



Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control

Shaocheng Qu¹ · Liang Zhao¹ · Zhili Xiong^{1,2}

Received: 11 November 2019 / Accepted: 23 January 2020 / Published online: 6 February 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Wireless sensor networks (WSNs) act as a building block of Internet of Things and have been used in various applications to sense environment and transmit data to the Internet. However, WSNs are very vulnerable to congestion problem, resulting in higher packet loss ratio, longer delay and lower throughput. To address this issue, this paper presents a fuzzy sliding mode congestion control algorithm (FSMC) for WSNs. Firstly, by applying the signal-to-noise ratio of wireless channel to TCP model, a new cross-layer congestion control model between transmission layer and MAC layer is proposed. Then, by combining fuzzy control with sliding mode control (SMC), a fuzzy sliding mode controller (FSMC) is designed, which adaptively regulates the queue length of buffer in congested nodes and significantly reduces the impact of external uncertain disturbance. Finally, numerous simulations are implemented in MATLAB/Simulink and NS-2.35 by comparing with traditional control strategies such as fuzzy, PID and SMC, which show that the proposed FSMC effectively adapts to the change of queue length and has good performance, such as rapid convergence, lower average delay, less packet loss ratio and higher throughput.

Keywords Wireless sensor networks (WSNs) · Internet of Things (IoT) · Cross-layer congestion control · Fuzzy sliding mode control · NS-2.35

1 Introduction

Nowadays, Internet of Things (IoT) is playing a significant role in achieving various real-world applications, such as agricultural monitoring, intelligent transportation, disaster prediction and smart cities [1]. Within IoT framework, wireless sensor networks (WSNs) act as a bridge that connects the “things” in the physical world to the virtual digital world. In a WSN, a group of tiny sensors or actuators are wirelessly connected with each other, capable of collecting, computing and transmitting sensory data to the Internet [2, 3]. However, owing to open characteristics like large-scale, self-organizing, dynamic topology and battery-

constrained, WSNs are very vulnerable to congestion problem, resulting in unsatisfactory situations such as higher packet loss ratio, longer delay and lower throughput [4]. Generally speaking, there are two representative reasons for congestion problem in WSNs [5]: (1) buffer overflow and (2) link collision. As shown in Fig. 1a, buffer overflow (node-level congestion) occurs when the number of transmitted packets exceeds the packet handling capacity of a particular node (node A), leading to packets drops and a waste of nodes' energy. On the other hand, link collision (link-level congestion) occurs when many active nodes attempt to communicate with a particular node (node B) simultaneously, causing packet loss due to competition and interference, as shown in Fig. 1b.

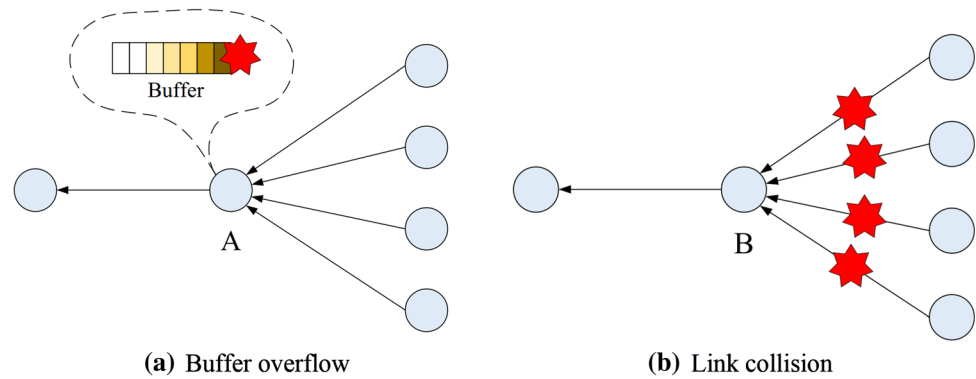
In recent years, many researches have been carried out on the congestion control of WSNs and are mainly divided into two categories [6]: (1) traffic-based control and (2) resource-based control. In traffic-based congestion control schemes, the data rate of incoming flows from the downstream nodes is adjusted against the forwarding capacity of the upstream node(s). For instance, Javid et al. [7]

✉ Zhili Xiong
zhilixiong@mails.cnu.edu.cn

¹ Department of Electronics and Information Engineering,
College of Physical Science and Technology, Central China
Normal University, Wuhan 430079, China

² School of Electric Information, Huanggang Normal
University, Huanggang 438000, China

Fig. 1 Typical congestion occurrence scenarios in WSNs



proposed a traffic-aware congestion control protocol (TACC) that operates on the end-to-end principle at the transport layer of wireless multimedia sensor networks (WMSNs). TACC uses burst loss information to detect congestion at the destination and directs source nodes to adjust reporting rate accordingly. However, TACC should be further improved to support prioritized event transport to multiple flows. Alipio and Tiglao [8] developed a new reliable transport protocol with a cache-aware congestion control mechanism called RT-CaCC. RT-CaCC utilizes cache management policies such as cache insertion, cache elimination and cache size allocation to mitigate packet losses in WSNs while maximizing cache utilization and bandwidth allocation. Nevertheless, the analysis of energy consumption, another significant metric of WSNs, is ignored. Zhuang et al. [9] analyzed the impact of congestion control on the data accuracy and proposed a Congestion-Adaptive Data Collection scheme (CADC) to efficiently resolve the congestion under the guarantee of data accuracy. CADC mitigates congestion by adaptive lossy compression while ensuring a given overall data estimation error bound in a distributed manner. Sonmez et al. [10] proposed a sensor fuzzy-based image transmission (SUIT) for wireless multimedia sensor networks (WMSNs). SUIT decreases the image quality on the fly to an acceptable level and thus improves the continuity of the video streaming. But SUIT does not use any ACK/NACK, resulting in receiver can only know about the loss of packets through packet sequence numbers. By combining standard particle swarm optimization (PSO) with single neuron PID control, Yang et al. [11] designed a standard PSO-neural PID congestion controller (PNPID), which outperformed the traditional PID controller in terms of convergence and accuracy. However, the performance of PNPID in different scenarios needs to be verified.

