2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing

A Cognitive QoS Method Based on Parameter Sensitivity

Zhikui Chen, Feng Han, Yang Liu
School of Software
Dalian university of Technology
Dalian 116620, China
zkchen@dlut.edu.cn, hanfengbest@gmail.com, lastyang1984@hotmail.com

***Abstract*—Different applications in the network have different sensitivity for the certain QoS parameters. The existing mechanisms cannot modify the packet loss policy exactly according to the needs of QoS parameters. Therefore, the network will not maximize its overall efficiency. This paper proposes a novel cognitive approach for QoS. It classifies the applications by various combinations of the QoS parameters. And it uses a cognitive layer to modify the loss probability of the packet. Simulation results prove that this approach can reduce the account of useless packets sent when the congestion occurs in the network. The overall efficiency of the useful data transmission can be improved as well. The method can be used by Internet of Things to improve its message processing and sensor data fusion to reduce energy consumption.**

Keywords- Parameter sensitivity; QoS; Cognitive Network; Internet of Things

I. INTRODUCTION

With the development of High-speed network and multimedia technology, it is difficult to meet the QoS requirements of various applications for the current network. Besides common data communications, the network also provides video conferencing, VoIP and other real-time multimedia applications. Variety complex applications lead to most of the bandwidth occupied by multimedia communications. It is estimated that delay-sensitive multimedia traffic will hold more than 80% of the bandwidth so that it is difficult to use reliable transmission to protect all critical applications.

The real-time multimedia applications are sensitive to the network delay, jitter, packet loss rate and other parameters. Unexpected high rate FTP applications or HTTP applications will make a great impact on it. Therefore, how to adjust the QoS of the packet in the network by the changes of the network environment becomes a hot issue.

In order to ensure the normal operation of the network, IETF proposed IntServ model, DiffServ model [1], MPLS technology [2], Traffic Engineering [3] and RSVP [4] to manage and control applications' QoS. These technologies do not have cognitive function. Most of them identified the transmission priority of the packet before the packet send to the network. The status of the network has great impact on the general level of QoS.

Therefore, academic research has begun into how to integrate cognitive elements in next generation network to overcome the inherent defects in the current network. The concept of Cognitive Network was proposed and got many achievements. Some researchers proposed self-configuration features of the routing algorithm from the perspective of routing optimization [5]. Some researchers, from the perspective of cross-layer design, used special algorithm to dynamically select the parameters of each layer, such as transmission power of the physical layer, avoidance time of the Mac layer and variable transmission rate range of the network layer [6]. Some researchers dynamically divided the network resources by the characteristics of data flow [7]. These researches have achieved good results. However, all the researches were under the conditions that the link state can meet the QoS parameters required for applications. The network can provide QoS guarantee according to certain priority policy. These policies ignored the situation when the link state cannot afford the QoS parameter requirements of some applications.

In this paper, we propose a cognitive QoS method based on parameter sensitivity. It classifies the applications by various combinations of the QoS parameters and uses a cognitive layer to modify the loss probability of the packet. So this method not only ensures the fairness of the priority-based scheduling when the links works well, but also ensures the correctness of packet loss when the link state suffered a sudden deterioration. The general efficiency of the network can also be improved.

II. SENSITIVITY PARAMETERS FOR DIFFERENT APPLICATIONS

Quality of Service can be described as a range of services the network required when transmitting data flow. It can be quantified as the bandwidth, delay, delay jitter, loss rate, throughput and other performance indicators. Different application has different requirements on the certain parameter. For example, Telnet protocol is only sensitive to the parameter of delay. Some real-time multimedia transmission protocols such as VoIP are sensitive to bandwidth, delay, delay jitter, loss rate and throughput. Table 1 shows the requirement of some kind of applications.

