2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS) | 978-1-6654-1538-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ACCTCS52002.2021.00012

A TCP friendly rate control algorithm based on GRU prediction model

Kuai Yu
Chongqing University of Posts and Telecommunications
School of Computing Science and Technology
Chongqing, China
2289642838@qq.com

Zhaohua Long
Chongqing University of Posts and Telecommunications
School of Computing Science and Technology
Chongqing, China

***Abstract*—In order to solve the congestion control problem encountered in the transmission of real-time data in the wireless network, combined with the GRU neural network, an optimized TCP-friendly rate control algorithm based on the GRU prediction model is proposed. Predict the sending rate of the sender in the next stage, so that TFRC can adjust the sending rate more timely and accurately. Experiments have proved that the improved algorithm can optimize the congestion caused by blind adjustment of the sending rate to a certain extent while ensuring TCP friendliness Packet loss problem.**

Keywords—Congestion control, friendliness, sending rate, GRU

I. Introduction

Data transmission on the Internet is mainly based on TCP and UDP. For network video transmission, TCP's retransmission mechanism will increase the delay at the receiving end, and it is not necessary; TCP's congestion avoidance mechanism with a rate halved can easily cause drastic fluctuations in the data transmission rate, causing delays and delays in the video image. Discontinuity. UDP does not have any congestion control mechanism. In a congested network environment, UDP streams will occupy a large amount of network bandwidth of TCP streams. It does not have TCP-friendly features. At the same time, the packet loss rate of UDP itself will increase rapidly. , And may bring the risk of system congestion collapse. Therefore, neither TCP nor UDP can meet the needs of network video transmission. In order to solve the above problems, Floyd et al. proposed the TFRC (TCPfriendly rate control) protocol. The TFRC protocol uses an end-to-end rate control mechanism based on the TCP throughput model. Under the same conditions, TFRC streams can share network bandwidth with TCP streams fairly; on the other hand, TFRC streams have stable throughput changes and low jitter, so they are very suitable for streaming media applications that require high transmission rate smoothness

II. TFRC rate control mechanism

A. Basic principles of TFRC protocol

The TFRC protocol adjusts the sending rate according to the TCP Reno throughput model proposed by Padhye et al. The model is

$$T = \frac{s}{R\sqrt{\frac{2p}{3}} + 3t_{rto}\sqrt{\frac{3p}{8}}p(1+32p^2)}$$

Among them: T is the data transmission rate, the unit is Bps; s is the data packet size, the unit is Byte; R is the round trip time RTT (round trip time), the unit is s; trto is the TCP retransmission timeout time, generally trto = 4 × R; p is the packet loss event rate.

a) The receiving end measures the packet loss event rate p, and feeds p back to the sending end together with the time stamp of the received data packet.

b) The sender uses the timestamp in the feedback packet to measure the loop time RTT.

c) The sender substitutes the packet loss event rate p and RTT into equation (1) to calculate the data transmission rate T.

d) The sender adjusts the sending rate of the data packet according to the calculated transmission rate T

B. Limitations of TFRC used in wireless environments

In the case of unstable network conditions such as wireless networks, TFRC cannot adapt well to network changes, mainly because the throughput calculation formula of TFRC only calculates throughput estimates based on changes in packet loss rate and delay. In wired networks, packet loss events are mainly caused by network congestion, so TFRC can operate effectively. However, in the wireless environment, most of the packet loss is caused by the unreliability of wireless channel transmission. Therefore, when TFRC is directly applied to the wireless network, random packet loss events caused by the wireless channel will be regarded as packet loss caused by congestion. , Thereby blindly lowering the rate. When the bit rate of wireless transmission is relatively high, it may cause serious degradation of the service quality of multimedia applications. Moreover, TFRC uses p (packet loss event rate) as a congestion signal. It is adjusted when the network has already been congested. This will inevitably cause a sudden change in the sending rate and cannot guarantee the smoothness of real-time data transmission.