Resource-based congestion control, on the other hand, exploits the idle network resources to balance the traffic load whenever congestion arises [12]. For example, Nikokheslat and Ghaffari [13] used a hierarchical tree and grid structure to produce an initial topology and then

utilized Prim's algorithm to find appropriate neighbors. In their proposed method, a hierarchical tree structure was used to produce network topologies and a resource control algorithm was used as a factor to control congestion in WSNs. However, more convincing experiments should be performed to further justify their method. Kafi et al. [14] designed an efficient congestion control-based schedule algorithm, dubbed REFIACC (Reliable, Efficient, Fair and Interference-Aware Congestion Control) protocol. REFIACC prevents the interferences and ensures high fairness of bandwidth utilization among sensor nodes by scheduling the communications. The congestion and the interference in inter- and intra-paths hot spots are mitigated through taking into account the dissimilarity between links' capacities at the scheduling process. Li et al. [15] proposed a discrete quasi-sliding mode control strategy (SMC) for link-level congestion in WSNs. By giving packets with minimum data a higher priority to be transmitted, queuing time can be decreased and congestion can be alleviated. Although MATLAB simulation results demonstrate the effectiveness of SMC, network characteristics of SMC should be comprehensively evaluated by using a dedicated WSNs simulation tool like NS-2.

To the best of our knowledge, this paper makes the following contributions: (1) We propose a cross-layer congestion control mathematical model between transmission layer and MAC layer by applying the signal-to-noise ratio (SNR) of wireless channel to transmission control protocol (TCP) model. (2) We design a fuzzy sliding mode control algorithm (FSMC) by combining fuzzy control with sliding mode control. FSMC not only adaptively regulates the queue length of buffer in congested nodes, but also effectively reduces the impact of uncertain disturbance and thus weakens the chattering phenomenon of queue length.

The rest of this paper is organized as follows: Sect. 2 introduces the mathematical model of cross-layer congestion control in WSNs. Section 3 describes the design of fuzzy sliding mode controller. Section 4 evaluates the simulation results in MATLAB/Simulink and NS-2.35,

respectively. Finally, Sect. 5 concludes the paper and presents our future work.

2 Mathematical model

2.1 A SNR-based cross-layer congestion control model

Reference [16] proposed a dynamical model of TCP behavior by using fluid flow and stochastic differential equation analysis, while, in this paper, we present a novel cross-layer version of that model that uses the signal-to-noise ratio instead of packet-marking probability, as shown in Fig. 2. It can be seen that based on the received SNR from the wireless channel, the fuzzy sliding mode controller can adjust the queue length of buffer in MAC layer by dropping packets. Therefore, the proposed cross-layer model can be described by the following nonlinear differential equations:

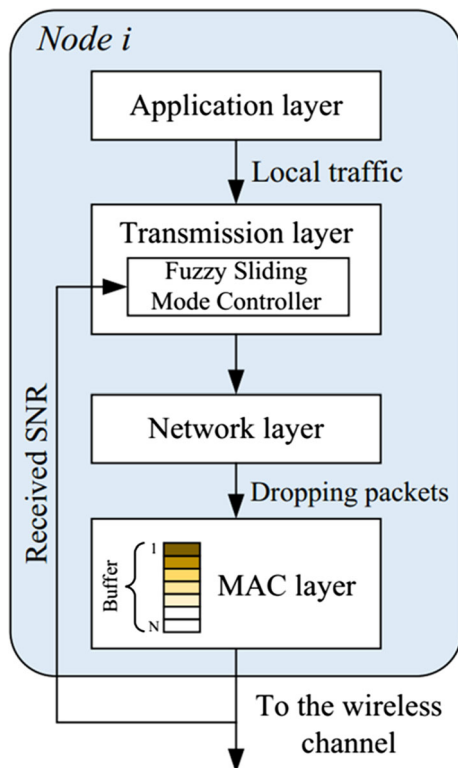


Fig. 2 Cross-layer congestion control model

$$\begin{aligned} \dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{2} \frac{W(t-R(t))}{R(t-R(t))} S_{nr}(t-R(t)) \\ S_{nr}(t) &= 1 - \frac{SNR(t)}{20} \\ q(t) &= \begin{cases} -C(t) + \frac{N(t)}{R(t)} W(t), & q > 0 \\ \max\left\{0, -C(t) + \frac{N(t)}{R(t)} W(t)\right\}, & q = 0 \end{cases} \end{aligned} \tag{1}$$

where $W(t)$ is the average TCP window size, $C(t)$ is the link capacity, $N(t)$ is the load factor, $S_{nr}(t)$ is the corresponding value of the $SNR(t) \in (0, 20]$, $R(t)$ is the round-trip time and $R(t) = T_p + \frac{q(t)}{C(t)}$, $q(t)$ is the average queue length and T_p is the propagation delay.

The first differential equation in (1) describes the TCP window control dynamic. To linearize (1), we assume that $N(t)$, $C(t)$ and $R(t)$ are constants, i.e., $N(t) \equiv N$, $C(t) \equiv C$ and $R(t) \equiv R$. Consider (W, q) as the state and S_{nr} as the input, the operating point (W_0, q_0, S_{nr0}) is then defined by $\dot{W} = 0$ and $\dot{q} = 0$. Hence, we have

$$\begin{cases} \dot{W} = 0 \Rightarrow W_0^2 S_{nr0} = 2 \\ \dot{q} = 0 \Rightarrow W_0 = \frac{R_0 C}{N}; R_0 = \frac{q_0}{C} + T_p \end{cases} \tag{2}$$

By linearizing (1) about the operating point (W_0, q_0, S_{nr0}) , it yields

$$\begin{cases} \delta \dot{W}(t) = -\frac{2N}{R_0^2 C} \delta W(t) - \frac{R_0 C^2}{2N^2} \delta S_{nr}(t-R_0) \\ \delta \dot{q}(t) = \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \end{cases} \tag{3}$$

where $\delta W(t) = W(t) - W_0$, $\delta q(t) = q(t) - q_0$, and $\delta S_{nr}(t) = S_{nr}(t) - S_{nr0}$.

