology, the local connection preference of nodes contributes significantly to the small-world phenomenon. In the Internet, the small-world phenomenon has revealed two findings, i.e., the short chains exist and they can be identified. When the network exhibits certain topologies, the local information can help find the short chains efficiently. The findings have inspired the distributed routing in the message delivery from the message holder to its direct interactors. The above information provides guideline for the design of the routing protocol. In this paper, we use the small-world phenomenon to optimize the distributed routing design. We design the cognitive behaviors for the network nodes in the proposed protocol named as Cognitive Rerouting. Performance evaluation shows that the new rerouting mechanisms have remarkable advantages over the current routing mechanisms. The cognitive network model is designed by exploiting the idea of knowledge plane. The model can 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.

2 Model development

2.1 User's Quality-of-Service Requirements Model

To address Quality-of-Service routing comprehensively, we consider as many Quality-of-Service 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 Quality-of-Service parameters on its adjacent links. In the search of Quality-of-Service routing paths, we should consider the current load status of the nodes. The stability degree Sd is a novel Quality-of-Service parameter to represents it. If the load of one node is too heavy, the routing path should bypass it. The stability degree Sd of the node is defined as:

$$Sd = \text{Min} \left\{ \frac{Ct}{T}, \frac{AM}{TM} \right\} \quad (1)$$

Where Ct is the available CPU cycles, T is the total CPU cycles, AM is the available memory, and TM is the total memory. Large values of Sd are expected. The user's requirements refer to the Quality-of-Service parameters specified by the user. We classify the network applications into different categories based on the DiffServ model [Blake(1998)]. Each application category is supported by a certain set of Quality-of-Service parameters. The mapping relationship is formulated by ITU-T G.1010 [End-user multimedia Quality-of-Service categories]. Since in practice the Quality-of-Service requirements could not be a fixed value, we represent them by intervals. Consider the set of application types as:

$$A_t = \{ AP_1, AP_2, \dots, AP_n \}$$

Each application type is associated with a set of Quality-of-Service requirements. For example, for AP_i , the set of Quality-of-Service requirements is denoted as :

$$Ar_i = (\delta_{BW}^i, \delta_{DL}^i, \delta_{DJ}^i, \delta_{ER}^i)$$

Among Ar_i , bandwidth, delay, delay jitter, and error rate requirements are represented by intervals $\delta_{BW}^i = [BW_L^i, BW_H^i]$, $\delta_{DL}^i = [DL_L^i, DL_H^i]$, $\delta_{DJ}^i = [DJ_L^i, DJ_H^i]$, $\delta_{ER}^i = [ER_L^i, ER_H^i]$ respectively. Four different service levels are provided for each application type such as for level-1, bandwidth $[BW_L^1, BW_H^1]$, Delay $[DL_L^1, DL_H^1]$, Delay Jitter $[DJ_L^1, DJ_H^1]$, error rate $[ER_L^1, ER_H^1]$ and Extra cost EC^1 etc. For each service level, the parameter values must fall into the corresponding intervals specified by $\delta_{BW}^i, \delta_{DL}^i, \delta_{DJ}^i, \delta_{ER}^i$.

2.2 User's Quality-of-Service Satisfaction Degree Model

Each Quality-of-Service parameter requirement is represented by an interval value instead of a single value.

(1) Bandwidth Satisfaction Degree Function

In terms of the bandwidth, the user always expects to get the largest value. Consider the bandwidth requirement interval as $[BW_L^i, BW_H^i]$. When the actual bandwidth of a routing

path is BW_p , the user's bandwidth satisfaction degree function $BWsd$ is defined as in equations (2).

$$\begin{aligned}
 f_{sat}(BW_p) &= -\rho, & BW_p < BW_L^i \\
 &= \mu, & BW_p = BW_L^i \\
 &= \left(\frac{BW_p - BW_L^i}{BW_H^i - BW_L^i} \right)^\alpha, & BW_L^i < BW_p < \left(\frac{BW_H^i + BW_L^i}{2} \right) \\
 &= \left(\frac{BW_p - BW_L^i}{BW_H^i - BW_L^i} \right)^\beta, & \left(\frac{BW_H^i + BW_L^i}{2} \right) \leq BW_p < BW_H^i \\
 &= 1, & BW_p \geq BW_H^i
 \end{aligned} \tag{2}$$

Where $\alpha > 1$, $0 < \beta < 1$, μ is a very small positive integer, and ρ is a penalty value. The penalty value will be applied only when the user's Quality-of-Service request cannot be satisfied even at the lower bound of the interval. With the increase of BW_p , the user's Quality-of-Service satisfaction degree also gradually increases.