978-1-6654-1538-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ACCTCS52002.2021.00012

III. GRU NEURAL NETWORK

A. Recurrent Neural Network RNN

Recurrent Neural Network (RNN) is a type of recursive neural network that takes sequence data as input, recurses in the evolution direction of the sequence, and all nodes (recurrent units) are connected in a chain network.

B. Gated Circulation Unit GRU

GRU is an improved version of the standard recurrent neural network, GRU is Gated Recurrent Unit

In order to overcome the inability of RNN to handle long-distance dependence well, LSTM was proposed, and GRU is a variant of LSTM. GRU maintains the effect of LSTM while making the structure simpler. The model of GRU is shown in Figure 1:

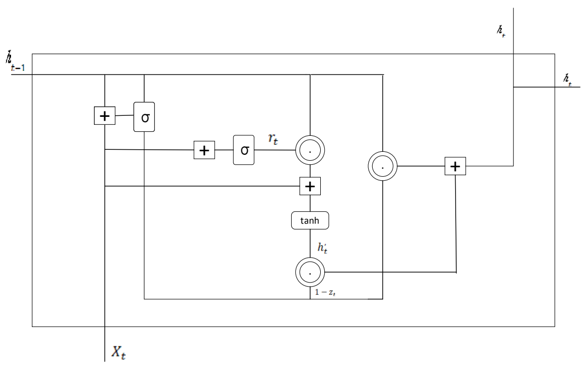

Fig. 1. GRU gated loop unit

GRU has two gates, namely a reset gate r_t (reset gate) and an update gate z_t (update gate). Intuitively, the reset gate determines how to combine the new input information with the previous memory, and the update gate defines the amount of the previous memory saved to the current time step.

Among them: update gate z_t:

$$z_t = \sigma(W^{(Z)}x_t + U^{(z)}h_{t-1})$$

Reset gate r_t:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

In order to solve the problem of vanishing gradient of standard RNN, GRU uses update gate and reset gate. Basically, these two gated vectors determine which information can ultimately be used as the output of the gated loop unit. The special feature of these two gating mechanisms is that they can save the information in the long-term sequence, and will not be cleared over time or removed because it is not relevant to the prediction

IV. THE TFRC-GRU ALGORITHM

A. Optimization of TCP friendly rate control algorithm based on GRU predictive model

In response to the above problems, this paper proposes a method to optimize the TCP-friendly rate control algorithm based on the GRU prediction model, which is based on the neural network gated recurrent unit (Gated Recurrent Unit) to predict the sending rate of streaming media video packets, and the TCP-friendly rate control algorithm TFRC (TCP-Friendly Rate Control) is combined to achieve the optimization effect of TFRC protocol. The flow chart of the model is shown in Figure 2:

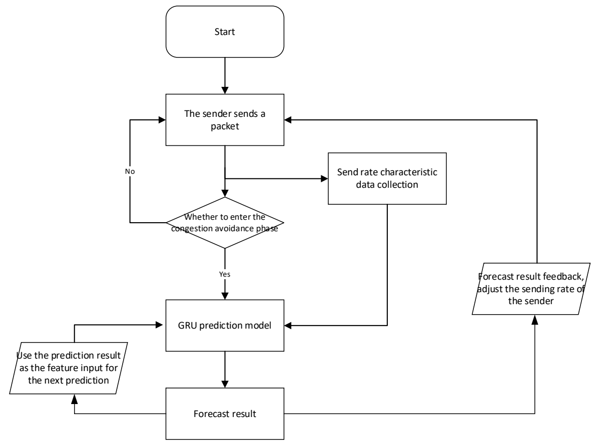

Fig. 2. Overall model flow chart

It mainly includes three parts: the acquisition of network special data, the prediction of the sending rate of video data packets, and the adjustment of the sending rate of data packets based on the predicted sending rate by TFRC; among them:

To obtain network special data, use packet capture software such as WireShark to collect the sending rate of video data packets per second;

The prediction of the sending rate of video data packets is to use the neural network gated recurrent unit GRU (Gated Recurrent Unit) to predict the sending rate of video packets. The historical sending rate and some influence parameters are used as the input of the GRU prediction model, and the future sending rate is used as The output of the GRU prediction model is used as the basis for the TFRC protocol to adjust the sending rate.