Let $x_1(t) = \delta q(t)$, $x_2(t) = \delta \dot{q}(t)$ and $u(t) = \delta S_{nr}(t)$ ($S_{nr0} \leq u(t) \leq 1 - S_{nr0}$); then, the system state equation can be written as:

$$\dot{x}(t) = Ax(t) + Bu(t-\tau), \tag{4}$$

where $x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$, $A = \begin{pmatrix} 0 & 1 \\ -\frac{2N}{R_0^3 C} & -\left(\frac{1}{R_0} + \frac{2N}{R_0^2 C}\right) \end{pmatrix}$,

$B = \begin{pmatrix} 0 \\ -\frac{C^2}{2N} \end{pmatrix}$ and $\tau = R_0$.

Assume that $R_0 \gg \frac{N}{C}$, and then, the transfer function of cross-layer congestion model can be described as:

$$G(s) = \frac{\delta q(t)}{\delta S_{nr}(t)} = \frac{\frac{C^2}{2N}}{\left(s + \frac{2N}{R_0^3 C}\right) \left(s + \frac{1}{R_0}\right)}. \tag{5}$$

2.2 Dynamics of the proposed model

In this section, the objective is to analyze the dynamics of the proposed model described in (5), which shows that $SNR(t)$ dynamically affects the queue length.

For (5), we consider the case when $T_p = 0.05s$, $C = 650$ packets/s and $N = 25$. If $SNR(t) = 18$ dB, then we have $W_0 = 2\sqrt{5}$ and $R_0 = 0.1720$, so (5) becomes

$$G(s) = \frac{8450}{(s + 2.6002)(s + 5.8138)} = \frac{8450}{s^2 + 8.414s + 15.117} \tag{6}$$

If $SNR(t) = 12$ dB, then we have $W_0 = \sqrt{5}$ and $R_0 = 0.0860$, so (5) becomes

$$G(s) = \frac{8450}{(s + 10.4006)(s + 11.6276)} = \frac{8450}{s^2 + 22.0282s + 120.934} \tag{7}$$

The magnitude and phase Bode plots for (6) and (7) are shown in Figs. 3 and 4, respectively. It can be seen that phase margin of Fig. 3 is 5.25° and that of Fig. 4 is 13.8° . According to the Nyquist theorem, systems (6) and (7) are stable because their phase margin is positive value. Furthermore, if $SNR(t)$ decreases, then S_{nr} will increase, and the value range of $u(t)$ will also change accordingly, which contributes to achieving the proper queue length. SNR has a great influence on the performance of WSNs, because unreasonable SNR will cause longer delay, higher packet loss ratio and even serious network congestion. Hence, the proposed SNR-based cross-layer congestion model is innovative and feasible.

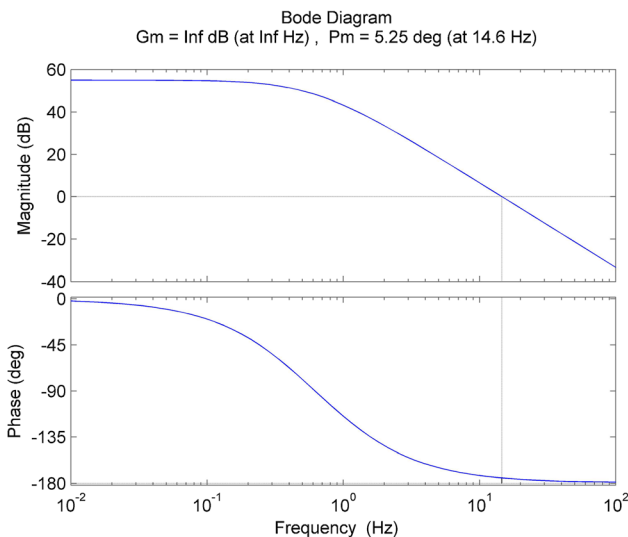


Fig. 3 Bode plots when $SNR(t) = 18$ dB

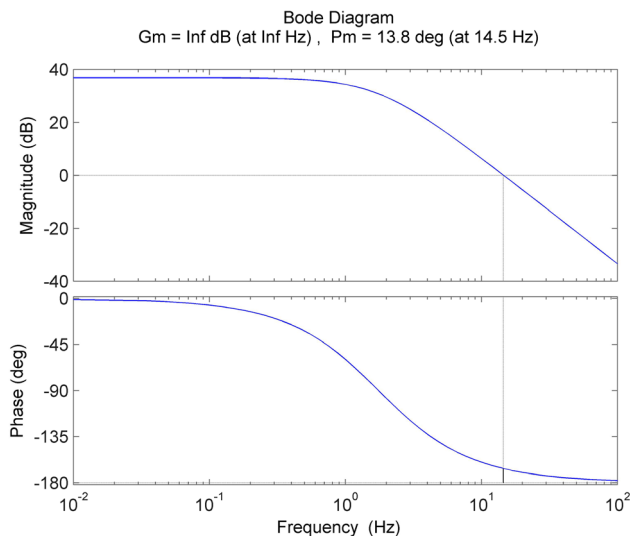


Fig. 4 Bode plots when $SNR(t) = 12$ dB

3 Design of fuzzy sliding mode controller

WSN is a complicated distributed system with features like nonlinearity, time variance and uncertainty [17, 18]. In order to overcome the drawbacks such as chattering action and sensitive to disturbance of the traditional sliding mode controller, a fuzzy sliding mode controller (FSMC) is designed by combining fuzzy controller [19, 20] with the sliding mode controller, as shown in Fig. 5. It can be seen from Fig. 5 that the sliding mode controller first uses the error between the desired and actual queue lengths to figure out ss , then gives ss to fuzzy controller to produce a reasonable $\hat{K}(t)$, finally utilizes $\hat{K}(t)$ to self-adjust and outputs a control u that can reduce the impact of uncertain disturbance $d(t)$. Thanks to its rapid response, simplicity and robustness, insensitivity to parameter variations, and rejection of disturbances and chattering, FSMC is very applicable to solving congestion problem of WSN systems.

