

Article

Research of Wireless Congestion Control Algorithm Based on EKF

Hui Wang *, Junyong Tang and Bo Hong

School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China; haoyujinguan@126.com (J.T.); hongbo123@xatu.edu.cn (B.H.)

* Correspondence: wanghuihy@xatu.edu.cn

Received: 26 February 2020; Accepted: 2 April 2020; Published: 17 April 2020



Abstract: The random variation of bandwidth in wireless networks causes some significant challenges to the congestion control protocols based on bandwidth estimation. In this paper, a wireless congestion control scheme based on extended Kalman filtering and bandwidth (CSEKB) is proposed. The CSEKB can effectively perceive the bandwidth oscillation of wireless networks and distinguish the type of packet loss by establishing a noise perception factor. According to the congestion factor, the congestion control parameters are adjusted to correspondingly improve the performance of the wireless network. Moreover, the variation trend of the size of the congestion window presents a law of similar normal distribution curve, which has a certain degree of local symmetry. The CSEKB was implemented in Network simulator 3 (NS3) and compared with TCP Westwood (TCPW), CUBIC, and extended Kalman filtering-based bandwidth estimation (EBE). Through extensive simulation studies, the proposed CSEKB demonstrated the significant performance in wireless networks. First, the CSEKB can achieve congestion control based on the accurate prediction of available bandwidth, and improve average throughput and link utilization. In addition, the CSEKB has good fairness and friendliness compared with several other well-known congestion control methods.

Keywords: bandwidth; wireless network; congestion control protocol; extended Kalman filtering; network simulator 3; TCPW; CUBIC; EBE; throughput

1. Introduction

Due to the particularity of the wireless network in terms of topology, data retransmission, and various services, the traditional congestion control algorithm such as transmission control protocol (TCP) New Reno [1] cannot fully guarantee the quality of service of the wireless network. Therefore, a lot of exploration and research have been done based on traditional congestion control algorithms. A variety of congestion control algorithms for wireless networks or high bandwidth time-delay networks have been proposed [2–8]. CUBIC is a high-speed improved version of transmission control protocol (TCP), and the size of the window in the CUBIC protocol is a cubic function of the time [9]. Moreover, it was explained in [10] that CUBIC modified the linear window growth function into a cubic function to better adapt to the fast and long distance networks. The main technical issue of these protocols is how to regulate the size of the congestion window more reasonably. Mascolo et al. proposed TCP Westwood (TCPW) [11] based on bandwidth for wireless networks. However, these protocols cannot take measures to control congestion in conjunction with the actual characteristics of wireless networks. In light of the large delay of wireless networks, asymmetric bandwidth, high bit error rate, and short-term stream, TCPW has the following main problems:

- (1) First, due to the influence of link state, medium competition, signal strength, and various interference factors, the available bandwidth of the wireless network changes significantly in a

dynamical and random manner. These make it impossible for TCPW to determine the actual amount of data to be sent, according to the change of bandwidth in real time. Hence, it leads to a significant decrease in the transmission performance of wireless network.

- (2) In addition, the weight of the filter used by the original bandwidth estimation algorithm is fixed in TCPW [12] and an improved algorithm (TCPW-M) [13], so the bandwidth cannot be estimated stably. By introducing the filtering technology into the estimation of the bandwidth, the impact of the frequent jitter of the rapid change of the wireless network bandwidth is effectively reduced. Thereby, a relatively smooth bandwidth estimation value that meets the actual situation is obtained. Thereafter, in order to ensure the smoothness of the estimated bandwidth, the filtering technique in the TCPW bandwidth estimated (TCPW BE) algorithm utilizes randomly varying weights. However, the estimation value of the method is high and the accuracy is insufficient. Then, the TCPW BE occupies excessive bandwidth.
- (3) Finally, there are many reasons for packet loss, and wireless network congestion is only one possibility. In addition, signal attenuation or external interference can cause noise to be lost. When the link environment of the wireless network is poor and noise packet loss occurs frequently, TCPW does not take different operations according to the packet loss type. As long as the packet is lost, TCPW is called to reduce the slow start threshold and the size of the congestion window. As a result, the throughput of the system is severely reduced and the performance of the wireless network is affected to a large extent.

Therefore, in light of the remarkable characteristic where the bandwidth in a wireless network changes randomly and the shortcomings of the TCPW protocol, the current available bandwidth of the wireless network should be estimated reasonably, accurately, and in real-time. Then, the size of the sending window should be adjusted dynamically. After that, according to different causes of packet loss, congestion loss or noise packet loss will decide what measures should be taken in the next step.

In this paper, a wireless congestion control scheme based on extended Kalman filtering and bandwidth estimation (CSEKB) is proposed. The CSEKB consists of two key components. (i) The method predicts network bandwidth based on extended Kalman filtering (EKF) in real time. Moreover, the model of bandwidth prediction based on EKF shows a law of similar normal distribution. (ii) In the CSEKB, we designed the message backlog perception factor F_{np} and distinguished whether congestion had occurred according to the type of packet loss, making the threshold for entering the slow start and the size of the congestion window more refined. The CSEKB is realized by improving the congestion signal in the existing wireless congestion control scheme TCP. The CSEKB aims to predict the network bandwidth in real time and improve congestion control of wireless networks. Finally, the service quality of the wireless network is improved and the relative stability of the wireless network system is ensured. The results demonstrate that the CSEKB can effectively predict network bandwidth in real time and achieve congestion control in wireless networks, and has better convergence, accuracy, and higher utilization.

The remainder of this paper is structured as follows. First, related works are discussed in more detail in Section 2. In Section 3, we describe the model of bandwidth prediction of CSEKB step by step. Next, the details of congestion control in CSEKB are presented in Section 4. In Section 5, we provide the simulation experiments and results analysis by comparing it with the three most commonly used wireless protocols including TCPW, CUBIC, and EBE. Finally, the paper is summarized in Section 6.

2. Research Background and Related Works

The accuracy of bandwidth prediction directly determines the quality of service in wireless networks. When the available bandwidth estimation of the wireless network is too high, more data will be injected into the network, which will reduce the quality of the network service and may cause congestion. In contrast, when the available bandwidth estimate is low, the utilization of the channel will be reduced, and the throughput of the system will be reduced because the channel capacity of the wireless network is not effectively utilized. Obviously, if congestion control is implemented on the

basis of inaccurate available bandwidth estimation, it will inevitably lead to congestion oscillation or a waste of resources.

To increase the performance in a wireless network, researchers have presented many strategies of congestion control protocol. These strategies can be categorized according to the technique they employ. In [14], the bandwidth was estimated in real time based on the amount of data confirmed in the round trip time (RTT), and filtering and smoothing were used to obtain a more accurate bandwidth value. The bandwidth utilization was used to first analyze the reason for packet loss and the degree of congestion, and then adaptively adjusted the congestion parameters dynamically. In [15], the available bandwidth of a node was monitored in real time, and the congestion level indicator of a node was constructed. The slow start threshold and the size of the congestion window were set on the basis of the available bandwidth, and congestion control was achieved by distinguishing different types of packet loss. A new mechanism was proposed to set *cwnd* and *ssthresh* during the congestion avoidance phase in [16]. The link bandwidth was detected by the number of bytes confirmed by two consecutive responses to reduce the slow startup time. It was presented to improve the bandwidth estimation based on the wireless link quality and the one-way delay change trend in [17]. A more reasonable congestion control would be realized on the basis of distinguishing packet loss. The random process characteristics of network traffic and statistical methods were used to estimate the available bandwidth in [18]. In [19], the impact of collision probability and frame synchronization on estimating the available bandwidth of the network was first considered. After that, the degree of dependence of the sender and receiver on the idle time was further evaluated, so that the available bandwidth estimation could be realized.

Bandwidth aggregation TCP (BATCP) was proposed in [20], which is an extension of TCP. BATCP schedules the segments based on the scheduling algorithm. In addition, multiple TCP connections to accept multiple IP addresses from multi-homed nodes were used in BATCP. Finally, in order to improve the performance on multi-homed nodes, the concurrent use of network interfaces was allowed in BATCP. A novel buffer occupancy-based transmission protocol (BOTCP) was presented in [21]. BOTCP presents a congestion signal based on buffer occupancy. Then, by using this congestion signal, a congestion control scheme was proposed to reduce the completion time of short flows. BOTCP gives an accurate measure of congestion based on this occupancy-based congestion signal. In addition, in order to reduce the completion time of short flows, BOTCP treats short flows and long flows differently.

However, the bandwidth prediction algorithms in these schemes are very simple, and the results are not ideal, and they cannot adapt well to the trend of dynamic random changes in wireless network bandwidth. Without proper compensation of bandwidth fluctuations, the system performance will be seriously deteriorated due to the influence of wireless network oscillation behavior.

Therefore, in order to solve the above problems, Kalman filtering (KF) technology has been considered to predict the available bandwidth of the network. KF can effectively deal with the linear filtering problem of discrete data; that is, it can estimate and correct the current state of the system and also predict the state of the system at a certain time in the future. A delay estimation based on Kalman filtering was proposed in [22], and congestion control was performed accordingly to improve the system throughput. Different bandwidth estimation methods based on Kalman filtering were proposed and applied to different scenarios in [23–35], however, these algorithms were all based on linear systems, while most of the actual networks were nonlinear systems.