(2) Delay Satisfaction Degree Function

In terms of the delay, the user always expects to get the smallest value. We denote the delay requirement interval as $[DL_L^i, DL_H^i]$. When the actual delay of the routing path is DL_p , the user's delay satisfaction degree function is defined as in equations (3).

$$\begin{aligned}
 f_{sat}(DL_p) &= -\rho, & DL_p > DL_H^i \\
 &= \mu, & DL_p = DL_H^i \\
 &= \left(\frac{DL_H^i - DL_p}{DL_H^i - DL_L^i} \right)^\alpha, & \left(\frac{DL_H^i + DL_L^i}{2} \right) \leq DL_p < DL_H^i \\
 &= \left(\frac{DL_H^i - DL_p}{DL_H^i - DL_L^i} \right)^\beta, & DL_L^i < DL_p < \left(\frac{DL_H^i + DL_L^i}{2} \right) \\
 &= 1, & DL_p \leq DL_L^i
 \end{aligned} \tag{3}$$

Where $\alpha > 1$, $0 < \beta < 1$, μ is a very small positive integer, and ρ is a penalty value. With the increase of $p dl$, the user's Quality-of-Service satisfaction degree gradually decreases.

2.3 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 routing path is calculated as shown in equation (4).

$$Cost = \sum_{i \in Set_l}^{c_b \times ubw_i} \quad (4)$$

Where Set_l represents the set of links belonging to the routing path. Pricing refers to the procedure of setting charge rules for the usage. By considering the time and application factors, we propose the following pricing strategy as shown in Table 1.

Table 1: Pricing strategies

Time Zone	AP ₁	AP ₂	—	AP _n
TZ1	P_{11}	P_{12}	—	P_{1n}
TZ2	P_{21}	P_{22}	—	P_{2n}
TZ3	P_{31}	P_{32}	—	P_{3n}

Equation (5) shows the price of leve j service in type i application within time slot t .

$$price = p_{ti} \times (1 + EC_t^j) \times ubw \quad (5)$$

(2) Path Evaluation

Equation (6) shows the user utility and equation (7) shows the network service provider utility, respectively.

$$u_p = \frac{Pay - Price}{Pay} \times P_{sat} \quad (6)$$

$$n_p = \frac{Price - Cost}{Price} \quad (7)$$

The path evaluation standard is to maximize both the user utility and the network service provider utility. the path evaluation metric is shown in equation (8).

$$E_{vp} = \frac{1}{\frac{\alpha_{up}}{u_p} + \frac{\alpha_{np}}{n_p}} \quad (8)$$

Where α_{up} , α_{np} represent the weights of the user utility and the network service provider utility to the path evaluation metric, respectively. $0 < \alpha_{up}, \alpha_{np} < 1$, $\alpha_{up} + \alpha_{np} = 1$. Large value of E_{vp} is always expected.

2.4 Problem Model

Consider the Quality-of-Service routing request as $R(n_s, n_d, AP_i, SL_i, Pay)$ Where n_s is source node, n_d is destination node, AP_i represents the application type requested by the user, SL_i represents the service level requested by the user, and Pay represents the upper limit cost that the user is willing to pay. The Quality-of-Service routing request aims to find a routing path P_{sd} from n_s to n_d . The path should support Quality-of-Service at service level SL_i^m of application type AP_i . The path cost should not be greater than Pay . The objective of Quality-of-Service routing problem is to maximize the user utility, the network service provider utility, and their summation under the Quality-of-Service constraints. The mathematical model of Quality-of-Service routing problem is described as below equations (9 -16).

$$u_p \rightarrow \max \{u_p\} \quad (9)$$

$$n_p \rightarrow \max \{n_p\} \quad (10)$$

$$u_p + n_p \rightarrow \max \{u_p + n_p\} \quad (11)$$

$$s.t. \min_{l \in P_{sd}} \{abw_j\} \geq BW_L^i \quad (12)$$

$$\sum_{l \in P_{sd}} DL_l \leq DL_H^i \quad (13)$$

$$\sum_{l \in P_{sd}} DJ_l \leq DJ_H^i \quad (14)$$

$$1 - \prod_{l \in P_{sd}} (1 - ER_l) \leq ER_H^i \quad (15)$$

$$price \leq Pay \quad (16)$$

2.5 Design of Cognitive Behaviors

To design a cognitive network, we need to design cognitive behaviors for the network nodes. These behaviors are reflected in the newly added knowledge plane and supported by the memory. The memory stores both the topology information and the experiential information accumulated during the protocol running. In the following, we develop primary cognitive behaviors for routing and rerouting.

2.6 Cognitive Behaviors for Routing

(1) Sensation

Through the sensation behavior, each node maintains two tables, i.e., the table of neighbour information and the table of link information. They are shown in Table 2 and Table 3, respectively.

Table 2: Neighbour information.

Neighbour	CPU Utilization Ratio	Memory Utilization Ratio	Standby
Nid_1	Ct_1	AM_1	NO
Nid_2	Ct_2	AM_2	YES
—	—	—	—
Nid_n	Ct_n	AM_n	No

Table 3: Link information.