As can be seen from Table 1, the parameter sensitivity for different applications has big gap. For a specific application, it will enable users to get satisfactory services only when all of the parameter requirements guaranteed. On the contrary, even if one of them cannot meet the needs of users, it may leads to grievances or unbearable. Take VoIP application for example, assuming that the other three

978-0-7695-4331-4/10 $26.00 © 2010 IEEE
DOI 10.1109/GreenCom-CPSCom.2010.85

parameters fully meet the requirements of the application, but the delay is more than 400ms. When the users talk to each other, they will clearly feel that they should wait a while after the other side finish talking or the call will be affected. Furthermore, if the delays of some packets increase to 1 second, when the packet arrived, it will have negative effect on the user calls. Even if the upper layer protocol can filter out these packets, they still increase the burden to the terminals. Therefore, these packets can be defined as "useless packet" and should be firstly dropped when congestion appeared.

TABLE I. PARAMETER REQUIREMENTS FOR APPS

ToS	Parameter requirements			
	Band-width (bps)	Delay (ms)	Jitter (ms)	Packet Loss (%)
FTP	>0.2M	-	-	-
Telnet	-	<800	-	-
VoIP	>21K	<400	<30	<1
HDTV	>20M	<250	<1	<1

The protocols like Interv/RSVP, they promise to meet the needs of all the QoS parameters before the packet sending into the network. And they only ensure the application to occupy adequate resources on condition that the link state is good enough. But when a sudden change in the hardware environment leads to bad link state, they cannot ensure all packets' arriving can meet the requirements. In this case, many "useless packets", as we defined before, will be transmitted in the network.

The current studies are limited to how to ensure the promised priority if the application does not occupy resource out of its range. In this condition, those "useless packets" will still be forwarded to other routers or terminals according to their priority. This may have little impact when there are adequate resources of network to ensure the quality of communication for all applications. But when the congestion occurs, there will be a improper situation that some useful but low priority packets get dropped while other useless but high priority packets be forwarded. In fact, the throughput of useful packets becomes smaller.

III. ALGORITHM FOR COGNITIVE QOS

The current cognitive networks often use the method of computing users' priorities to ensure the satisfaction of the terminal. Some studies suggest that the delay and throughput can be use as the main target to measure the satisfaction of users. Because of delay-sensitive multimedia stream will occupy more than 80% [8] of all throughputs in the cognitive network, some studies suggest that we can only use delay instead of throughput to measure the satisfaction of users. By reducing average delay of packets, we can improve the performance of the network. These methods are actually a kind of estimation of system state. They are not able to establish the corresponding relationship between users' QoS and the system objective. Figure 1 shows the packets' forwarding mode in the current network.

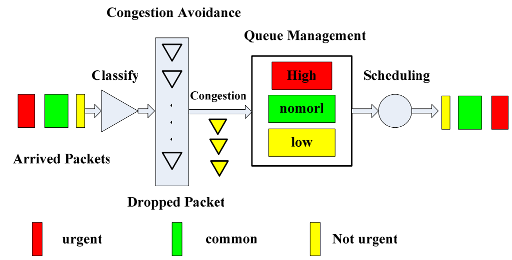


Figure 1. Forwarding mode in the current network

When the congestion occurs, the network will drop low priority packets to ensure high priority packets' forwarding.

Actually, different applications in the network have different sensitivities for the certain QoS parameters. According to this theory, this paper proposes a new method to dynamically adjust the applications' QoS. It determines the parameter space according to the packet's ToS (Type of Service). It compares real-time values of the parameters calculated by the cognitive layer with pre-set boundary values of applications and uses exponential weighting form to remark the packet's drop probability.