3.1 System description

Consider a second-order uncertain system:

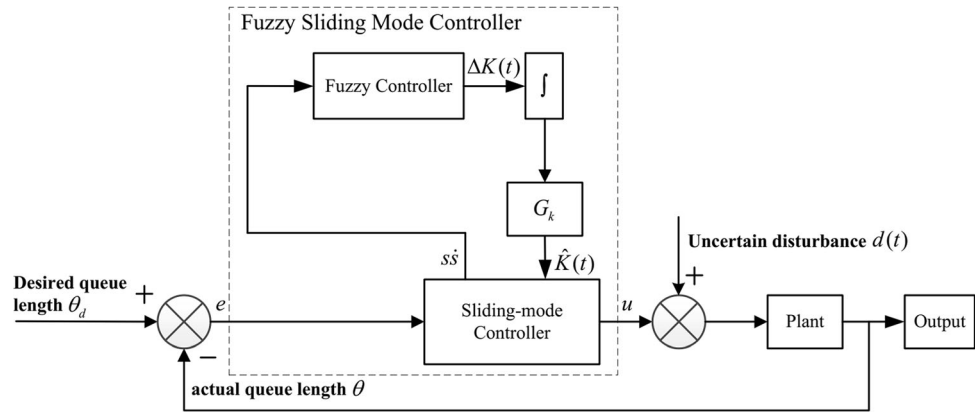
$$G(s) = \frac{k}{s^2 + as + b}, \tag{8}$$

and system (8) can also be written as:

$$\ddot{\theta} = -a\dot{\theta} - b\theta + ku(t) + d(t), \tag{9}$$

where θ is the queue length signal, $u(t)$ is the control input, $d(t)$ is the unknown disturbance, $a = \frac{1}{R_0} + \frac{2N}{R_0^2 C}$, $b = \frac{2N}{R_0^2 C}$ and $k = \frac{C^2}{2N}$ are known parameters.

Fig. 5 Architecture of fuzzy sliding mode controller



3.2 Design and analysis of sliding mode controller

Define sliding mode function as:

$$s = ce + \dot{e}, \tag{10}$$

where $c > 0$, e is tracking error and $e = \theta_d - \theta$, θ_d and θ are the desired and actual queue length, respectively.

The sliding mode controller is designed as:

$$u(t) = \frac{1}{k} (a\dot{\theta} + b\theta + \ddot{\theta}_d + c\dot{e} + K(t)\text{sgn}(s)), \tag{11}$$

where $K(t) = \max|d(t)| + \eta$, $\eta > 0$.

Stability analysis is given as follows:

Let the Lyapunov function be

$$V = \frac{1}{2}s^2,$$

and hence, we have

$$\begin{aligned} \dot{V} &= s\dot{s} = s(c\dot{e} + \ddot{e}) = s(c\dot{e} + \ddot{\theta}_d - \ddot{\theta}) \\ &= s(c\dot{e} + \ddot{\theta}_d + a\dot{\theta} + b\theta - ku(t) - d(t)). \end{aligned} \tag{12}$$

Substitute control law (11) into (12), we can get

$$\begin{aligned} \dot{V} &= s\dot{s} = s(-K(t)\text{sgn}(s) - d(t)) \\ &= -K(t)|s| - sd(t) \leq -\eta|s| \leq 0. \end{aligned} \tag{13}$$

In (11), $K(t)$ is used to compensate the uncertain disturbance $d(t)$ and satisfy the existence condition of sliding mode plane. If $d(t)$ is a time-varying disturbance, $K(t)$ should also be designed as time-varying in order to weaken the chattering phenomenon. Hence, fuzzy logic rules are introduced to realize the estimation of $K(t)$.

3.3 Design of fuzzy logic rules

According to the existing condition of the sliding mode

$$s\dot{s} < 0, \tag{14}$$

$K(t)$ must effectively eliminate the influence of uncertain disturbance $d(t)$ and satisfy (14), which guarantees system state converge to the sliding mode surface. Therefore, $K(t)$ needs to be adjusted adaptively according to the value of $s\dot{s}$, and the basic fuzzy logic is described as:

- If $s\dot{s} < 0$, then $K(t)$ should be reduced;
- If $s\dot{s} > 0$, then $K(t)$ should be increased.

Based on the above idea, fuzzy control rules between input $s\dot{s}$ and output $\Delta K(t)$ are designed. Fuzzy languages to describe $s\dot{s}$ and $\Delta K(t)$ are defined as {negative big, negative medium, negative small, zero, positive small, positive medium, positive big}, that is {NB, NM, NS, ZO, PS, PM, PB}, and their membership functions (MFs) and universe are shown in Figs. 6 and 7, respectively.

According to experts' knowledge and practical experience in [21, 22], seven fuzzy logic rules describing the relationship between $s\dot{s}$ and $\Delta K(t)$ are established as follows:

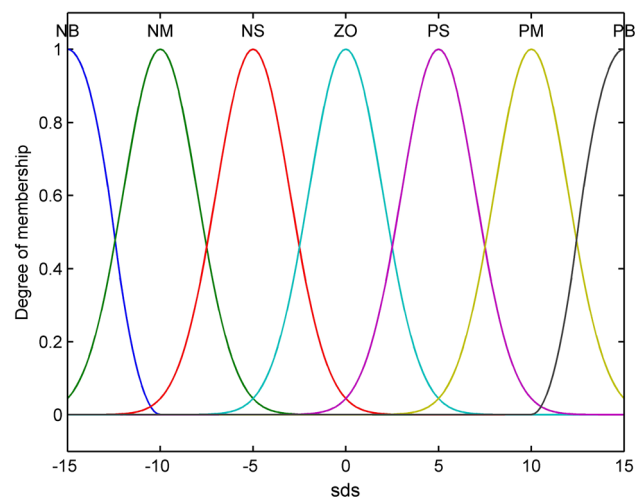


Fig. 6 Membership functions and universes of $s\dot{s}$

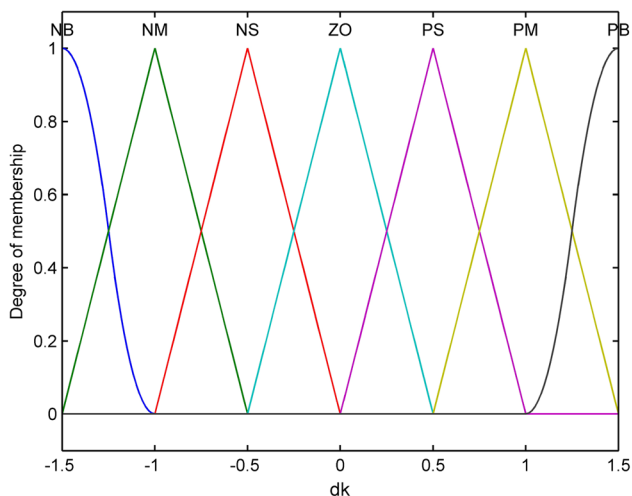


Fig. 7 Membership functions and universes of $\Delta K(t)$

- Rule 1: If ss is PB, then $\Delta K(t)$ is PB;
- Rule 2: If ss is PM, then $\Delta K(t)$ is PM;
- Rule 3: If ss is PS, then $\Delta K(t)$ is PS;
- Rule 4: If ss is ZO, then $\Delta K(t)$ is ZO;
- Rule 5: If ss is NS, then $\Delta K(t)$ is NS;
- Rule 6: If ss is NM, then $\Delta K(t)$ is NM;
- Rule 7: If ss is NB, then $\Delta K(t)$ is NB.

By using the integration method, the top limit of $\hat{K}(t)$ can be estimated as:

$$\hat{K}(t) = G_k \int_0^t \Delta K dt, \tag{15}$$

where G_k is integral coefficient and depends on experts' experience.

Replace $K(t)$ in (11) with $\hat{K}(t)$ in (15); then, the fuzzy sliding mode controller is designed as:

$$u(t) = \frac{1}{k} (a\dot{\theta} + b\theta + \ddot{\theta}_d + c\dot{e} + \hat{K}(t)\text{sgn}(s)). \tag{16}$$

When the queue buffer is full, all arrival packets will be dropped. Otherwise, controller (16) will control the queue length to reach the expected value by dropping packets according to the signal-to-noise ratio $S_{nr}(t)$ ($S_{nr}(t) \in [0, 1]$). As a result, we have:

$$S_{nr}(t) = \begin{cases} 1 & \text{if } S_{nr}(t) > 1, \\ u(t) & \text{if } 0 \leq S_{nr}(t) \leq 1, \\ 0 & \text{if } S_{nr}(t) < 0. \end{cases}$$

The pseudo-code of the proposed FSMC algorithm for WSNs is shown as follows.

Algorithm : Fuzzy Sliding Mode Control Algorithm

```

/*parameters*/
q : instantaneous queue length    q_d : desired queue length
q_m : maximum queue length    c : sliding-mode surface weight
N : load factor    C : link capacity    R_0 : round-trip time
G_k : integral coefficient    T : update period
Enqueue()/*each packet arrival*/
{
update(q);
if (q ≥ q_m)
    drop the packet;
else
    Calculate_SNR()
    {
        x_1 = q - q_d, x_2 = q̇ - q̇_d;
        s = cx_1 + x_2;
        ΔK = Fuzzylogic_table(ss);
        K̂ = G_k ∫_0^t ΔK dt;
        a = 1 / R_0 + 2N / R_0^2 C, b = 2N / R_0^3 C, k = C^2 / 2N;
        SNR = (aq̇ + bq + q̇_d + cx_2 + K̂ sgn(s)) / k;
        if (SNR > 1)
            { SNR = 1; }
        if (SNR < 0)
            { SNR = 0; }
    }
    if (drop_early())
        {
            p = uniform_Random(0,1);
            if (p ≤ SNR)
                {drop or mark the packet;}
            else
                {enqueue the packet;}
            update_rate(1/T)
        }
}

```

4 Simulation results and analysis

In this section, the performance of the proposed FSMC and other traditional methods (RED [23], DropTail [24], fuzzy [10], PID [11] and SMC [15]) is assessed using MATLAB/Simulink [25] and NS-2.35 [26], respectively. During simulation, the control parameters of traditional algorithms are repeatedly adjusted to make sure that their performance is as good as possible.

4.1 Simulation in MATLAB/Simulink

In this subsection, the proper parameters are set as follows: $T_p = 0.05$ s, $T = 0.001$ s, $C = 650$ packets/s, $N = 25$ and $SNR(t) = 18$ dB, that is, (6) is considered as the transfer function of WSNs' congestion model. Therefore, (9) can be written as:

$$\ddot{\theta} = -8.414\dot{\theta} - 15.117\theta + 8450u(t) + d(t), \tag{17}$$

where $d(t) = 300 \exp\left(-\frac{(t-c_i)^2}{2b_i^2}\right)$.

Let $b_i = 0.8$, $c_i = 5.0$, $\eta = 1.0$. Then, $\hat{K}(t) = \max|d(t)| + \eta = 300 + 1 = 301$.

Define the desired queue length signal $\theta_d = \sin(\pi t) + 1$ and the initial queue length $\theta_{init} = 0.75$. Fuzzy sliding mode controller is constructed by combining the sliding mode controller with fuzzy logic controller, as shown in Fig. 8.

When the FSMC (16) with parameters $c = 150$, $a = 8.414$, $b = 15.117$, $k = 8450$ and $G_k = 300$ is implemented for system (17), then the estimation of $d(t)$, control input u and change of reaching law s is shown in Figs. 9, 10 and 11, respectively.

Figure 12 shows the performance comparison of traditional algorithms and the proposed FSMC when the desired queue length is a sine signal. It can be seen that all these methods perform similarly in the initial stage of simulation. However, when uncertain disturbance occurs, traditional approaches show poor anti-jamming performance, while FSMC still performs well. The performance metrics of these control methods when uncertain disturbance occurs are summarized in Table 1.