Reachable Node	Total Bandwidth	Available Bandwidth	Delay	Delay Jitter	Error Rate
Nid_1	TBW_1	ABW_1	DL_1	DJ_1	ER_1
Nid_2	TBW_2	ABW_2	DL_2	DJ_2	ER_2
—	—	—	—	—	—
Nid_n	TBW_n	ABW_n	DL_n	DJ_n	ER_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 4.

Table 4: topology information.

Node ID	Link List Pointer	Link List Address	Link List of Reachable Nodes
Nid_1	LLP_1	LLA_1	LL_1
Nid_2	LLP_2	LLA_2	LL_2
—	—	—	—
Nid_n	LLP_n	LLA_n	LL_n

(3) Memorization

The memorization behavior adds the experiential route information into the memory. The experiential route information is used to speed up the path probing procedure. Once a path satisfying the user's Quality-of-Service requirements has been found, the control packets will travel back to the source node along the discovered path. Meanwhile, all the intermediate nodes are recorded. Each visited intermediate node memorizes the indicated experiential route from itself to the destination. The experiential route memorized by the source node is actually the complete path discovered between the source node and the destination node. The rationale is that these paths can be directly used when the same routing requests arrive next time. The experiential routes have the format shown in Table 5. The table is updated when new information arrives.

Table 5: Information of experiential route section.

Upstream Node	Experiential Route	Bandwidth Interval	Delay Interval	Delay Jitter Interval	Error Rate Interval
Ph_1	$Path_1$	$[bw_L^1, bw_H^1]$	$[dl_L^1, dl_H^1]$	$[dj_L^1, dj_H^1]$	$[er_L^1, er_H^1]$
Ph_2	$Path_2$	$[bw_L^2, bw_H^2]$	$[dl_L^2, dl_H^2]$	$[dj_L^2, dj_H^2]$	$[er_L^2, er_H^2]$
—	—	—	—	—	—
Ph_n	$Path_n$	$[bw_L^n, bw_H^n]$	$[dl_L^n, dl_H^n]$	$[dj_L^n, dj_H^n]$	$[er_L^n, er_H^n]$

(4) Maintenance of the contents in the memory

When links or nodes become invalid, the maintenance procedure is triggered to update the topology information. In our design, the learning focuses on the maintenance of the experiential routes. The variation intervals of their Quality-of-Service parameter values are estimated and provided as reference for 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 the actual Quality-of-Service parameter value of this routing path. The actual value helps estimate the

variation interval of the corresponding Quality-of-Service parameter. Formula 17 shows the calculation method.

$$\left[\bar{\mu} - \frac{S \cdot t_{1-\alpha/2}(n-1)}{\sqrt{n-1}}, \bar{\mu} + \frac{S \cdot t_{1-\alpha/2}(n-1)}{\sqrt{n-1}} \right] \quad (17)$$

Where $\bar{\mu}$ 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 of the network topology, only the most recent records are kept and used for the interval estimation to avoid the influence of the outdated information. In addition, for each experiential route we set two counters, i.e., $ExpSN$ and $ExpFN$, to record its successful times and failure times, respectively. We set a threshold value for the failure times. If the probing of an experiential route returns satisfactory Quality-of-Service information, $ExpSN$ is increased by one; otherwise, $ExpFN$ is increased by one, $ExpSN = 0$. When the failure threshold is reached, the experiential route is deleted.

(5) Reasoning

In our cognitive network, reasoning is used to perform two tasks. First, statistically summarize the usage of all the links in the network. The statistical results are used by the network administrator to reconstruct and optimize the network topology. Second, speculate on the possible causes when the exceptions occur. The two tasks are directly related to the routing protocol. Each node senses the usage of both adjacent links and experiential routes. The links, which have not been used for a long time, will be deleted by the network administrator. The frequently used experiential routes will be replaced by direct links. To implement the above topology reasoning, we create two statistical counters, i.e., $NF\ sm$ and $NS\ sm$, to record the times that the neighbour interfaces have been used and the times that the experiential routes have been used, respectively. Each time the routing protocol updates the routing table, $NF\ sm$ is increased by one for each used neighbour interface. If a certain interface has not been used for a long time period, its link will be deleted and its $NF\ sm$ is set to 0. When the routing protocol is running, a procedure is triggered to probe the Quality-of-Service information in the experiential route. The returned probing result will decide whether to use the path or not. If the experiential route can be used, $NS\ sm$ is increased by one.