We add a cognitive layer to calculate a new drop probability of the arriving packet. The cognitive layer sets the total parameter space as P={P1,P2…,Pn}, the weights of n parameters can be set as w1,w2…,wn(w1 + w2 + … + wn = 1). Each cognitive node should set the threshold max[i] of every parameters the application interested before development. The algorithm is described as follow:

```
Algorithm Cognitive_user_pkt
Cognitive_user_packet(pkt)
type ← pkt.TOS
P_pkt ← init_Param_P_pkt(type)
n ← P_pkt .length
result ← 0
while n≥0
      do param_avg_value[i] ← get_avg_value(pkt.P[i])
         pthrow[i] ← param_avg_value[i]/max[i]
        if pthrow[i]≥1
            then result=1
                 break
         else result ← result + pthrow[i] * w_i
          n ← n-1
```

For each packet, the cognitive layer calculates a new drop probability by its parameter space P={P1,P2…,Pn} and the threshold max[i] as the formula:

$$P_{throw} = \begin{cases} 1 & if\ (\exists)\, pthrow[i] \geq 1 \\ \sum_{i=1}^{n} pthrow[i] * W_i & \end{cases}$$

See Figure 2 for the process of cognitive QoS method based on parameter sensitivity.

When the congestion does not occur, the route still forwards the packet by its priority. But when the network congestion occurs, the route will drop the packet which has relatively larger probability of the packet loss. In this way, the network can transmit more useful data to the clients.

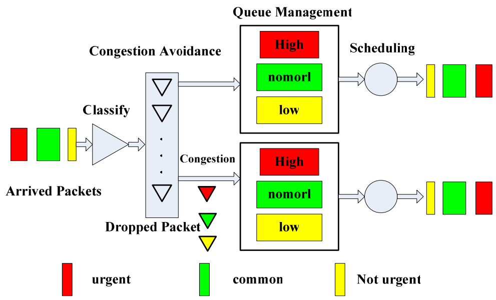


Figure 2. Forwarding mode joined our method

IV. SIMULATIONS AND ANALYSIS

A. *Environment and Process*

We run simulations on the Network Simulation Platform ns-2.33 and design a topology as Figure 3 to test the performance of the network after joining the cognitive QoS method based on parameter sensitivity.

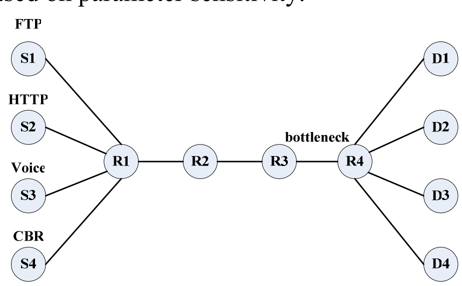


Figure 3. Simulation topology

As is shown in Figure 3, from S1 to S4 are four different application servers and form D1 to D4 are the corresponding receivers. R1 to R4 are route nodes. The simulation time is 50s. There is a bottleneck link from R3 to R4. Its bandwidth is 0.8Mbps and link delay is 20ms. While other links have 1Mbps bandwidth and 20ms link delay.

S1 and S2 are FTP and HTTP servers on TCP protocol. S3 is a Voice server with 64Kbps sending rate. S4 is a CBR flow with 512Kbps sending rate. The network is running in the congestion state for most of the time in order to facilitate analysis.

The experiment simulated in two scenes. One is the situation described above; the other is the link between S3 and R1 suddenly becomes deterioration. Its delay changes into 300ms. Then we analysis the simulation experiment from the following aspects:

(1) The first scene, when the state of the link between S3 and R1 suddenly becomes bad, some parameters of the arriving packets may not meet the minimum requirements of the applications. Make the comparison once again to observe whether the number of "useless packet" sent is reduced and the number of other useful packet send is increased. If it is true, we can conclude that the actual performance of the network has been improved. Section 4.2 discusses this experiment.

(2) The second scene, we compare the network performance after joining Cognitive QoS Method with that does not join the method. This comparison can observe whether there is a great negative impact for the network after joining the method. If not, we can conclude that this method is friendly to other protocols. Section 4.3 discusses this experiment.

B. *Network Performance*

For the four applications in this topology, the sort order of their priorities is: 1.Voice 2.FTP and HTTP 3.CBR. Therefore, most of the dropped packets should be the CBR's when congestion occurs. We define the situation of joining Cognitive QoS Method to the network as "cognitive_mode". The opposite one is "common_mode".