As can be seen from Table 1, compared with classic fuzzy, PID and SMC approaches, FSMC can reduce

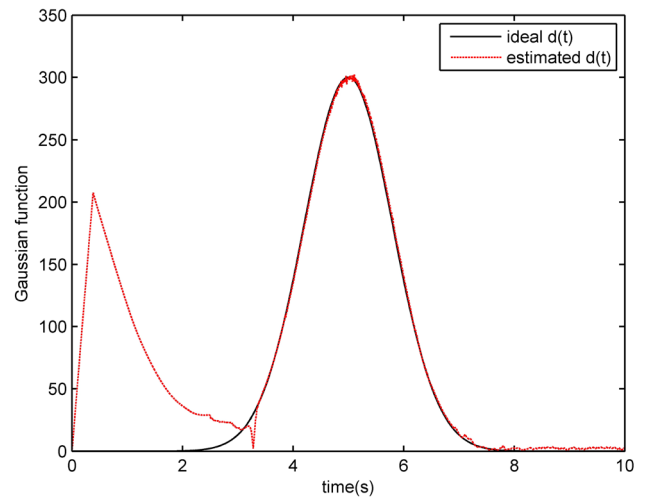


Fig. 9 Estimation of uncertain disturbance $d(t)$

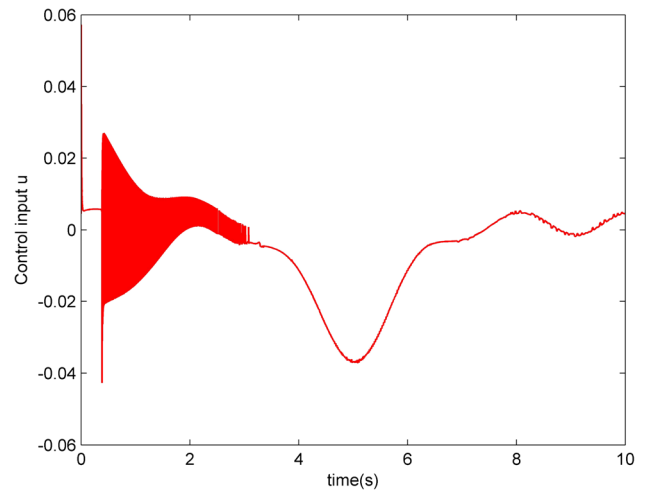


Fig. 10 Control input u of FSMC

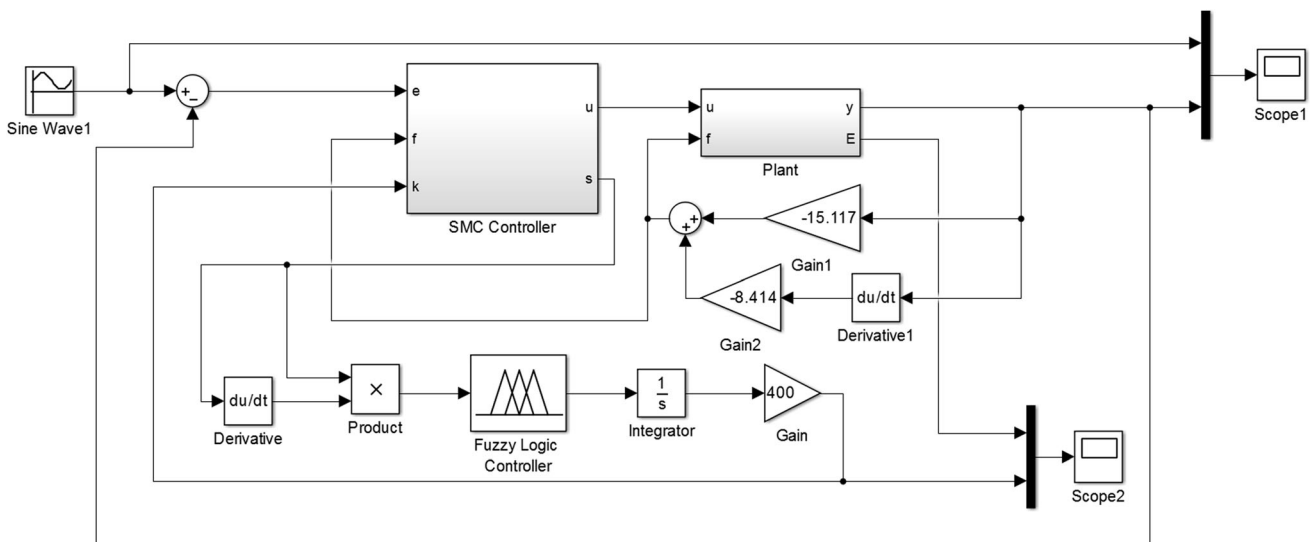


Fig. 8 Detailed structure of FSMC in Simulink

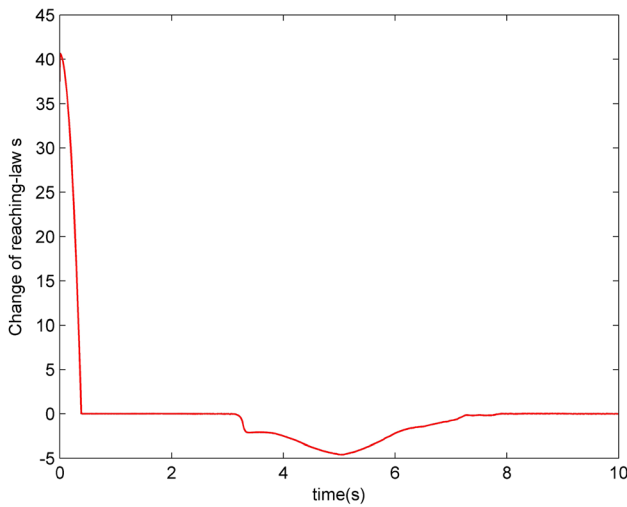


Fig. 11 Change of reaching law s

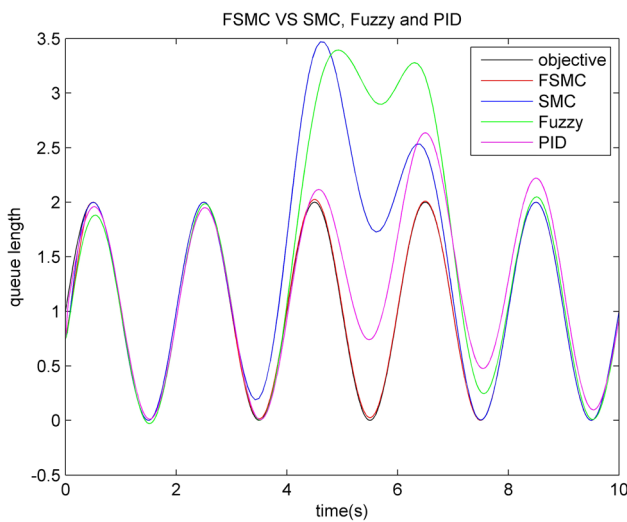


Fig. 12 Performance comparison of traditional algorithms and the proposed FSMC when the desired queue length is a sine signal

overshoot to 1.24%, adjustment time to 2.33 s and maximum error to 0.0248 and has zero steady-state error. Hence, FSMC is more robust to tolerate the influence of disturbance and more adaptive to the dynamical queue length and converges more rapidly, which can effectively avoid and alleviate WSNs’ congestion problem.