Extended Kalman filtering (EKF) can be generally used to solve the state estimation problem in nonlinear systems. EKF can linearize the nonlinear dynamics and assume that the process and noise obey Gaussian distribution. In order to achieve the optimal path selection of the network, the bandwidth and round-trip delay of each path were estimated by using EKF in [36–38]. In [39], Cavusoglu B et al. proposed the use of EKF and unknown input streams in a non-linear system to adaptively achieve prediction of available bandwidth [40], and proposed an EKF-based bandwidth estimation (EBE) scheme. In order to predict the available bandwidth, the status of each stream and the size of the persistent queue of the sender were monitored. The congestion control performance of a

wireless network is effectively improved by using the EBE based on extended Kalman to predict the available bandwidth. However, in the process of bandwidth estimation, EBE has overemphasized the accuracy of bandwidth estimation values, resulting in high algorithm complexity, large calculation volume, slow convergence, and insufficient real-time performance. Therefore, it is hoped to propose an effective prediction method of available bandwidth that can not only ensure the accuracy of bandwidth prediction, but also quickly converge, and meet the actual demands of wireless network congestion control.

According to the above analysis, CSEKB is proposed in combination with the actual characteristics of the wireless network in the paper. CSEKB builds a bandwidth prediction model and estimates the bandwidth using EKF. After that, the current status of the wireless network is determined according to the predicted bandwidth, then different types of packet loss are used for congestion control to optimize the performance of the wireless network.

3. Model of Bandwidth Prediction Based on Extended Kalman Filtering (EKF)

The available bandwidth of a wireless network changes dynamically and randomly because of link status, media competition, signal strength, and other various uncertain factors. However, as a key parameter in congestion control and Quality of Service (QoS) guarantee, whether the bandwidth estimation is accurate or not is of great significance to the design of the congestion control algorithm. Therefore, in order to predict the available bandwidth of the network accurately and quickly, three main models of bandwidth prediction were first analyzed in this paper.

Then, the EKF was researched and applied to the detection interval model Gap Probe Model (GPM). This method provided us with an idea through EKF, and a prediction model of available bandwidth in wireless networks based on EKF was designed to calculate the available bandwidth and perform congestion control accordingly.

Finally, the purpose of improving the efficiency of wireless network congestion control was achieved.

3.1. Analysis of Available Bandwidth

In a wireless network, a certain link cannot always be idle. Therefore, the capacity of the link being used at a certain time can be defined as cross traffic, and the remaining link capacity is the available bandwidth. Although the cross traffic and the available bandwidth share the link capacity, there are only two states of the link at a certain moment: the "idle" state and the "used" state.

In order to explain the following process more accurately, let

$\rho_i(t)$ = the instantaneous utilization state of the link i at time t

$B_i^{\tau}(t)$ = the cross traffic when the link capacity, C is used with the average utilization rate, $\rho_i^{\tau}(t)$ of the link

$A_i^{\tau}(t)$ = the available bandwidth, the unused part of the link capacity C

A^{τ} = the end-to-end available bandwidth

$A(t)$ = the available bandwidth

g_{in} = the time of input interval

g_{out} = the output interval

C_{bott} = the bandwidth of the bottleneck link

g_B = the transmission delay of the probe packet on the output link

B_f = the cross traffic in the time interval

L_p = the length of the probe packet

$f(\cdot)$ = a non-linear state transition function

w_{k-1} = independent zero-mean Gaussian white noise, the noise error of the state process

Q_{k-1} = the covariance matrix of the state process

$h(\cdot)$ = the non-linear observation function

v_k = independent zero-mean Gaussian white noise, the noise error of the observation process and is given by the covariance matrix R_k

R_k = the covariance matrix of the observation process

When the network is in a stable state, it is assumed that its cross traffic remains constant during the period of $(t, t + \tau)$, and $\rho_i(t)$ is defined as the instantaneous utilization state of the link i at time t , which is 0 when idle, and it is 1 when used, which is shown in Equation (1).

$$\rho_i(t) = \begin{cases} 0 & \text{link } i \text{ "idle"} \\ 1 & \text{link } i \text{ "used"} \end{cases} \quad (1)$$

From the instantaneous state of link i during the period of $(t, t + \tau)$, the average utilization rate $\rho_i^\tau(t)$ of link i in the interval $(t, t + \tau)$ can be calculated, which is shown in Equation (2).

$$\rho_i^\tau(t) = \int_t^{t+\tau} \rho_i(t) dt \quad (2)$$

Therefore, during the period of $(t, t + \tau)$, the cross traffic $B_i^\tau(t)$ is defined as the traffic when the link capacity C is used with the average utilization rate $\rho_i^\tau(t)$ of the link, and $B_i^\tau(t)$ is shown in Equation (3).

$$B_i^\tau(t) \equiv C_i * \rho_i^\tau(t) \quad (3)$$

Obviously, the available bandwidth $A_i^\tau(t)$ is the unused part of the link capacity C . $A_i^\tau(t)$ can be defined in Equation (4).

$$A_i^\tau(t) \equiv C_i * (1 - \rho_i^\tau(t)) \quad (4)$$

After analyzing the available bandwidth of a single link, the end-to-end available bandwidth A^τ of different paths $P(l_1, l_2, \dots, l_n)$ within the time interval $(t, t + \tau)$ can be defined. A^τ is defined in Equation (5).

$$A^\tau \equiv \min_{i=1 \dots n} (C_i * (1 - \rho_i^\tau(t))) \quad (5)$$

It is known that bandwidth measurement usually adopts active measurement methods, which mainly include three models, namely the Rate Probe Model (RPM), the Probability Probe Model (PPM), and the Gap Probe Model (GPM), etc.

RPM is a method based on the theory of self-induced congestion, which is applicable to linear systems. PRM uses a certain method to find the sending rate that will cause congestion on the link. As this rate has a large impact, it can be set to the available bandwidth of the link. PPM calculates link utilization by using probability statistics to derive available bandwidth. If the link utilization rate is high, the network congestion situation will become very serious, the error of the obtained minimum delay will also increase, and the result will also have a large deviation. In addition, when the initial filtering parameters are set unreasonably, the PPM is likely to cause large deviations in filtering and has a strong dependence on the initial parameters, which is suitable for use in the case of obtaining better filtering parameters. GPM calculates the available bandwidth of a link by comparing the changes of the packet interval before and after the probe packet passing through the bottleneck link. Moreover, GPM needs to know related information such as link status in the network, which has certain limitations.

This paper adopted the method of GPM, and assumed that there was only one bottleneck link in the network, as shown in Figure 1. Since the cross traffic has a certain effect on the sequence of the probe packets, the function between the input interval of the probe packets and the cross traffic can be obtained by analyzing the changes in the interval of the probe packets. Then, the bandwidth of the bottleneck link is subtracted from the cross traffic to obtain the available bandwidth of the link [22].

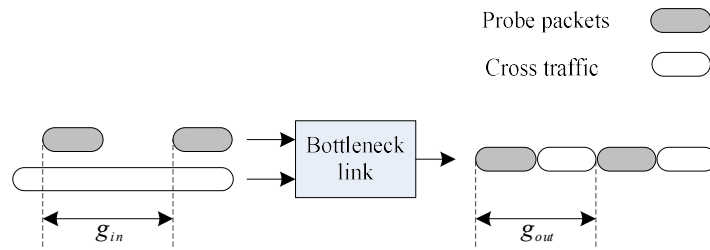


Figure 1. Diagram of probing bandwidth with probe packets.

The algorithm defines the available bandwidth $A(t)$ as shown in Equation (6).

$$A(t) = \left(1 - \frac{g_{out} - g_{in}}{g_{in}}\right) C_{bott} \quad (6)$$

where g_{in} is the time of input interval, representing the time difference between the probe packet pair P1 and P2 at the sending end. g_{out} is the output interval, which represents the time difference between the detection packet pair P1 and P2 at the receiving end. C_{bott} stands for the bandwidth of the bottleneck link. The sending interval g_{in} of the probe packet sequence is set on the source side, and the arrival time of each probe packet is recorded on the destination side. Finally, the receiving packet interval g_{out} is calculated.

In light of the above analysis, the paper introduced extended Kalman filtering into the GPM model, and established a set of available, complete, and efficient prediction models of available bandwidth based on GPM to obtain the available bandwidth of the link.

Although both CSEKB and EBE [40] can predict the available bandwidth of a wireless network based on EKF and perform congestion control accordingly, the basis of the state prediction model and observation model established by the two methods are different. EBE monitors the status or persistent queue size of each flow on the sender and predicts link bandwidth based on router queue length and packet rate. CSEKB uses the Gap Probe Model to predict bandwidth. CSEKB has low computational complexity and is more versatile in wireless networks.

3.2. System Modeling

Through a detailed analysis of the GPM in the previous section, the paper built a dynamic model based on Equation (6) including a state model and an observation model. Assume that the start time of each bit period of the transmitted data is the initial time of the system.