(6) Emotion

The emotion behavior exploits the history routing requesting information to aid the next hop selection. The network traffic follows a time-dependent pattern. Statistical method is used 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 search. The probability of selecting neighbor $Neib_{bk}$ as the next hop by application type AP_{bk} arriving at destination node v_{bk} .

within time period TZ_{bk} is denoted as $P\left(\frac{Neib_{bk}}{TZ_{bk} \wedge v_{bk} \wedge AP_{bk}}\right)$. By the probability theory, we get the following equation (18)

$$P\left(\frac{Neib_{bk}}{TZ_{bk} \wedge v_{bk} \wedge AP_{bk}}\right) = \frac{P(Neib_{bk} \wedge TZ_{bk} \wedge v_{bk} \wedge AP_{bk})}{P(TZ_{bk} \wedge v_{bk} \wedge AP_{bk})} \quad (18)$$

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

2.7 Cognitive Behaviours for Rerouting

The cognitive behaviors for re-routing include problem solving, reasoning, and interpretation. The problem solving behavior is responsible for calling the exception handler. The reasoning behavior consists of two processes. One is to find out the reason why the exception occurs, and the other is to obtain the exception status. The process for exception reasoning is included in the rerouting protocol. When the system recovers the exception status, the network administrator needs to be notified the reason why the exception occurs. This job is done by the interpretation behaviour. In addition, lots of experiential knowledge was ‘memorized’ at the nodes through the cognitive behaviours. However, the exceptions in the network are usually caused by the node or link failures, leading to invalidation of the contents in the memory. The interpretation behaviour needs to notify the nodes to update their experiential knowledge in addition to giving the reasons of the exceptions. There are four types of information carried by the interpretation behaviour, i.e., original route, new route, exceptional upstream node, and exception type. The interpretation is an operation following rerouting. The destination node sends back the interpretation information upstream along the feasible partial paths on the original route and the new route, respectively. Each node creates a set of unavailable routes. Upon receiving the interpretation information, each node conducts the following processing (Fig.1)

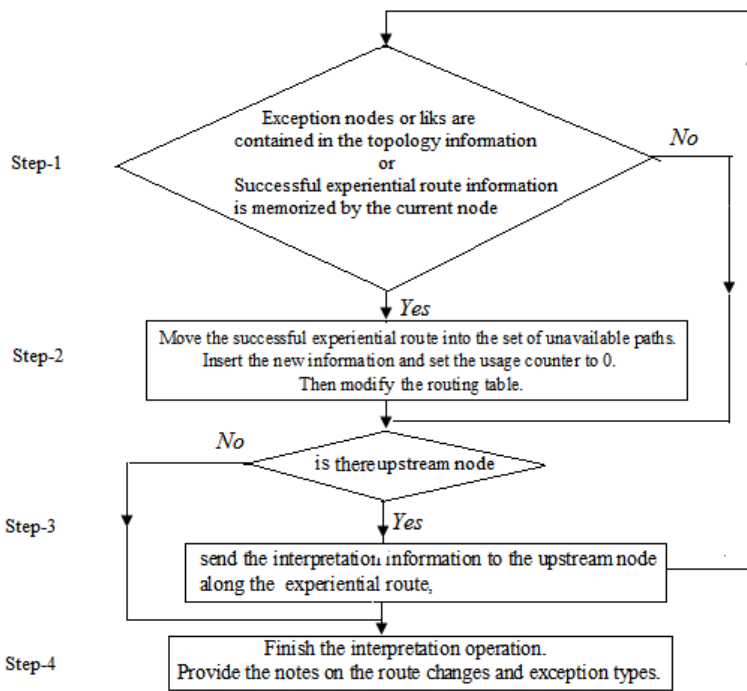


Figure 1: Processing flow of Cognitive Behaviours for Rerouting

The set of unavailable routes are regularly probed. If some routes turn to be available, move them into the successful experiential route set. If they are still unavailable, delete them. Meanwhile, the protocol detects the freshness of the information before using it. Thus, the obsolete information is eliminated. The information inconsistency will not last long between different nodes and no negative effect is caused by the existence of obsolete information.

2.8 Design of the Cognitive Routing and Rerouting Protocols

Based on the aforementioned cognitive behaviours, we develop the Quality-of-Service routing and rerouting protocol oriented to cognitive network, which works on the classic Bellman-Ford algorithm. In the protocol, we have two types of probing packets, i.e., the short distance probing packet and the long distance probing packet. The short distance probing packets are just sent to the neighbour nodes while the long distance probing packets are forwarded along the path set out by the probing packets.

2.9 Quality-of-Service Routing Path Search Procedure

Step 1: The source node v_s starts to send the probing packets which contain the Quality-of-Service requirements in terms of bandwidth, delay, delay jitter, and error rate.

Step 1.1: Initialization. For each probing packet, put the source node v_s at the bottom of the Path Stack, and set $CurBW_{ur} = 0$, $CurDL_{ur} = 0$, $CurDJ_{ur} = 0$, and $CurER_{ur} = 0$. Set the maximum hop number TTL_{ur} to a default value.