Table 1 The performance metrics of four control methods when uncertain disturbance occurs

Control method	Overshoot (%)	Adjustment time (s)	Maximum error	Steady-state error
Fuzzy	69.695	5.5	2.8996	Very small
PID	5.825	5.4	0.7400	Small
SMC	73.455	4.37	1.7259	Zero
FSMC	1.24	2.33	0.0248	Zero

4.2 Simulation in NS-2.35

In order to thoroughly evaluate the characteristics of FSMC in a dedicated WSN simulation tool, NS-2.35 simulation experiments are performed under two different scenarios: $100 \times 100 \text{ m}^2$ and $200 \times 200 \text{ m}^2$. The detailed NS-2.35 simulation parameters are summarized in Table 2.

4.2.1 Case 1: simulation in a $100 \times 100 \text{ m}^2$ monitoring area

In this NS-2.35 simulation, there are 20 nodes randomly deployed in a $100 \times 100 \text{ m}^2$ area, as shown in Fig. 13. Sensor nodes are sending CBR (constant bit rate) packets to sink node (node 9) through intermediate nodes 1, 4, 6, 14 and 15 by using AODV routing protocol and different queuing methods. And the experiments are performed repeatedly by increasing the data rate from 0 to 200 kbps, while keeping all other parameters fixed. The performance metrics of traditional queuing methods and the proposed FSMC mainly include four aspects [27, 28]: average delay, loss ratio, throughput and queue length.

4.2.1.1 Average delay in $100 \times 100 \text{ m}^2$ As shown in Fig. 14, average delay (AD) for traditional methods and FSMC in $100 \times 100 \text{ m}^2$ increases dramatically when the data rate reaches 90 kbps. However, AD of the proposed FSMC increases more humbly than that of other traditional methods. After 130 kbps, AD of these methods all tends to become stable, but that of FSMC is the lowest (around 0.9 s) and reduced by 0.4 s.

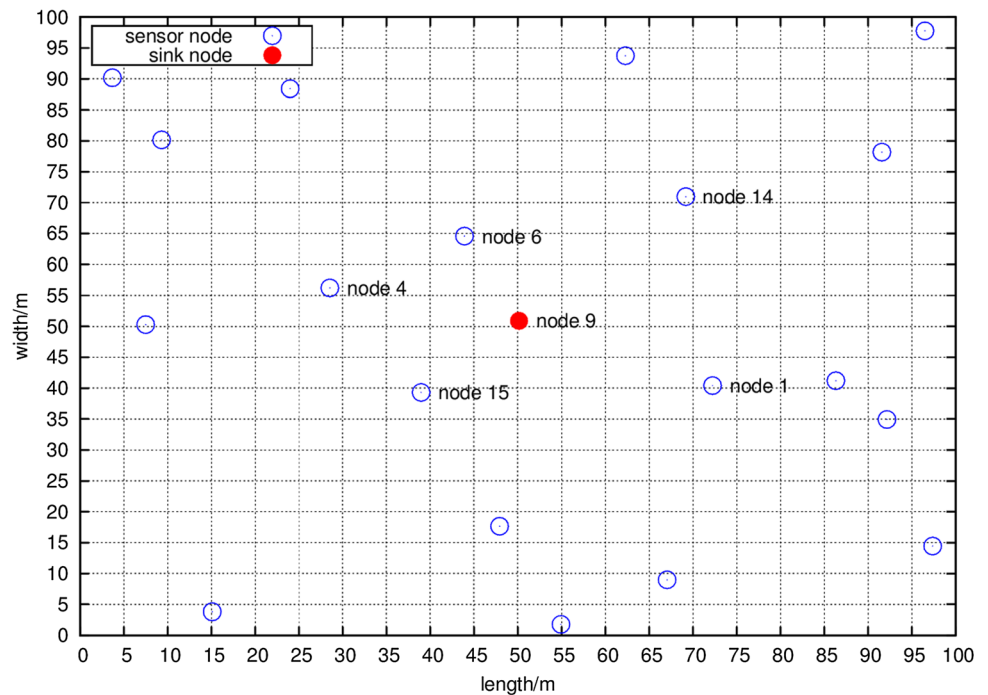
4.2.1.2 Loss ratio in $100 \times 100 \text{ m}^2$ Figure 15 shows the comparison of packet loss ratio for traditional methods and FSMC in $100 \times 100 \text{ m}^2$. It can be seen that they begin to lose packets when the data rate exceeds 90 kbps. Although loss ratio of them raises significantly after 90 kbps, FSMC climbs the most slowly. And when the data rate is 200 kbps, loss ratio of fuzzy control is almost 68%, while that of FSMC is much lower (almost 48%) and reduced by around 20%.