When the state of the link between S3 and R1 suddenly becomes bad, there will be many "useless packets". If the cognitive nodes drop these packets early rather than sending them to their destinations, the number of useful packets sent will increase. We can validate whether the overall performance of the network by comparing the two models. The main applications' send/lost packets are shown in Figure 4.

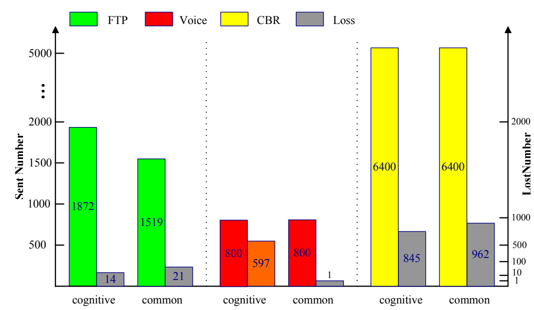


Figure 4. First scene's Packets sent/lost

There are many useless packets which belong to the Voice stream in the network because of the bad link. The "common_mode" still sent these useless packets firstly because that they have high priority. On the contrary, the "cognitive_mode" will drop the packets according there packet loss probability. So the useless packets will be dropped firstly though they may have high priorities. This leads to the CBR and FTP's number of packet loss reduced and the FTP's number of packet sent increased.

In terms of total throughput of the network, the "cognitive_mode" is a higher than the "common_mode". See Figure 5.

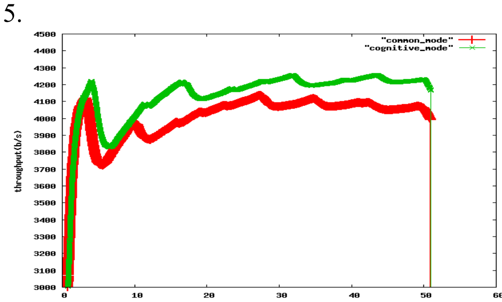


Figure 5. First scene's Throughput Comparison

Voice stream's delay increased because its transmission link' state got worse. Many packets' delay parameter became more than 400ms which is the threshold value of delay when the packet arrived at the cognitive route R3. So R3 dropped most of the useless packets, shown in Figure 8(c). Therefore, the CBR's packet loss rate gets small but the average delay does not change much, see Figure 8(d).

As mentioned above, our Cognitive QoS method can drop the useless packet when the congestion occurred. The other applications can send more useful packets and get small packet loss rate so the total throughput increased and the performance of network improved.

C. *Protocol-Friendly*

Figure 6 shows the number of packets sent/lost for three applications in the experiment. We can see that "cognitive_mode" has little negative impact for packet loss.

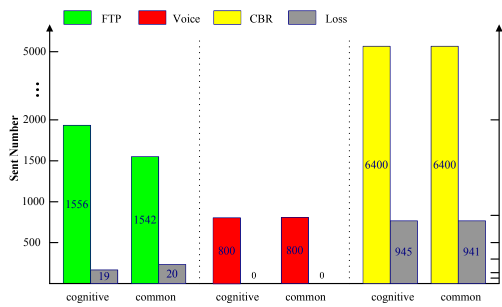


Figure 6. Second scene's Packets sent/lost

Another two aspects we should discuss are the influences of the "cognitive_mode" on the throughput of the network and the delay of the applications. Figure 7, 8(c) (d) shows the throughput comparison of the simulation.