Step 1.2: If one of the neighbour nodes belongs to the same subnet as the destination node v_d and it satisfies the Quality-of-Service constraints at service level SL_{ur} of application type AP_{ur} , short distance probing packet is directly sent to the neighbour node, then go to Step 1.3. Otherwise, look up the local experiential route information, if there exists a path to the destination node and the path satisfies the Quality-of-Service constraints, a long distance probing packet is directly sent to the destination node through an active neighbour node. Then short distance probing packets are sent to all the other neighbour nodes if they meet the following conditions: being active, not present in the Path Stack, satisfying the Quality-of-Service constraints, stability degree above the threshold. If all of them are in the standby mode, they will be activated first and then receive the probing packets, then go to Step 1.3. Otherwise, there is no next hop node which meets the conditions, and the probing packet is discarded.

Step 1.3: When node v receives a probing packet, it first performs the sense of spatiality behaviour to get the Path Stack information and then extracts the useful topology information which will be stored in the memory. $TTL_{ur} \equiv TTL_{ur} - 1$. If $TTL_{ur} = 0$, discard the probing packet. Update $CurBW_{ur}$, $CurDL_{ur}$, $CurDJ_{ur}$, and $CurER_{ur}$. If node v is the destination node, then go to Step 1.5; otherwise, go to Step 1.4.

Step 1.4: Check $PType_{ur}$. If it is a long distance probing packet, update $CurNode_{ur}$ and forward the packet along the path indicated by the packet, then go to Step 1.3. If it is a short distance probing packet, add node v into the Path Stack, then go to Step 1.2.

Step 1.5: When a probing packet arrives at the destination node, $PType_{ur}$ is checked. If it is a long distance probing packet, the actual Quality-of-Service values are returned to the intermediate nodes as acknowledgements. The long distance probing packets with satisfactory Quality-of-Service values are stored at the destination node. All the arrived short distance probing packets are also stored.

Step 2: All the paths with satisfactory Quality-of-Service values are sent back to the source node v_s along the reverse path in the Path Stack. All the intermediate nodes from v_d to v_s will memorize or update their experiential route information.

Step 3: At the source node v_s , among all the paths with satisfactory Quality-of-Service values, the best one is selected by the evaluation standard set out in Section 3.3. If $price$ is not greater than Pay (i.e., the user pays less than the upper limit), the routing succeeds; otherwise, the routing fails. If no path is returned within time interval TS_{ur} , the routing fails.

2.10 Quality-of-Service Routing Protocol Oriented to Cognitive Network

Step 1: Upon receiving the routing request $R(v_s, v_d, AP_{ur}, SL_{ur}, Pay)$, the source node v_s looks up the successful experiential routes in its local memory. If there are paths to v_d with Quality-of-Service at service level SL_{ur} of application type AP_{ur} , then go to Step 2; otherwise, go to Step 3.

Step 2: The source node v_s sends the probing packets to v_d along all the discovered paths. Once arriving at v_d , the probing packets return the acknowledgements which contain the actual Quality-of-Service information of the paths. If the probed paths can satisfy the Quality-of-Service requirements, then go to Step 4; otherwise, go to Step 5.

Step 3: The neighbour nodes look up the experiential routes in their memories. If there are paths to v_d with satisfactory Quality-of-Service values, add the current node as the source node to form new paths, then go to Step 2; otherwise, go to Step 5.

Step 4: Among all the paths with satisfactory Quality-of-Service values, the best one is selected by the evaluation standard set out in Section 3.3. If $price$ is not greater than Pay (i.e., the user pays less than the upper limit), then go to Step 6; otherwise, go to Step 5.

Step 5: Search the routing path by the extended Bellman-Ford algorithm. If the routing succeeds, then go to Step 6; otherwise, the routing fails.

Step 6: Reserve the resources on the selected routing path based on the Quality-of-Service requirements and set up the routing entries.

2.11 Quality-of-Service Rerouting Protocol Oriented to Cognitive Network

In the routing protocol oriented to cognitive network, each router maintains the neighbours' state information instead of the global information. Therefore, the rerouting only involves the local routing tables instead of the routing tables in the whole network. The rerouting needs to generate the interpretation information accompanied by the interpretation behaviour. The rerouting exploits the experiential route information in the memory to detect if the next hop node is still available. Thus, the failure type can be determined, i.e., whether it is node failure or link failure. In the cognitive protocol, every node periodically interacts with its neigh-

bours. When one node identifies an unreachable neighbour, it searches its routing table to check if there is a routing entry where the unreachable neighbour is set as the next hop. If so, the rerouting will be triggered. By exploiting the cognitive behaviour and the useful information accumulated during the routing process, we have designed a simple but effective rerouting scheme as described below.