Definition 1. State prediction model for wireless network links

When two adjacent probe packets P1 and P2 arrive at the router one after the other, and there are data packets queued in the router [22], g_{out} satisfies Equation (7).

$$g_{out} = g_B + \frac{B_f \times g_{in}}{C_{bott}} \quad (7)$$

where g_B represents the transmission delay of the probe packet on the output link, as shown in Equation (8); B_f represents the cross traffic in the time interval when the probe packet pair P1 and P2 reach the router, and it is assumed that the length of the probe packet is L_p .

$$g_B = \frac{L_p}{C_{bott}} \quad (8)$$

Considering the change of the actual network cross traffic and sending only one data packet cannot obtain the average value of the cross traffic accurately, a series of equal-sized packet sequences were used. Therefore, the value of g_{out} can be sorted into Equation (9).

$$g_{out} = \frac{1}{C_{bott}}(L_p + B_f \times g_{in}) \quad (9)$$

Sends a set of data packets at the source, and keeps increasing the input interval g_{in} of the probe packets. The input interval g_{in} is compared with the output interval g_{out} . Once the values of g_{in} and g_{out} are equal, the transmission of the probe packet is stopped. At this time, Equation (9) is substituted into Equation (6) to obtain the bandwidth of the wireless network link, which is shown in Equation (10).

$$A(t) = 2C_{bott} - B_f - \frac{L_p}{g_{in}} \quad (10)$$

According to the above analysis, the state prediction model of the wireless network link bandwidth is defined as shown in Equation (11).

$$A_k = f(A_{k-1}, g_{in}, L_p, B_f) + w_{k-1} \quad (11)$$

where $A_k = \left[g_{in_k} \quad L_{p_k} \quad B_f \right]^T$ represents the available bandwidth of the wireless network at time k and $f(\cdot) = 2C_{bott} - B_f - \frac{L_{p_k}}{g_{in_k}}$ is a non-linear state transition function that transitions from time $k-1$ to time k . w_{k-1} is defined as independent zero-mean Gaussian white noise, which represents the noise error of the state process, and is given by the covariance matrix Q_{k-1} using covariance.

Definition 2. Observation model for wireless network links

The sending interval g_{in} of the probe packet can be set at the source, and record the arrival time of each probe packet at the destination. By calculating the interval g_{in} of the received packet and using it as an observation variable, the wireless network link bandwidth observation model is obtained, which is shown in Equation (12).

$$g_{out_k} = h(A_k, g_{in}, L_p, B_f) + v_k \quad (12)$$

where $h(\cdot) = \frac{1}{C_{bott}}(L_{p_k} + B_f \times g_{in_k})$ represents the non-linear observation function transferred from time $k-1$ to time k and v_k is defined as independent zero-mean Gaussian white noise, which represents the noise error of the observation process and is given by the covariance matrix R_k . Moreover, in the simulation experiments, the diagonal elements of Q_k and R_k can usually be set to 10^{-2} or 10^{-5} , while the non-diagonal elements are set to zero.

3.3. Bandwidth Estimation

Through the continuous prediction and correction of bandwidth based on EKF, the optimal estimate of bandwidth is obtained.

In order to explain the following process more accurately, let

A_k^f = the estimated bandwidth

A_{k-1}^a = the optimal estimate at time $k-1$

∇J_f = the Jacobian matrix of $f(\cdot)$

H.O.T._f = the higher order terms

∇J_h = the Jacobian matrix of $h(\cdot)$

H.O.T._h = the higher order terms

\hat{A}_k^a = the best unbiased estimate

K_k = the Kalman gain

P_k = the mean square error of the estimate state

3.3.1. Prediction Process

Based on the current value of the bandwidth at time $k - 1$, the estimated bandwidth A_k^f at next time k is predicted, which is shown in Equation (13).

$$A_k^f = f(A_{k-1}, g_{in}, L_p, B_f) \quad (13)$$

Expanding $f(\cdot)$ in Taylor Series, we can get Equation (14).

$$f(A_{k-1}) = f(A_{k-1}^a) + \nabla J_f(A_{k-1} - A_{k-1}^a) + H.O.T._f \quad (14)$$

where A_{k-1}^a is the optimal estimate at time $k - 1$. ∇J_f is the Jacobian matrix of $f(\cdot)$ and the higher order terms $H.O.T._f$ are considered negligible. Hence, the EKF is also called the first-order filter. The partial derivatives of g_{in} , L_p and B_f of the function $f(\cdot)$ can be calculated separately to obtain the Jacobian matrix ∇J_f , which is defined in Equation (15).

$$\nabla J_f \equiv \left[\frac{\partial f}{\partial g_{in}} \quad \frac{\partial f}{\partial L_p} \quad \frac{\partial f}{\partial B_f} \right] = \left[L_p / g_{in}^2 \quad -1 / g_{in} \quad -1 \right] \quad (15)$$

Then, the covariance matrix of the prediction error can be obtained from the error between the estimated bandwidth value and the true value, which is shown in Equation (16).

$$P_k^- = \nabla J_f P_{k-1} \nabla J_f^T + Q_k \quad (16)$$

Through the above steps, the change of various state information of the link in the wireless network can be predicted, which provides the direction for the next iteration of EKF and further accelerates the prediction of network bandwidth.

3.3.2. Calibration Process

Expanding $h(\cdot)$ in Equation (12) with the Taylor Series, we can get Equation (17).

$$h(A_k) = h(A_k^f) + \nabla J_h(A_k - A_k^f) + H.O.T._h \quad (17)$$

where A_k^f is the estimated value of bandwidth at time k ; ∇J_h is the Jacobian matrix of $h(\cdot)$; and the higher order terms $H.O.T._h$ are considered negligible. Hence, the EKF is also called the first-order filter. The partial derivatives of g_{in} , L_p and B_f of the function $h(\cdot)$ can be calculated separately to obtain the Jacobian matrix ∇J_h , which is defined in Equation (18).

$$\nabla J_h \equiv \left[\frac{\partial h}{\partial g_{in}} \quad \frac{\partial h}{\partial L_p} \quad \frac{\partial h}{\partial B_f} \right] = \left[B_f / C_{bott} \quad 1 / C_{bott} \quad g_{in} / C_{bott} \right] \quad (18)$$

In a wireless network, the actual measured value g_{out_k} (observed value) of the current bandwidth is used to modify the predicted value A_k^f obtained in the previous step. Thereby, an optimal estimate value closer to the true value of the network bandwidth is obtained, which is shown in Equation (19).

$$\hat{A}_k^a = A_k^f + K_k(g_{out_k} - \nabla J_h \cdot A_k^f) \quad (19)$$

where \hat{A}_k^a is the best unbiased estimate value and K_k is the Kalman gain. Hence, K_k is shown in Equation (20).

$$K_k = P_k \nabla J_h^T (\nabla J_h P_k \nabla J_h^T + R_k)^{-1} \quad (20)$$

Therefore, after verifying and updating the actual measurement values, the mean square error P_k of the estimate state will be reduced according to Equation (21).

$$P_k = (I - K_k \nabla J_h) P_k^- \quad (21)$$

On the basis of a thorough analysis of EKF in the previous section, this paper used the ideas and methods of prediction, correction, and recursion in EKF to predict the available bandwidth of wireless networks. Finally, the goal of using EKF to obtain the optimal estimate of available bandwidth was achieved. Moreover, in actual networks, the latest bandwidth measurement value is usually used as the initial value of the bandwidth estimation. In addition, an arbitrary non-zero value is used as an initial value of the estimation error. The proposed prediction algorithm can run in a loop automatically, and finally realize the prediction of the end-to-end bandwidth A^τ .

4. Congestion Control Algorithm

It was pointed out in the previous section that EKF has the characteristics of minimizing the estimation error under appropriate conditions. Therefore, this feature of EKF can be applied to wireless networks, and a method for predicting the available bandwidth of wireless networks is proposed in the paper. This method can accurately and efficiently predict the bandwidth and has good convergence. Moreover, the prediction result is used as the basis for congestion avoidance in the next step. Then in this section, the current congestion level of the network would be judged according to the predicted bandwidth value. In addition, the perception factor of a packet backlog is designed. When a packet loss is detected, the perception factor of the packet backlog is first calculated, and the packet loss type is refined by using the congestion threshold value. Therefore, this method provides a basis for judging noise packet loss, and the new window parameters are set accordingly, so that the parameter changes are more consistent with the actual situation of the wireless network. In addition, it also reduces the waste of bandwidth caused by a larger congestion window and further improves the transmission performance of wireless networks.

We have learned from the previous analysis that network congestion is not the only cause of packet loss in wireless networks. If noise packet loss occurs due to external interference, the source will still call the TCPW algorithm frequently and reduce the threshold $ssthresh$ to half of the size of $cwnd$, and reduce the $cwnd$ value of the sending window, that is, reduce the sending rate. However, as a matter of fact, the network is not congested at this time. This kind of processing without distinguishing the cause of packet loss will inevitably lead to a significant decrease in network throughput and affect the transmission performance of wireless networks. Therefore, in order to make up for the shortcomings caused by the TCPW algorithm using multiplicative reduction in the congestion avoidance phase, the CSEKB algorithm is proposed in the paper, which distinguishes the type of packet loss by constructing the perception factor F_{np} of the packet backlog.