The GRU prediction model has three layers, using Gated Recurrent Unit (Gated Recurrent Unit), using a gating mechanism to control input, memory and other information to make predictions at the current time step, and use the prediction information as the input information for the next prediction. Go ahead and continue to predict.

The historical sending rate is used as the input information of the prediction model, and the input information styles are adjusted into samples, step sizes, and features. Two layers of GRU plus a hidden layer are used. The learning rate is set to 0.01, the number of training times is 1000, and the adam optimizer is used.

TFRC adjusts the data packet sending rate based on the predicted sending rate based on the TFRC protocol and adjusts the sending rate of video packets according to the prediction model to achieve better TCP friendliness and better congestion control effects.

The GRU predictive model optimizes the TCP-friendly rate control algorithm steps as follows:

The data sending process of the TFRC protocol is divided into two stages, slow start and congestion avoidance. After the connection is established, the slow start phase is initially entered. The sender first sends data packets at a very low initial speed, and then doubles the sending rate every RTT period

a) if the receiver detects a message loss event, the sender enters the congestion avoidance phase after receiving the feedback packet;

b) if the feedback timer expires, that is, the sender has not received the feedback report after the timer expires Text, the sending rate is directly halved, entering the congestion avoidance stage

At the same time, use professional tools to collect the sending rate in this stage as the input data of the prediction model, and predict the future sending rate.

When entering the congestion avoidance phase, the GRU prediction model predicts the sending rate and feeds back the prediction result to the sending end.

The sending end uses the predicted sending rate as the upper limit of the actual sending rate. For example, the current sending rate T < the predicted sending rate T1, the sending end increases the sending rate every RTT time period until the current sending rate T equals the predicted sending rate T1, the purpose is to make the sending rate increase steadily; if the current sending rate T > the predicted sending rate T1, the sending rate is directly adjusted to the predicted sending rate, the purpose is to reduce the sending rate in time and avoid network congestion

B. Experiment Analysis

The experiment uses the WireShark packet capture tool to collect more than 8 million pieces of data in time, classified by time, a total of about 30,000 groups, and standardize and normalize the data, using 80% of the data as the training set and 20% as the verification set , Use the keras platform to build the training model, the learning rate is set to 0.01, the number of training times is 1000, and the adam optimizer is used. The prediction results are shown in Figure 3:

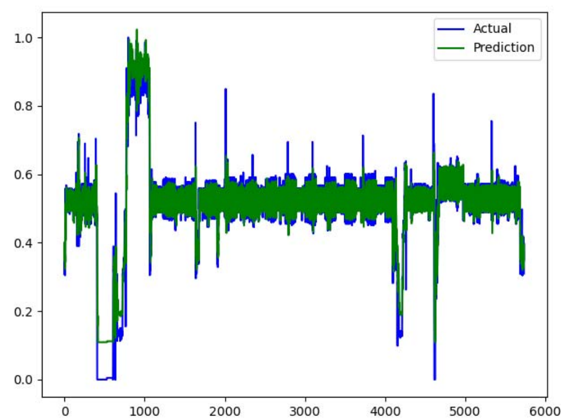

Fig. 3. GRU model's prediction of sending rate

In order to verify the improvement effect, the NS-2.35 network simulation software is used to simulate and compare the TFRC and TFRC-GRU algorithms. The simulation experiment uses the same network topology. As shown in Figure 4