Step 1: When node v_{er} identifies a routing entry where the next hop is unreachable, it starts to construct the interpretation information. The upper stream node of the unreachable neighbour is denoted as v_{er} and $Type_{ep} = NULL$. Then node v_{er} looks up its memory and get the experiential route which also exists in the routing table.

Step 2: If $v_{er} = v_s$, then go to Step 4, otherwise, go to Step 3.

Step 3: Send the probing packet to the source node along the upstream node on the experiential route, and obtain the application type AP_{er} and service level SL_{er} of the original routing request.

Step 4: Set the original route in the interpretation information as a successful experiential route denoted as $OPath_{ep}$.

Step 5: Look up the topology information in the memory of v_{er} . If there exists another route to the next hop node, detect if the node is reachable. If it is reachable, set the $Type_{ep}$ as link failure.

Step 6: Node v_s uses the routing path search procedure to find a new route which satisfies the requirements of service level SL_{er} in application type AP_{er} . If the probing is successful, the rerouting succeeds and the new route $RPath_{ep}$ is stored at the destination node, then go to Step 7, otherwise, the rerouting fails.

Step 7: If $Type \neq NULL$, then go to Step 8, otherwise, if $RPath_{ep}$ contains the next hop of v_{er} in the original route, then go to Step 9.

Step 8: Send the probing packet reversely along $OPath_{ep}$ to determine if the next hop of v_{er} is reachable, then set $Type_{ep}$.

Step 9: Execute the interpretation behaviour at the destination node, modify the routing entries, and reserve resources along the new route.

3 Experimental Study

3.1 Experimental Environment

To implement the cognitive Quality-of-Service routing and rerouting protocol, we need to add a knowledge plane to each node and the knowledge plane is responsible for the implementation of cognitive behaviours. We use NS2 for performance evaluation. In NS2, to implement the functions of knowledge plane, we need to add

a new protocol called Agent and bind Agent with each node. The Agent protocol inherits the Agent class in NS2. We implement the aforementioned cognitive behaviours based on the extended Agent protocol. According to the generic service Quality-of-Service specified in the ITU-T G.1010 standard, we use the following generic service classes in the simulation experiments. They are listed in Table 6. In the experiments, we further classify each application type into four levels and each level represents one service level of the user Quality-of-Service requirements. The user request includes both the application type and the service level. By the mapping, we can get the user's detailed Quality-of-Service parameter requirements.

Table 6: Generic service class.

Application	Bandwidth	Delay	Delay Jitter	Error Rate	Duration Time
App1	> 3Mbps	< 120ms	< 10ms	0	10-90min
App2	384Kbps-1.44Mbps	< 250ms	< 10ms	< 0.01	1-30min
App3	1Mbps-6Mbps	< 250ms	< 250ms	< 0.01	5-180min
App4	10Kbps-10Mbps	250ms-1s	—	0	0.5-20min

The network topologies play fundamental roles in the simulation experiments. To evaluate the protocol performance, we must select appropriate topology examples which can reflect the practical networks. In the following experiments, we have used random topology generated by the Waxman random graph model [Waxman(1998)]. The random topology consists of 50 nodes and 80 links. We name the proposed Cognitive Rerouting protocol as CoRp and evaluate its performance in various aspects. In [Khadiivi, Samavi and Todd(2008)], a multi-constraint Quality-of-Service routing protocol is proposed based on a new single mixed metric. It is a novel heuristic Quality-of-Service routing protocol named as SMMQR. It considers multiple Quality-of-Service constraints over the links and integrates them into a single mixed metric. Then the shortest path routing algorithm is utilized to find the Quality-of-Service routing path. Simulation and analytical results show that SMMQR can well solve the multi-constraint Quality-of-Service routing problems with high routing success ratio. In the following experiments, we compare CoRp with SMMQR and DV (Distance Vector) protocol implemented in NS2.

3.2 Experimental Results on Cognitive Routing

(1) *Control Overhead* Based on the selected topology, we run CoRp, SMMQR, and the standard DV protocol in NS2 to compare their control overhead. The source and destination nodes are randomly selected and routing requests arrive continuously. For CoRp, four different scenarios are designed. First, all the nodes turn off all the

cognitive behaviours. Second, the source node turns off all its cognitive behaviours and all the other nodes turn on their memorization behaviour. Third, the source node turns on the search help function and all the other nodes turn on their cognitive behaviours. Fourth, all the nodes turn on the search help function and memorization behaviour. These four scenarios are denoted as CoRp-1, CoRp-2, CoRp-3, and CoRp-4. The comparison results are shown in Fig 2.

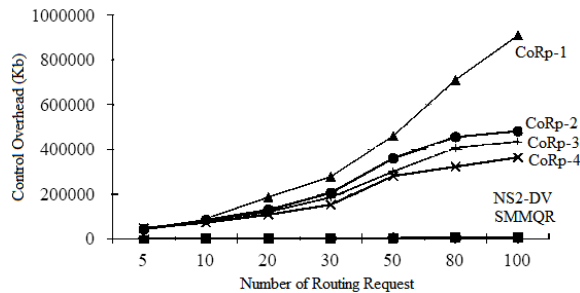


Figure 2: Comparison of control overhead.