Similarly, slow start, congestion avoidance, fast retransmission, and fast recovery are the four phases of the CSEKB. Although [20] proposed the Gentle Slow Start (GSS), which can smoothly transition from a slow start phase to a congestion avoidance phase. In addition, GSS can regulate the congestion window in a gentle manner based on the real-time congestion status. However, the processing in the slow start and additive growth phase of CSEKB is still consistent with the standard TCPW algorithm. The specific description of the algorithm is as follows.

(1) Calculate the optimal sending rate V_{opti} .

Whenever the sender receives an Acknowledge (ACK), it needs to record the congestion window size $cwnd$ and the round-trip delay Rtt at the current moment. In addition, the minimum round-trip delay is stored by Rtt_{min} , and the definition of the optimal transmission rate V_{opti} is shown in Equation (22).

$$V_{opti} = cwnd / Rtt_{min} \quad (22)$$

(2) Calculate the smoothed delay Rtt_{smo} .

In order to eliminate the noise interference, the method of mean filtering is used to smooth the round-trip delay Rtt . First, define a template for Rtt so that the template contains all the current delay values Rtt_{all} . Then, the smoothed delay Rtt_{smo} is defined as the average of all delays in the template, which is shown in Equation (23).

$$Rtt_{smo} = Rtt_{all}/n \quad (23)$$

Thereafter, the actual transmission rate V_{act} can be calculated from the smoothed delay Rtt_{smo} , as shown in Equation (24). It is worth noting that the transmission rate should be as close as possible to the actual available bandwidth of the network in order to make full use of network resources.

$$V_{act} = cwnd/Rtt_{smo} \quad (24)$$

(3) Constructing the perception factor of packet backlog F_{np} .

If $V_{act} > V_{opti}$, it means that in the bottleneck link, there are some accumulated data in the sender's buffer waiting to be sent, which makes the actual sending rate not reach the ideal optimal rate. Therefore, the delay is increased due to the accumulation of a large number of packets in the cache. Then, in this paper, the backlog volume can be reflected by defining the backlog perception factor F_{np} , as shown in Equation (25).

$$F_{np} = (Rtt_{smo} - Rtt_{min}) \times V_{act} \quad (25)$$

Moreover, if the transmission rate does not reach the optimal transmission rate V_{opti} , then it is necessary to calculate the perception factor F_{np} of the packet backlog. Tuan Anh and other scholars carried out detailed research on packet loss in a wireless network. First, the characteristics of the packet data backlogged in the buffer area due to noise packet loss were analyzed, then the congestion threshold characteristic values f under different bandwidth conditions were obtained. Therefore, the purpose of distinguishing the type of packet loss was achieved by comparing the perceptual factor F_{np} of the packet backlog and the characteristic value f of the congestion threshold. If $F_{np} < f$, the packet loss at this time is noise packet loss, which will not cause network congestion, and the network is still in a normal working state. Therefore, there is no need to re-estimate the network bandwidth, just fine-tune the parameters according to multiplicity reduction. Otherwise, if $F_{np} > f$, the packet loss at this time is caused by congestion, which will further affect the current congestion situation of the network.

The pseudo code of the CSEKB algorithm is shown in Algorithm 1.

The algorithm flow chart presented in the paper is shown in Figure 2.

When the multiplicative reduction mechanism of congestion is triggered in the TCPW, it does not distinguish whether the reason of packet loss is congestion or noise, but directly determines that congestion has occurred in the network. This approach will reduce throughput and degrade network performance. The CSEKB algorithm does not change its parameter settings during the slow start and additive growth phases. Once the network is congested, CSEKB designs the perception factor of the packets backlog F_{np} and distinguishes whether it is congestion or packet loss according to the type of packet loss. However, EBE [40] does not distinguish whether or not it is noise packet loss. In this way, the threshold for entering the slow start and the parameter settings of the congestion window are refined in CSEKB. When the perception factor of the packets backlog F_{np} is less than the threshold f and enters the stage of fast recovery or timeout retransmission, if the parameters are not adjusted, the congestion window will grow too fast and the real congestion will come in advance. When the message backlog perception factor F_{np} is less than the threshold value f and enters the fast recovery or timeout retransmission phase. At this time, fine-tuning the parameters will make the size of the congestion window grow quickly, and the time of real congestion will be advanced. In the bottleneck link of the wireless network, it is considered that there are still some backlog packets in the buffer of the sending end that have not been confirmed. The packets are backlogged due to noise loss. It is very

important to limit the size of the congestion window wnd , and then reserve a sufficient space for the threshold value $ssthresh$ to limit the window value to enable it to cope with noise loss and ensure the transmission quality efficiency of a wireless network.

Algorithm 1. Wireless congestion control based on EKF

```

//This algorithm would be executed after the predicted available bandwidth was updated using the extended
Kalman filter algorithm
Input:  $ssthresh$ ,  $wnd$ ,  $A^t$ ,  $Rtt_{min}$ ,  $MSS$ 
Output:  $wnd$ ,  $ssthresh$ 
Begin
//Adjust the congestion window and threshold based on the latest bandwidth estimates
For Every Receive ACK Do
Set  $Rtt_{smo} = Rtt_{all} / n$ 
If  $C_{act} > C_{opti}$  Then
Set  $Rtt_{min} = Rtt$ 
Else
Set  $C_{act} = wnd / Rtt_{smo}$ 
Set  $C_{opti} = wnd / Rtt_{min}$ 
Compute  $F_{np} = (Rtt_{smo} - Rtt_{min}) \times C_{act}$ 
If  $F_{np} < f$  Then
Set  $ssthresh = wnd$ 
Set  $wnd = wnd - F_{np}$ 
Else If Fast_Recovery==True Then
Set  $ssthresh = \frac{A^t \times Rtt_{min}}{MSS}$ 
If  $wnd > ssthresh$  Then
Set  $wnd = ssthresh$ 
Else
Set  $wnd = wnd$ 
If Retransmission==True Then
Set  $wnd = 1$ 
Set  $ssthresh = \frac{A^t \times Rtt_{min}}{MSS}$ 
End

```

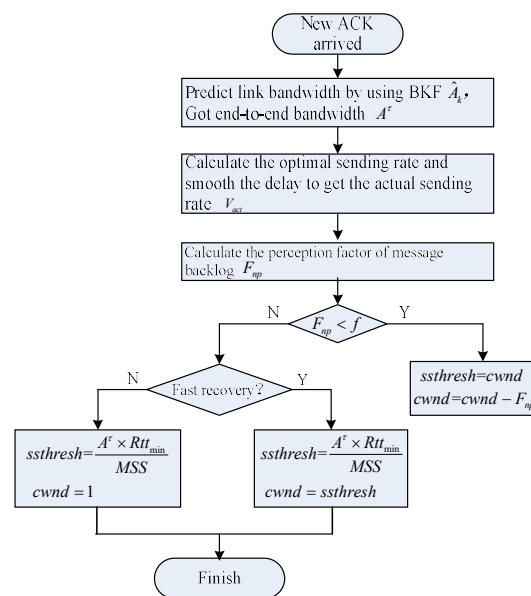


Figure 2. Flow chart of control scheme based on extended Kalman filtering and bandwidth estimation (CSEKB).

5. Simulation Experiment and Result Analysis

Network simulator 3 (NS3) is a simulation platform based on extensively absorbing the successful technologies and experiences of existing excellent open source network simulators. In addition, NS3 has a very good development environment with rich modules and open source code, which can provide high-performance, network simulation that is closer to the real network. Therefore, the paper uses NS3 as the simulation experiment environment.

In this paper, NS3 was installed on Linux Ubuntu 16.04 system as the simulation experiment environment to build a highly controllable and reusable simulation platform. Simulation experiments under different scenarios were designed to compare the performance of CSEKB with TCPW [9], CUBIC [10] and EBE [40] in terms of the size of the congestion window, throughput, and link utilization. In addition, the convergence, fairness, and friendliness of the algorithm were also tested, and was verified that the wireless network congestion control algorithm CSEKB based on bandwidth prediction had better performance.

5.1. Experimental Environment

Due to the complexity of wireless networks, it is difficult to implement different network transmission environments with different network service requirements accurately. Therefore, the most typical dumbbell topology in the network was selected as the experimental topology, and the data transmission of the bottleneck link could most effectively illustrate the performance of the proposed algorithm, so we built a dumbbell-shaped network as a simulation experiment topology. The topology diagram is shown in Figure 3.