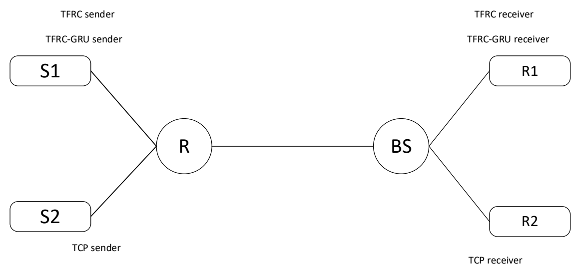

Fig. 4. Simulation network topology

In Figure 4, the link between router R and base station BS is a bottleneck link, with a bandwidth of 3Mb/s, a delay of 10ms, and a queue length of 200. The ARED queue algorithm is used. The 802.11b wireless link connection between the base station BS and R1 and R2 has a bandwidth of 11Mb/s. Router R and S1 and S2 are all wiredly connected, with a bandwidth of 100Mb/s and a delay of 5ms. The packet size of TFRC and TFRC-GRU streams is 1000B, and the packet size of the simulated burst data TCP stream dynamically changes to 400B, 800B, 1200B, and the simulation time of each stage is 20s. The simulation experiment was carried out twice, from S2 to R2 to simulate TCP connection, and from S1 to R1 to simulate UDP connection. According to the above deployment situation, the experiment was completed in two times: the first time was to simulate the working condition of TFRC under TCP flow; the second experiment was specifically aimed at the results of the first experiment, TFRC-GRU made in the same experimental environment Algorithm comparison test. The experimental results are as follows:

TABLE I. EXPERIMENTAL RESULT

	TFRC	**TFRC-GRU**
Time of packet loss	1.34	0.82
Average packet loss rate	1.89	1.45

The comparison of packet loss rate is shown in Figure 5:

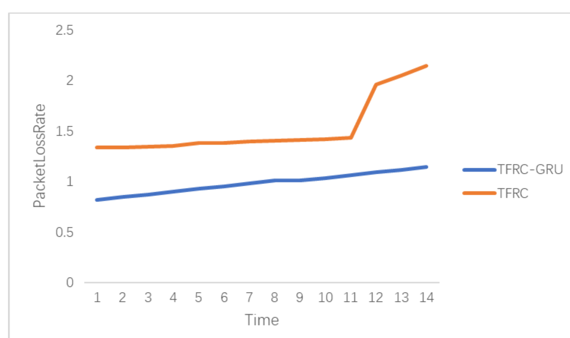

Fig. 5. Packet loss rate comparison

The simulation results show that tfrc-gru model can optimize the congestion and packet loss caused by blind adjustment of transmission rate in TFRC on the premise of ensuring TCP friendliness.

V. CONCLUSION

In this paper, the response of TFRC protocol to variable bandwidth is slow When the network environment is unstable, the response of TFRC scheme to the actual environment is delayed and the adjustment is not timely. Combined with the Gru neural network and the feature extraction of the transmission rate data, the characteristic data is used as the prediction model input, and the prediction result of the Gru prediction model is used as the upper limit of the transmission rate of the TFRC protocol The model optimization method of TCP friendly rate control algorithm can make use of the current sending rate of the sender and the network situation to predict the next sending rate in time, so that TFRC can adjust the transmission rate more timely and accurately, more TCP friendly and achieve better congestion control effect.

REFERENCES

[1] Li Wen Chao, Fang Dingyi, Feng Jian, et al. Research on P2P streaming media transmission protocol and QoS optimization [J]. Minicomputer system, 2009,03:472-476J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Yang Chuandong, Yu Zhenwei, Wang Xinggang, et al. Overview of end-to-end streaming media transmission control technology [J]. Computer engineering and applications, 2005,08:26-30K. Elissa, “Title of paper if known,” unpublished.

[3] Wu Yuanbao, Liu Zhensheng, Zhang Wenliang. Learning based extended AIMD congestion control mechanism [J]. Computer Engineering, 2008,09:232-234

[4] R Vinayakumar,K. P. Soman,Prabaharan Poornachandran.Applying deep learning approaches for network traffic prediction[C].Udupi, India:IEEE,13-16 Sept. 2017.

[5] Braden R, Zhang L, Berson S, et al. Resource Reservation Protocol(RSVP)-Version 1 Function Specification[S]. IETF RFC 2205,September 1997.

[6] [6] EMMANUEL J, MARTIN F, GHANBARI M. Fuzzylogic congestion control of transcoded video streaming without packet loss feedback[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2008, 18(3):387-393..