Since DV does not consider Quality-of-Service and only exchanges connectivity information with neighbour nodes, its control overhead has nothing to do with the number of routing requests. So its control overhead is fixed. SMMQR combines multiple routing metrics into a single one. Its control overhead is increased with the increase in the number of routing requests. For CoRp, since it has used much more control messages than both DV and SMMQR, its control overhead is also higher than them.

(2) Routing Success Ratio

A certain number of traffic flows is generated randomly and classify them into five levels.

The number of data flows = $N \times L \times T \times i$

Where N is the number of nodes, L is the number of links, T is a benchmark and ($i=1,2,3,4,5$). In the protocol, the source node and the destination node is randomly determine and randomly set the application type of the corresponding routing requests. 100 continuous routing request under each load level is generated and then count the times that the routing requests return successful results. The routing success ratio is then calculated and the results are shown in Fig.3

The results show that for the protocols the routing success ratio decreases when the traffic flow level goes up. CoRp can guarantee the successful routing when feasible routes exist. The reason is that CoRp has considered all the Quality-of-Service con-

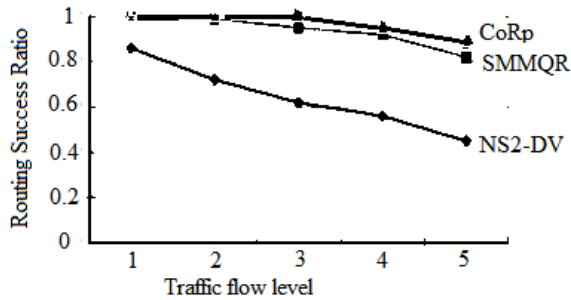


Figure 3: Comparison of the routing success ratio over random topology.

straints and utilized the cognitive behaviours. DV cannot adjust the routing dynamically when there are changes to the link Quality-of-Service parameters. Therefore, with the traffic flow level going up, the routing success ratio decreases significantly. For most of the time, SMMQR can satisfy the Quality-of-Service requirements. However, due to the combination of several Quality-of-Service constraints into a single metric, SMMQR cannot guarantee that all the Quality-of-Service constraints can be satisfied.

(3) User's Quality-of-Service Satisfaction Degree

Fig. 4 show the comparison results of the user's Quality-of-Service satisfaction degree, which exhibits similar trend as the routing success ratio. Under low traffic flow levels, the three protocols have competitive performance. With the traffic flow level going up, the user's Quality-of-Service satisfaction degree of DV decreases as the routing success ratio. DV's performance is much worse than the other two protocols. The user's Quality-of-Service satisfaction degree of SMMQR is slightly worse than that of CoRp also due to the combined single metric.

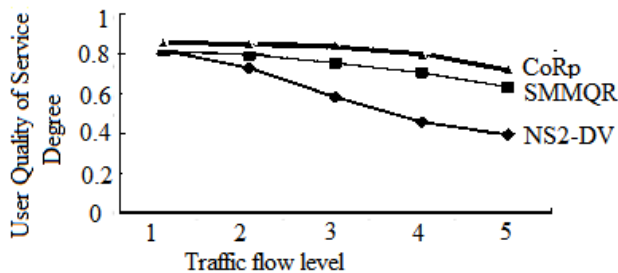


Figure 4: Comparison of the user's Quality-of-Service satisfaction degree

(4) Ratio of Successful Usage of Experiential Routes

The learning behavior updates the Quality-of-Service parameter estimation intervals according to the feedback from the usage of the experiential routes. The estimation interval of an experiential route is used to judge if it is suitable for use. To evaluate the effects of the learning behaviour, we calculate the ratio of successful usage of the experiential routes. In the experiments, we first generate 100 routing requests randomly over the random topology in order to memorize a number of experiential routes which are prepared for the learning behaviour. Then under each traffic flow level, we randomly generate 100 routing requests again to count the times that the experiential routes have been used. The results are shown in Fig. 5.

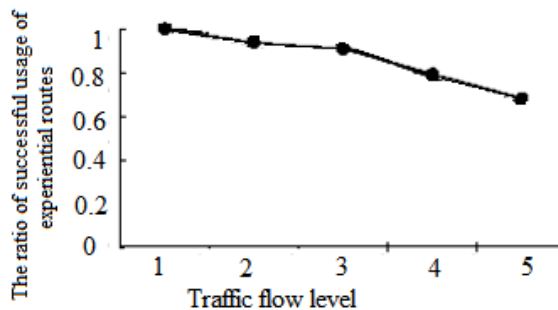


Figure 5: Comparison of the ratio of successful usage of experiential routes

From the results, with the traffic flow level going up, the ratio goes down. The reason is that at high traffic flow levels, some experiential routes cannot satisfy the Quality-of-Service requirements of the applications. Therefore, the ratio of successful usage of experiential routes decreases. When an experiential route is identified to be invalid, it will not be deleted immediately. Instead, it will be deleted after a few failure attempts. Thus, the failure times are not reduced. In the topology with good route diversity, new experiential routes can be easily found and memorized and successfully applied. This will help to alleviate the decreasing trend of the successful ratio.