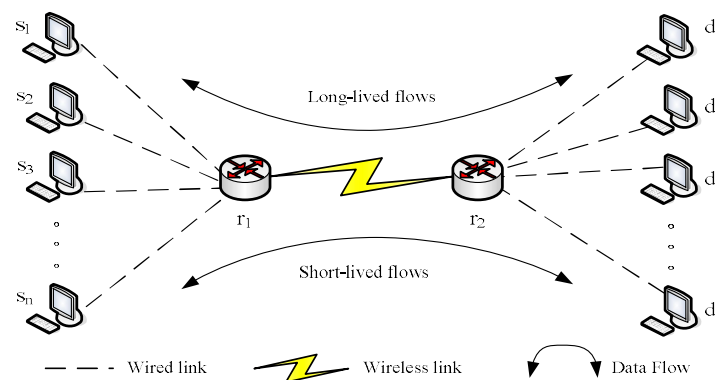


Figure 3. Topological diagram of the dumbbell network.

Nodes s_1, s_2, \dots, s_n represent the sending end of TCP, nodes r_1, r_2 are the routers of the backbone link, and nodes d_1, d_2, \dots, d_n represent the receiving end of TCP. Figure 3 shows the dumbbell topology of the simulation. It was assumed that the bandwidth of the bottleneck link changes randomly according to the uniform distribution. The end-to-end aggregated File Transfer Protocol (FTP) data stream traversing the topology contains two types, namely long-life stream and short-life stream. In the simulation experiment, 10 long-life FTP flows passed through the bottleneck link in two directions, and 20 short-life FTP flows passed through the bottleneck link from the source to the destination from left to right. The simulation parameter settings are shown in Table 1.

In order to verify the performance of the CSEKB more effectively, the wireless network with random background flow and random noise was simulated in the experiment. Moreover, these background streams could randomly join or leave the network.

Table 1. Table of simulation parameters.

Parameter	Range of Values
Bottleneck bandwidth	1 Mbps–20 Mbps
Packet size	1 KB/Package–150 KB/Package
Number of FTP streams	5–50
One-way delay	5 ms–10 ms
Send rate	20 Packets/s
BER	[0.01, 0.05]
Simulation time	[0, 150 s]

5.2. Experimental Process

In this paper, experimental tests were carried out for the short-term streams that often occur in wireless networks. The performance of CSEKB and other different congestion controls such as TCPW, CUBIC, EBE at different bit error rates and different numbers of data streams was compared. It was verified that CSEKB had better performance through the change of window, link utilization, throughput, and other QoS performance indicators.

According to the network simulation environment described in Table 1, wireless network file transmission was performed using CSEKB, TCPW, CUBIC, and EBE, and the variants in the size of the congestion window were compared. The comparison results are shown in Figure 4.

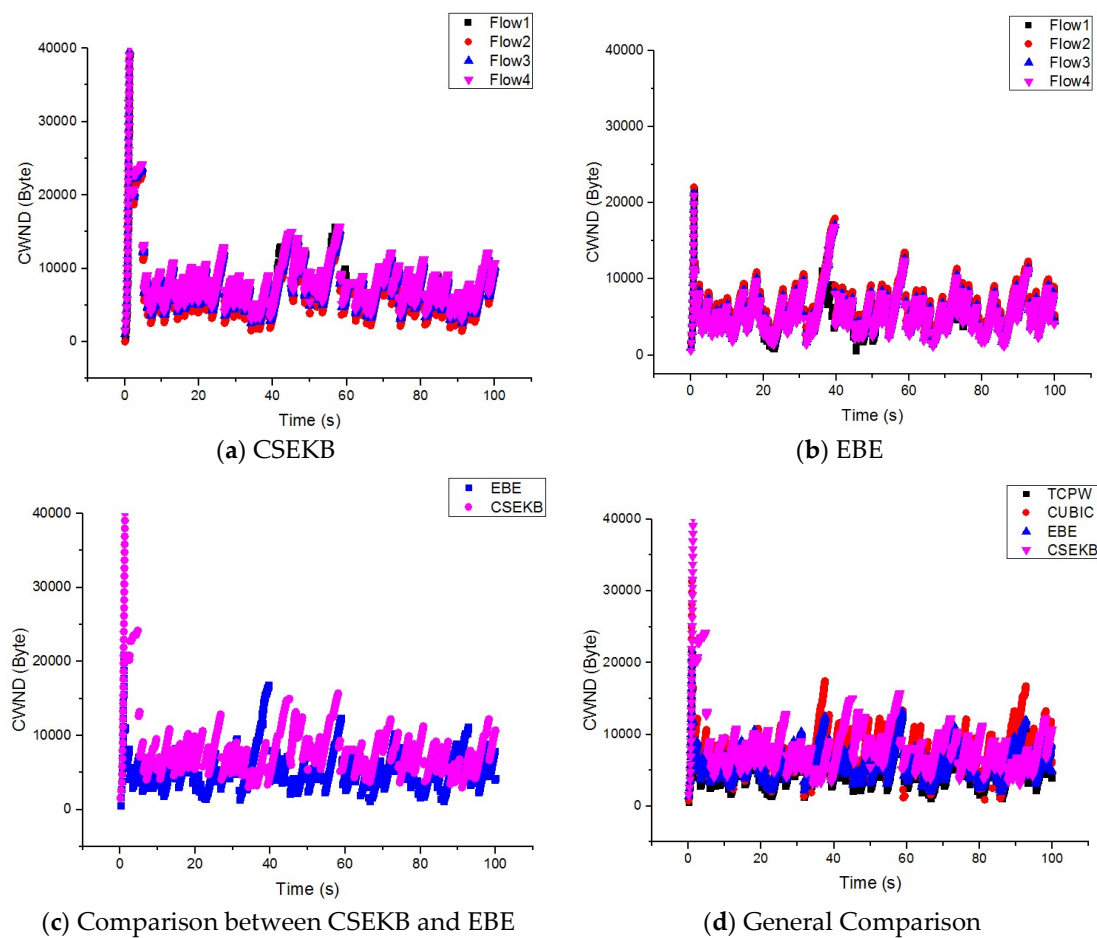
**Figure 4.** Comparison of congestion window changes.

Figure 4a,b were obtained by statistically comparing the four long-life flows used by the CSEKB and EBE, and the dynamic changes of *cwnd* in the two protocols can be obtained from the figure. Figure 4c compares the variants in the average values of *cwnd* between the CSEKB and EBE. The CSEKB could maintain the average value of *cwnd* in a relatively high and stable region. In addition, it could adapt to the changes in the bandwidth of the wireless network, showing better responsiveness and efficiency. In contrast, when *cwnd* oscillated, the ability of EBE to efficiently regulate the value of *cwnd* was slightly insufficient.

It can be seen from Figure 4d that after the perception factor of the packet backlog was introduced, compared with the TCPW, CUBIC and EBE algorithm, CSEKB had a more obvious advantage. In TCPW, once packet loss was detected, the size of the congestion window *cwnd* was reduced. When the multiplicative reduction mechanism was activated, TCPW directly reduced the window value to a minimum. The window size in CUBIC was a cubic function of time since the last loss event, so window changes were more sensitive than TCPW. This illustrates that the CSEKB and EBE could always converge, while the bandwidth changes, and the value of *cwnd* for all flows was maintained in a stable range. In contrast, the value of *cwnd* oscillated seriously, which means that CUBIC was difficult to converge. However, EBE predicted the available bandwidth by monitoring the status of each stream or the size of the persistent queue on the sender. Therefore, it took longer to converge, and was more complicated to implement.

The CSEKB uses the perceptual factors of packet backlog to distinguish the type of packet loss. If it is judged that the current packet loss is noise packet loss, the CSEKB algorithm can keep the window value *cwnd* continuing to increase and set appropriate new window parameters instead of blindly reducing the congestion window value as in TCPW. In the end, *cwnd* was maintained at a high level and network transmission performance was improved. Obviously, the CSEKB could effectively adapt to changes in bandwidth and maintained the value of *cwnd* in a relatively stable and high region, which indicates that it improved the performance of congestion control in a wireless network.

Similarly, by comparing the round-trip delay of these four algorithms when transferring files, we can see that the CSEKB had lower delay and relatively small delay jitter. From another perspective, it also shows that the more accurate the bandwidth predicted by the EKF-based bandwidth prediction method, the greater the positive effect on the performance of wireless network congestion control, and the greater the improvement in transmission performance.

We now describe the comparison of bottleneck link utilization when using the CSEKB, EBE, CUBIC and TCPW protocols, which is shown in Figure 5. TCPW could not use the increased bottleneck bandwidth effectively and the aggregation was very slow, which is mainly due to the data stream being sent synchronously from 0 to 100 s. As the flows were generated in the range of 0 to 100 s asynchronously, and CUBIC could not capture the increased bandwidth in time, CUBIC converged very slowly. It should be noted, however, that despite the poor performance of CUBIC while the bandwidth increased, its utilization was still superior to TCPW by using the active detection of the available bandwidth. When there were link bandwidth variants, EBE could respond very quickly and achieved nearly double that of the utilization during the simulation period, indicating that it had better performance than CUBIC. In contrast to TCPW, although CUBIC and EBE significantly improved the utilization of bottleneck links, its utilization was still lower than that of CSEKB. In summary, even if the bandwidth of the wireless network is unstable, the CSEKB will still shows high bottleneck link utilization.