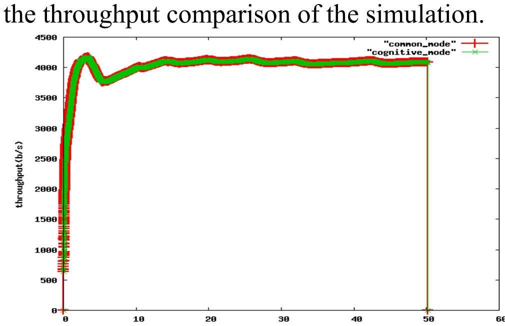
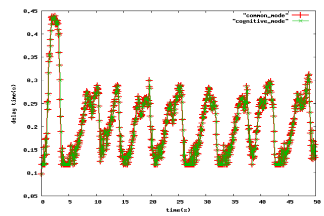
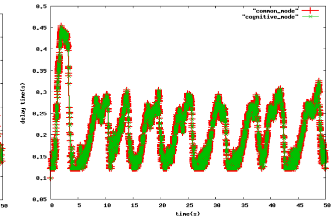


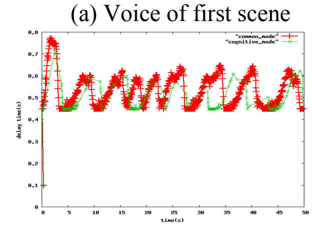
Figure 7. Second scene's Throughput Comparison

Because of the green line and the red line has the consistent trend. So we bold the red line to distinguish them. That is why the green line looks like surrounded by the red line. Figure 8(a) (b) shows the delay comparison of the simulation.

We can see that neither the throughput nor the delay of applications change a lot. The Cognitive QoS Method can work well with other protocols the applications use. Therefore, we can conclude that our method is protocol-friendly.


(a) Voice of first scene


(b) CBR of first scene


(c) Voice of second scene

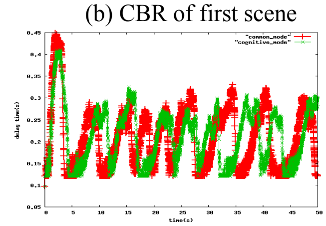

(d) CBR of second scene

Figure 8. Delay Comparison

V. CONCLUSIONS

This paper proposed a cognitive QoS method based on parameter sensitivity. We consider that different applications in the network have different sensitivity for the certain QoS parameters. So the applications can be classified by various combinations of the QoS parameters. Then we add a cognitive layer to modify the loss probability of the packet dynamically. This method can solve the problem that the packet whose important parameters does not meet the requirement of terminals (we call it "useless packet") will be sent early just because it has high priority. The simulations proved that our method is protocol-friendly and will improve the performance of the network when it is congested.

ACKNOWLEDGEMENTS

This work was supported by the Fundamental Research Funds for Central Universities: DUT10ZD110.

REFERENCES

[1] S. Shioda and K. Mase, Performance comparison between IntServ-based and Diffserv-based network, Proceedings of IEEE Global Telecommunications Conference 2005, vol.1, pp. 529-534, Dce. 2005.

[2] R. Aggarwal, OAM mechanisms in MPLS layer 2 transport networks, IEEE Communications Magazine, vol.42, no. 10, pp.124-130, Oct.2004

[3] M. J. Kim, Traffic Engineering for Next Generation Mobile Communication Systems, Proc. of IEEE Wireless Communications and Networking Conference 2000, Vol. 3, pp. 1453-1456, Sept. 2000.

[4] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, RSVP: A New Resource Reservation Protocol, IEEE Communications Magazine, vol. 40, no.5, pp.116-127, May 2002

[5] KATSUNO Y, AIHARA T, RES I. Autonomic network configuration for networkable digital appliances. IEEE Transactions on Consumer Electronics, 2005, 51(2):494-500.

[6] BAYNAST A D, MAHONEN P , PETROVA M. ARQ-based cross-layer optimization for wireless multicarrier transmission on cognitive radio networks . Computer Network, 2008, 52 (4): 778-794.

[7] TANSUPASIRI T, KANCHANASUT K, BARAKA C. Using active networks technology for dynamic QoS. Computer Networks, 2006, 50(11):1692-1709.

[8] LIN C, WANG Y Z, REN F Y. Research on QoS in next generation network. Chinese Journal of Computers, 2008, 31(9): 1525-1535.