(5) Usage Proportion of Experiential Routes

- i) Generate a certain amount of routing requests to accumulate the experiential routes under traffic flow level 3.
- ii) Generate randomly 100 continuous routing requests.
- iii) Count the times that the experiential routes are used in the successful routing of CoRp.

iv) Calculate the usage proportion of the experiential routes in the final routes found by SMMQR and DV.

The results are shown in Fig. 6. Since DV has the lowest routing success ratio among the three protocols, its usage proportion of experiential routes is also lower than the proportions in both CoRp and SMMQR. For SMMQR, the routes in it are slightly different from the ones in CoRp.

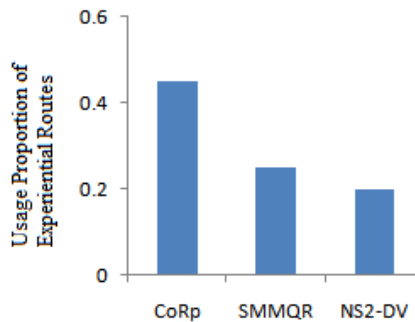


Figure 6: Comparison of proportion of usage of experiential routes

(6) Routing Discovery Time

As CoRp is a distributed routing protocol, the routing discovery time heavily depends on the link delay in the network. The actual running time of the protocol cannot directly reflect the performance of the protocol. Therefore comparing the routing discovery time by mapping it to the relative relation between CoRp with SMMQR and NS-DV. The comparison results are shown in Fig. 7.

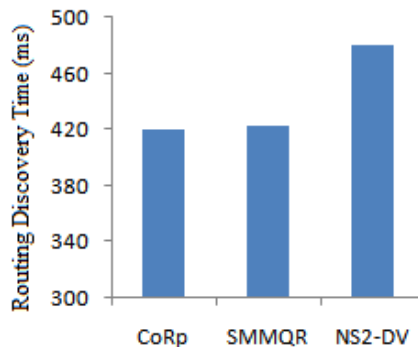


Figure 7: Comparison of the routing discovery time

4 Conclusion

With the ever-increasing spatial and temporal complexity in the topology, the network administration and maintenance is facing huge challenges. Especially when the users propose different Quality-of-Service requirements, the current network and routing schemes cannot deal with these requests effectively. The incurred problems not only bring inconvenience to the users but also burden the network administrators. The cognitive network is developed to solve the above problems by reducing the human intervention in the network management. It is expected to help enhance the network in scalability, self-adaptation, self-organization, self-protection, mobility and diversity. In this paper, we have designed a Quality-of-Service rerouting protocol oriented to cognitive network. The design utilizes the cognitive behaviours presented in the cognitive science. The cognitive behaviours in our model mainly consist of the sensation, sense of spatiality, memorization, learning, reasoning, and emotion. We map the behaviours to specific functions at each node for supporting the Quality-of-Service routing and the network fault recovery. We compare its performance with two other popular protocols. The experimental results show that with the help of cognitive behaviours, Quality-of-Service rerouting protocol has greatly improved the network performance in Quality-of-Service provision and fault recovery.

References

- Pragyansmita, P.; Raghavan, S.** (2002): Survey of Quality-of-Service routing. *Proc. 15th International Conference on Computer Communication*, Mumbai, India, pp.50-75.
- Wang, Y. X.; Wang, Y.** (2006): Cognitive informatics models of the brain. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.36, no.2, pp.203-207.
- Adrian, F.; Gerald, A.; Bruce, D. et al.** (2008): *Network quality of service: know it all*. Burlington, MA: Elsevier Inc, 2008.
- Blake, S.; Black, D.; Carlson, M. et al.** (1998): An architecture for differentiated services. IETF draft, draft-ietf-diffserv-arch-01.txt, Aug. 1998.
- ITU-T G.1010**, End-user multimedia Quality-of-Service categories.
- Waxman, B.** (1998): Routing of multipoint connection. *IEEE Journal on Selected Areas in Communications*, vol.6, no.9, pp.1617-1622.
- Khadivi, P.; Samavi, S.; Todd, T.** (2008): Multi-constraint Quality-of-Service routing using a new single mixed metrics. *Journal of Network and Computer Applications*, vol.31, no.4, pp.656-676.