In the same simulation environment, FTP files were used to transfer 1 KB–150 KB files. At the same time, Wireshark analysis software was used to collect real-time throughput data in the network. The experimental results of comparing the CSEKB with TCPW, CUBIC and EBE are shown in Figures 6 and 7.

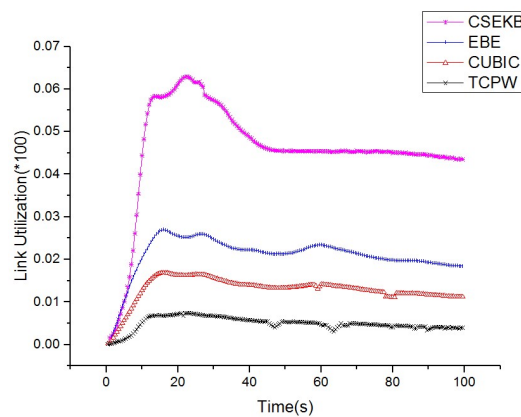


Figure 5. Comparison of bottleneck link utilization.

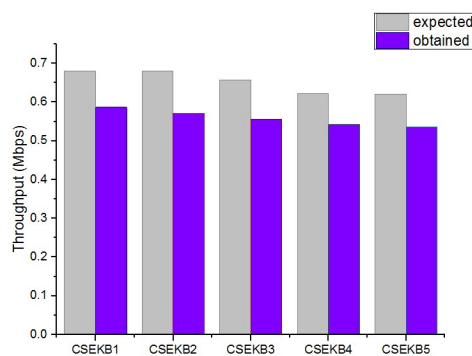


Figure 6. Comparison chart of throughput.

A comparison of the throughput achieved through combining different file sizes is shown in Figure 6. CSEKB1 represents the flow with a 10 KB file size; CSEKB2 is the flow with a 50 KB file size; CSEKB3 is the flow with a 100 KB file size; and CSEKB4 is the flow with a 150 KB file size. It can be seen from the experiment that the throughput obtained was not much different from the expected value.

Figure 7 illustrates that when the number of cross traffic flow changed, the CSEKB could achieve higher throughput than EBE, CUBIC, and TCPW.

By comparing Figure 7a, it can be seen that the network throughput of CSEKB remained at a high level and was relatively stable when transferring files. However, with traditional TCPW, the throughput was relatively low. The throughput of CUBIC was higher than TCPW. As demonstrated in Figure 7a, CUBIC was unable to obtain the increased available bandwidth. EBE could predict network bandwidth and achieved congestion control. Although the throughput of EBE was improved relative to TCPW when transferring files, the jitter was relatively large, resulting in a decrease in overall throughput. In addition, because the perception factor F_{np} of the packet backlog was introduced into the CSEKB algorithm, by comparing and judging F_{np} and the characteristic value f of the threshold, it avoids the situation where the window frequently drops rapidly and the throughput is reduced.

Next, 10 long data flows at most were added in this experiment to test the impact of the cross traffic flow on the throughput in each protocol. The cross traffic flows were sent by a specially designated PC. Figure 7b,c illustrate the comparison of the network throughput for each protocol with one cross traffic and 10 cross traffic, respectively. The throughput of the CSEKB, EBE, CUBIC, and TCPW protocols were less than the throughput without cross traffic. Fortunately, with the perception factor F_{np} , the CSEKB could achieve higher throughput than the EBE, CUBIC, and TCPW. This is because the perception factor avoids the situation where the window frequently drops rapidly and the throughput is reduced.

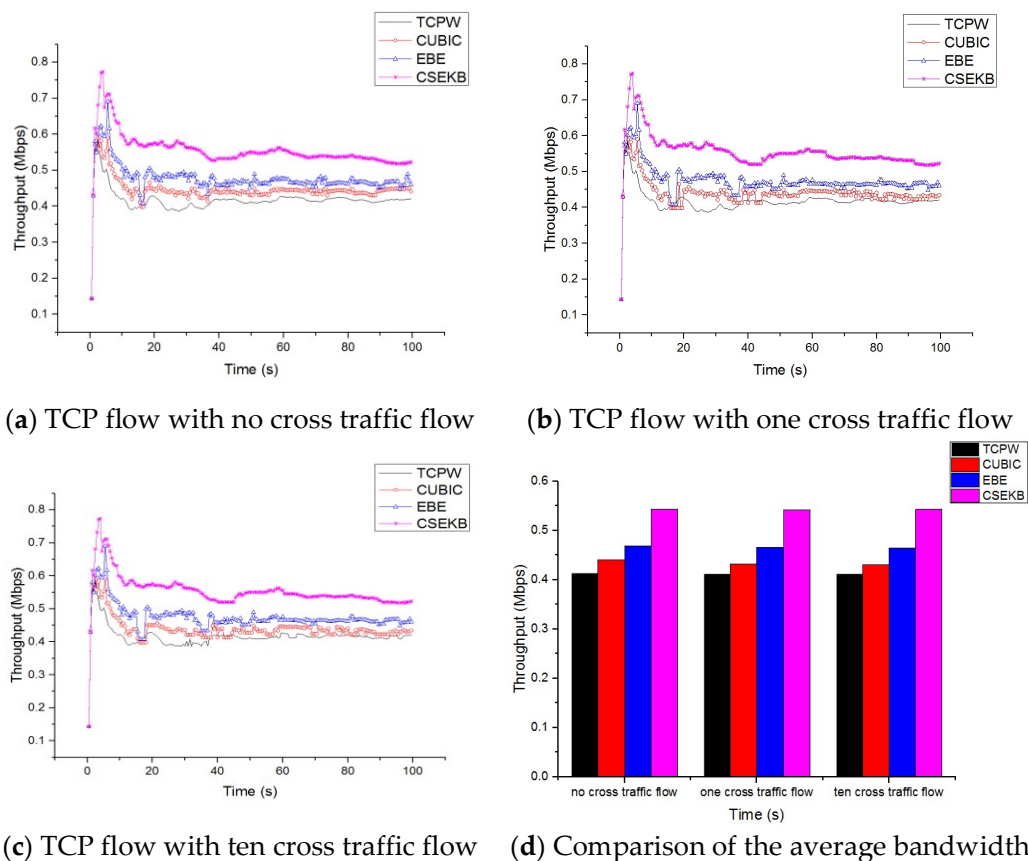


Figure 7. Comparison chart of throughput.

After a period of simulation, it will stabilize after about 20 s. At this time, the stable state can more accurately explain the bandwidth obtained by different algorithms. However, due to the problem of wireless networks, the bandwidth will be randomly changed. Therefore, the average value is used. The average bandwidth of CSEKB, EBE, CUBIC and TCPW were obtained from the stable zone and compared in Figure 7d. It can be seen that the change in the number of background streams had little effect on the performance of CSEKB. In addition, regardless of the number of background streams, the throughput obtained by CSEKB was about 5% higher than other algorithms. Obviously, compared with traditional TCPW, CUBIC, and EBE, the throughput using CSEKB was more stable, which was reflected in the significantly enhanced throughput performance of the overall data transmission process.

5.3. Fairness and Friendliness

Next, we tested the fairness and friendliness of the CSEKB algorithm through experiments in this section. Ten long-lifetime FTP streams and 20 short-lifetime FTP streams were established for testing. In the test, data streams were sent at the same time each time, and four data streams were randomly sampled and the throughput of each data stream was counted. Figure 8 shows the fairness values of CSEKB under different wireless bit error rates.

Fairness means that the bandwidth can be used evenly for each data stream in the CSEKB protocol. Moreover, it is not desirable that a certain data stream occupies a large amount of bandwidth while other data streams have no or only a small amount of bandwidth used. After a period of simulation, the average bandwidth of each data stream were obtained from the stable zone and compared. Figure 8 shows that under different circumstances of low error rate and high error rate, the throughput of each data stream in CSEKB was in the range of 0.02 Mbps–0.04 Mbps, with little difference. It can be concluded that the change of BER had little effect on the fairness of CSEKB. As the BER became

larger, each data stream of CSEKB could still share the network bandwidth fairly, which proves that the CSEKB had better fairness.

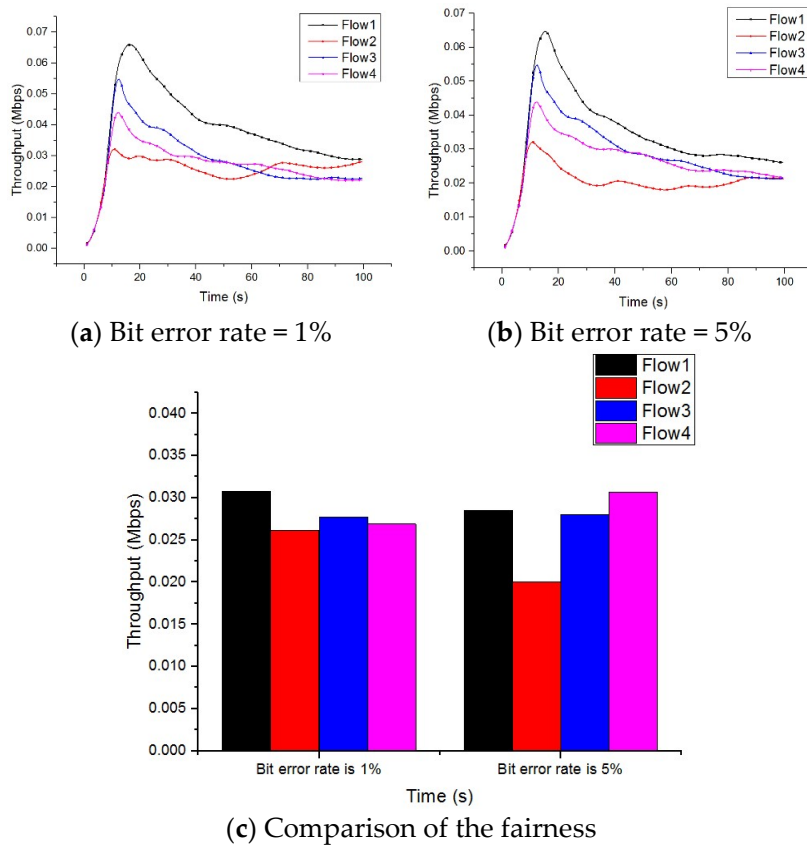


Figure 8. Comparison of fairness under different wireless bit error rates.

In addition, in order to verify that the CSEKB algorithm will not have a large impact on other TCP protocols, the following experiments were designed. The experiment still used the dumbbell topology described in Figure 3. In the experiment, four short-life-cycle data streams were established, one of which ran the CSEKB algorithm, and the other three streams ran EBE [40], CUBIC [10], and TCPW [9], respectively. Then, the average throughput of the four data streams was counted, and the friendliness of the CSEKB algorithm shown in Figure 9 was obtained.

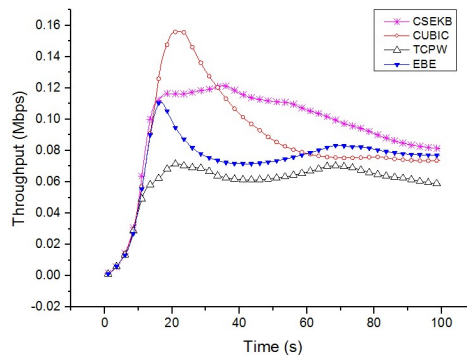


Figure 9. Friendly comparison.

Figure 9 illustrates that although the CSEKB took up a higher share of the bandwidth than other protocols, it had no significant impact on the throughput of other protocols. This indicates that the CSEKB has good friendliness.

5.4. Discussion

The CSEKB is discussed in this subsection effectively. The CSEKB, EBE, CUBIC, and TCPW were used to run 20,000 sets of experiments to achieve the statistical test. The experimental parameters and settings were the same as described in Section 5.1. The average bandwidth utilization of each protocol was compared, as shown in Table 2. It is necessary to clarify whether there is a significant difference between the mean values of the two sets of data. In this case, one-way analysis of variance (ANOVA) was used for analysis. The experimental results were analyzed using one-way ANOVA with a significance level of 0.01, as shown in Table 3.

Table 2. Bandwidth utilization (Mbps).

Protocol	N	Mean	Std. Deviation	Std. Error
CSEKB	20,000	11.89	0.0278	0.0052
EBE	20,000	10.01	0.0291	0.0053
CUBIC	20,000	9.89	0.0317	0.0057
TCPW	20,000	7.38	0.0301	0.0054

Table 3. One-way analysis of variance (ANOVA) of bandwidth utilization (Mbps).

Protocol	F	Sig.
TCPW	47,981.733	0.000
CUBIC	52,787.032	0.000
EBE	63,841.245	0.000
CSEKB	71,608.750	0.000

In all tests, ANOVA demonstrated that the performance improvements of the CSEKB were significant at a level of 0.01. One-way ANOVA for different protocols showed that there were significant differences between the two groups. According to the statistical test, the means of the CSEKB were larger than EBE, CUBIC, and TCPW. Therefore, compared with EBE, CUBIC, and TCPW, the bandwidth utilization of the CSEKB was considered to be significantly improved. Additionally, the CSEKB had about 10% improvement over EBE performance.

CSEKB used the Gap Probe Model to predict bandwidth, design the perception factors of packet backlog to distinguish the loss type, noise loss or congestion loss, and refine the setting of the slow start threshold and the size of the congestion window. CSEKB had low computational complexity and was more universal in wireless networks with low latency. However, when the bandwidth-delay product was high, its performance was not better than EBE. EBE predicts link bandwidth by taking advantage of changes in queue length, and EBE does not discriminate between loss types. When the queue length oscillated suddenly, EBE did not give significant performance improvement in such cases.

Compared with CUBIC and TCPW, EBE was able to map the changes in latency to the changes in bandwidth. In addition, EBE's estimation of available bandwidth was more accurate than that of CUBIC and TCPW. However, in low delay and low bandwidth networks, the variations of bandwidth and delay may be relatively small, which may lead to insufficient variations of delay. Therefore, EBE is more suitable for high delay and high bandwidth networks.

Finally, it is important to realize that the CSEKB complements the congestion control protocol in wireless networks. The CSEKB does not change the control algorithm for such a protocol, so their behavior is not affected significantly.

6. Conclusions

In this paper, an accurate estimation of bandwidth in wireless networks was given by extending the Kalman filter algorithm. Based on this prediction value, the congestion window was adjusted appropriately to improve the congestion control of the wireless network. Aiming at the practical

characteristics of significant random bandwidth changes in wireless networks, the CSEKB algorithm was proposed in the paper. The CSEKB utilized the feature where EKF can reduce the estimation error to the minimum under appropriate conditions, and can effectively predict the available bandwidth based on EKF that is suitable for wireless networks. The CSEKB can dynamically adjust the size of the send window based on a reasonable, accurate, and real-time estimation of the current available bandwidth in the wireless network. Moreover, the CSEKB can effectively distinguish between congestion loss and noise loss, according to the type of the packet loss, and then use corresponding congestion control methods. In this way, it can effectively avoid blindly limiting the transmission rate and reduce the transmission performance of the wireless network as well as improve the wireless network resource utilization.

In the NS3 simulation environment, the CSEKB was compared with existing TCP algorithms in terms of throughput, link utilization, and so on. The results of simulation experiments effectively prove that CSEKB can accurately predict the available bandwidth, improve network throughput, and achieve the purpose of avoiding congestion. It was further verified that the CSEKB had good friendliness and fairness. In addition, the experimental data showed that the CSEKB had about 10% improvement over EBE performance. Moreover, these illustrate the CSEKB can significantly enhance the transmission performance of wireless networks and optimize the overall performance of wireless networks.

It should be noted that the proposed CSEKB is only applicable to wireless networks and not to data centers. In a data center network, data are sent to the same node using a large number of transmission control protocol (TCP) flows. If there is huge packet loss and significant timeout occurs, this is called TCP Incast, the congestion crash. However, the CSEKB does not consider the issue of TCP Incast. The CSEKB cannot solve the situation of congestion collapse caused by a large number of packet collisions in shared media. In the context of intelligent network development, congestion control in a data center should have strong adaptive capabilities and high control efficiency. In the future, machine learning will be deployed in the congestion control of a data center. In reinforcement learning models, agents can interact with the environment to learn and make good decisions.

Author Contributions: H.W. conceived and designed the whole system; H.W. and B.H. designed the algorithm and model; H.W. and J.T. conducted the experiment and analyzed the data; H.W. wrote the research paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research and Development Plan Project of Shaanxi Science and Technology Department (Grant No. 2017ZDXM-GY-016), the Project of Shaanxi Natural Science Basic Research (Grant No. 2020JM-565), The project of students' innovation and entrepreneurship training program (Grant No. S201910702022), The project of students' innovation and entrepreneurship training program (Grant No. X201910702141), the Project of Department of Education Science Research of Shaanxi Province (Grant No. 17JK0371), the Industrial Research Project of Science and Technology Department of Shaanxi Province (Grant No. 2016KTZDGY4-09), the Pre-Research Project of 13th Five-year Equipment Development (Grant No. 41402020202), the Research Project on Teaching Reform of Education in Shaanxi Province (Grant No. 17JZ004, 17Y015), the Fund of National Laboratory of Network and Detection Control (Grant No. GSYSJ2017007), the Characteristic Disciplines in Education Department of Shaanxi Province (Grant No. 080901), the Key Project of Educational Reform at the School Level of Xi'an Technological University (Grant No. 18JGZ01), the Project of Innovation and Entrepreneurship Training Program for College Students at the National Level (Grant No. 1070214030).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cardwell, N.; Cheng, Y.; Gunn, C.S.; Yeganeh, S.H.; Jacobson, V. BBR: Congestion-Based Congestion Control. *Queue* **2016**, *14*, 50. [[CrossRef](#)]
2. Ru, X.; Liu, Y. A New Dynamic Adjustment Strategy of Transmission Control Protocol Congestion Window. *J. Henan Univ. Sci. Technol. (Nat. Sci.)* **2018**, *39*, 45–57.
3. Wang, J.; Gong, H.; Chen, J. A Cooperant Congestion Control Protocol in High Bandwidth-Delay Product Networks. *J. Softw.* **2008**, *19*, 125–135. [[CrossRef](#)]
4. Li, Z.P.; Huang, M.; Yang, Y.; Wang, D.Q. QoS Based Adaptive Congestion Control Scheme in Wireless Multi-Hop Networks. *J. Syst. Simul.* **2008**, *18*, 4889–4894.

5. Xu, L.; Yang, X.; Wang, Y. Information Optimization Control Simulation of Congestion Link in Wireless Networks. *Comput. Simul.* **2019**, *36*, 285–288.
6. Wang, Z.M.; Zeng, X.P.; Liu, X.; Chen, L.; Guo, B. TCP Congestion Control Algorithm for Heterogeneous Networks. *J. Electron. Inf. Technol.* **2016**, *38*, 780–786.
7. Li, J.; Zhang, L.; Feng, X.; Jia, K.; Kong, F. Feature Extraction and Area Identification of Wireless Channel in Mobile Communication. *J. Internet Technol.* **2019**, *20*, 544–553.
8. Nan, F.; Li, Y.; Jia, X.; Dong, L.; Chen, Y. Application of improved SOM network in gene data cluster analysis. *Measurement* **2019**, *145*, 370–378. [[CrossRef](#)]
9. Bao, W.; Wong, V.W.S.; Leung, V.C.M. A Model for Steady State Throughput of TCP CUBIC. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2010), Miami, FL, USA, 6–10 December 2010.
10. Sangtae, H.; Rhee, I.; Lisong, X. CUBIC: A new TCP-friendly high-speed TCP variant. *Acm Sigops Oper. Syst. Rev.* **2008**, *42*, 64–74.
11. Mascolo, S.; Casetti, C.; Gerla, M.; Sanadidi, M.Y.; Wang, R. TCP westwood: Bandwidth estimation for enhanced transport over wireless links. In Proceedings of the 7th annual international conference on Mobile computing and networking, London, UK, September 21–25 2001; pp. 287–297.
12. Wu, D.; Che, W. Research on Congestion Control Algorithms Based on Bandwidth Estimation in Hybrid Network. *J. Chongqing Inst. Technol. (Nat. L. Sci.)* **2010**, *24*, 62–65.
13. Meng, X.; Liu, J.; Wang, L. Research on TCPW improvement in hybrid network. In Proceedings of the IEEE International Conference on Computer Science & Automation Engineering, Shanghai, China, 10–12 June 2011.
14. Zhao, Y.; Liu, H.; Zhang, X. Wireless TCP Congestion Control Based on Bandwidth Estimation. *Comput. Simul.* **2015**, *32*, 285–289.
15. Juang, H.; Huang, J.; Wang, X. Simulation of TCP congestion control based on bandwidth estimation in MANET. *Comput. Eng. Des.* **2016**, *37*, 201–205.
16. Ren, J.; Wang, S.; Wen, C. An improved TCPW algorithm applied in congestion control. *Microcomput. Appl.* **2017**, *36*, 63–65.
17. Yuan, C.; Dong, Y. Wireless-loss aware cross-layer congestion control algorithm in 4G networks. *J. Nanjing Univ. Posts Telecommun. (Nat. Sci. Ed.)* **2016**, *36*, 65–73.
18. Li, S.Y.; Wang, X.J.; Xu, D.; Liu, Q.; Qian, J.S. Method of Available Bandwidth Estimation for TCP Based on Channel Noise Model. *J. Syst. Simul.* **2008**, *20*, 5058–5061.
19. Zhao, H.; Zhu, H.; Liu, N.; Dong, Y.; Zhang, H. Joint Optimization of Statistical Bandwidth Based on Interference-Aware Probability Prediction for Wireless Heterogeneous Mesh Networks. *J. Nanjing Univ. Posts Telecommun. (Nat. Sci.)* **2014**, *34*, 19–26.
20. Valdovinos, I.A.; Diaz, J.A.P.; Villalba, L.J.G.; Kim, T.H. BATCP: Bandwidth-Aggregation Transmission Control Protocol. *Symmetry* **2017**, *9*, 167. [[CrossRef](#)]
21. Ahmed, H.; Arshad, M.J. Buffer Occupancy-Based Transport to Reduce Flow Completion Time of Short Flows in Data Center Networks. *Symmetry-Basel* **2019**, *11*, 646. [[CrossRef](#)]
22. Tao, Y.; Li, J.; Ou, H. Goodput improvement based on controlling difference of delay. *Comput. Eng. Des.* **2018**, *39*, 320–325. [[CrossRef](#)]
23. Angrisani, L.; Miele, G.; Moriello, R.S.L.; Vadursi, M. A Kalman filtering based method for available bandwidth measurement. In Proceedings of the IEEE Instrumentation & Measurement Technology Conference, Pisa, Italy, 11–14 May 2015.
24. Kim, P.S. A finite memory filtering for end-to-end available bandwidth estimation. *Int. J. Control Autom. Syst.* **2013**, *11*, 206–210. [[CrossRef](#)]
25. Nguyen, T.M.T.; Pujolle, G. Kalman Filter Based Bandwidth Estimation and Predictive Flow Distribution for Concurrent Multipath Transfer in Wireless Networks. In Proceedings of the 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, Beijing, China, 21–23 September 2012.
26. Sedighizad, M.; Seyfe, B.; Navaie, K. *MR-BART: Multi-Rate Available Bandwidth Estimation in Real-Time*; Elsevier Ltd.: Amsterdam, The Netherlands, 2008.
27. Zhang, W. *Research on Application of Kalman Filter in Bandwidth Measurement*; University of Electronic Science and Technology of China: Guangdong, China, 2015.
28. Li, X.; Zhao, Z.; Liu, L.; Liu, Y.; Li, P. An Optimization Model of Multi-Intersection Signal Control for Trunk Road under Collaborative Information. *J. Control Sci. Eng.* **2017**, *2017*, 11. [[CrossRef](#)]

29. Lioris, J.; Pedarsani, R.; Tascikaraoglu, F.Y.; Varaiya, P. Platoons of Connected Vehicles Can Double throughput in Urban Roads. *Transp. Res. Part C Emerg. Technol.* **2017**, *77*, 292–305. [[CrossRef](#)]
30. Talebpour, A.; Mahmassani, H.S. Influence of Connected and Autonomous Vehicles on Traffic Flow Stability and Throughput. *Transp. Res. Part C Emerg. Technol.* **2016**, *71*, 143–163. [[CrossRef](#)]
31. Day, C.M.; Li, H.; Richardson, L.M.; Howard, J.; Platte, T.; Sturdevant, J.R.; Bullock, D.M. Detector-Free Optimization of Traffic Signal Offsets with Connected Vehicle Data. *Transp. Res. Rec. J. Transp. Res. Board* **2017**, *2620*, 54–68. [[CrossRef](#)]
32. Lin, X.; Xu, J.; Lin, P.; Cao, C.; Liu, J. Improved Road Network-Flow Control Strategy Based on Macroscopic Fundamental Diagrams and Queuing Length in Connected-Vehicle Network. *Math. Probl. Eng.* **2017**, *2017*, 7. [[CrossRef](#)]
33. Li, X.; Sun, J.Q. Signal Multi objective Optimization for Urban Traffic Network. *Ieee Trans. Intell. Transp. Syst.* **2018**, *19*, 3529–3537. [[CrossRef](#)]
34. Wada, K.; Usui, K.; Takigawa, T.; Kuwahara, M. An Optimization Modeling of Coordinated Traffic Signal Control Based on the Variational Theory and its Stochastic Extension. *Transp. Res. Procedia* **2017**, *23*, 624–644. [[CrossRef](#)]
35. Xiong, Z.; Wu, Y.; Ye, C.; Zhang, X.; Xu, F. Color image chaos encryption algorithm combining CRC and nine palace map. *Multimed. Tools Appl.* **2019**, *78*, 31035–31055. [[CrossRef](#)]
36. Wen, L.; Wenbo, W.; Xiaojun, J.; Wen, L. CMT performance optimization algorithm based on union prediction of bandwidth and round trip time. *Chin. J. Eng.* **2015**, *37*, 132–143.
37. Kadota, I.; Baiocchi, A.; Anzaloni, A. Kalman Filtering: Estimate of the numbers of active queues in an 802.11e EDCA WLAN. *Comput. Commun.* **2013**, *39*, 54–64. [[CrossRef](#)]
38. Ahmed, A.; Ngoduy, D.; Watling, D. Prediction of Traveller Information and Route Choice Based on RealTime Estimated Traffic State. *Transp. B Transp. Dyn.* **2016**, *4*, 23–47.
39. Cavusoglu, B.; Oral, E.A. Estimation of available bandwidth share by tracking unknown cross-traffic with adaptive extended Kalman filter. *Comput. Commun.* **2014**, *47*, 34–50. [[CrossRef](#)]
40. Huang, Z.; Li, X.; Yousefi-Zadeh, H. Robust EKF-Based Wireless Congestion Control. *Ieee Int. Conf. Commun.* **2013**, *61*, 5090–5102.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).