4.2.1.3 Throughput in $100 \times 100 \text{ m}^2$ Figure 16 shows the comparison of throughput for traditional methods and FSMC in $100 \times 100 \text{ m}^2$. Obviously, throughput of these

Table 2 Detailed NS-2.35 simulation parameters

Parameter	Value
Monitoring area	100 × 100 m ² /200 × 200 m ²
Number of nodes	20
Routing protocol	AODV
Queuing method	RED/DropTail/Fuzzy/PID/SMC/DSMC
Intermediate node	Node 1, 4, 6, 14, 15/node 2, 10, 11, 14, 15
Sink node	Node 9/node 3
Mac type	IEEE 802.11
Max. packet in queue buffer	70
Data rate	0–200 kbps
Simulation time	30 s

Fig. 13 Distribution of 20 nodes in a 100 × 100 m² area



six algorithms all increases in positive proportion when the data rate is less than 100 kbps. However, when the data rate exceeds 100 kbps, throughput of them all converges. This is because when the data rate exceeds the maximum packets that a WSN node can transmit, the redundant packets will be dropped, and throughput will no longer increase. Furthermore, FSMC wins the “champion” in the comparison due to its biggest throughput (around 525 kbps and increased by 125 kbps).

4.2.1.4 Queue length in 100 × 100 m² Figure 17 shows the comparison of instantaneous queue length of node 1 using FSMC and traditional approaches when the data rate is 100 kbps and the monitoring area is 100 × 100 m². It can be seen from Fig. 16 that queue length of these algorithms increases gradually over time. However, traditional

control methods can easily fill the queue buffer and result in dropping packets, because queue length of them always fluctuates around the top limit (70 packets). It can also be seen from Fig. 16 that FSMC outperforms traditional methods, because it stays in “full-queue” state for only a short time (during 17–19 s) and its queue length finally stabilizes around 45.

Figure 18 shows the comparison of average queue length of five sensor nodes (nodes 1, 4, 6, 14 and 15) under different queuing methods when the data rate is 100 kbps and the monitoring area is 100 × 100 m². As expected, in whichever node, average queue length of FSMC is relatively shorter than that of traditional approaches, which can effectively avoid packet loss, reduce end-to-end delay and increase the total amount of data.

Fig. 14 Comparison of average delay for FSMC and traditional approaches in $100 \times 100 \text{ m}^2$

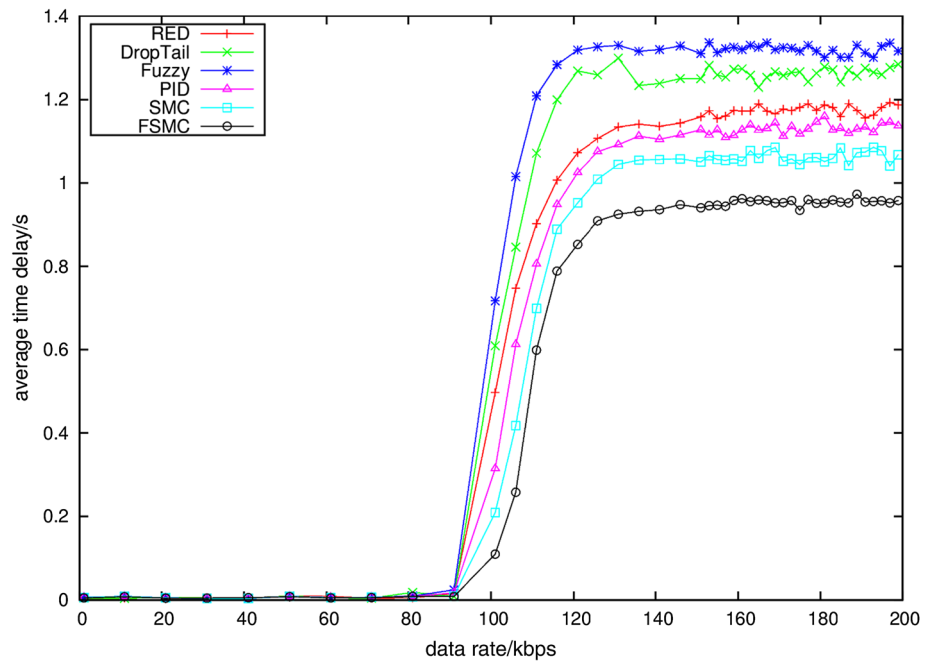
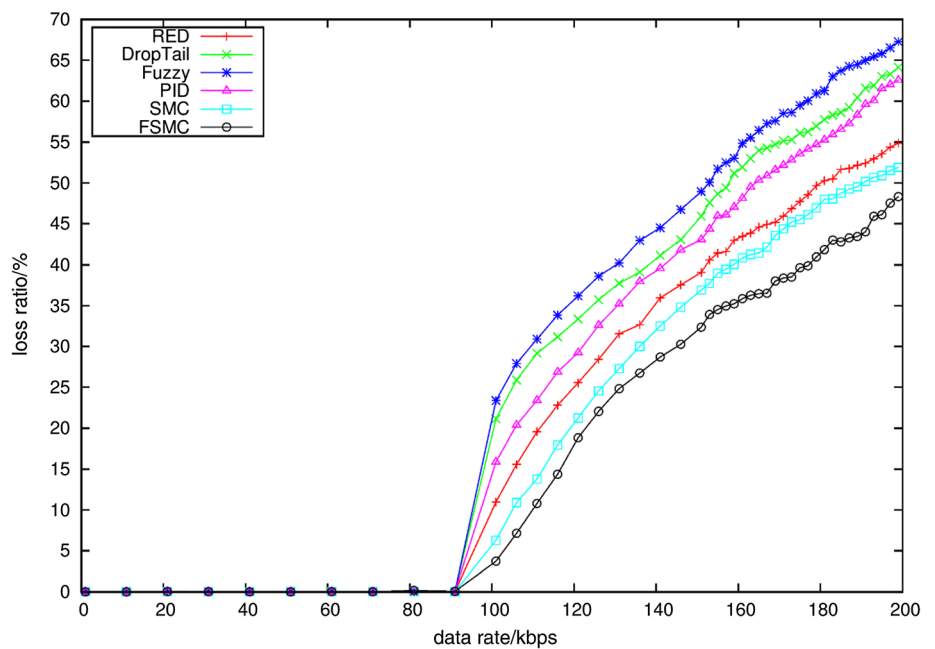


Fig. 15 Comparison of loss ratio for FSMC and traditional approaches in $100 \times 100 \text{ m}^2$



4.2.2 Case 2: simulation in a $200 \times 200 \text{ m}^2$ monitoring area

This subsection aims at evaluating the performance of FSMC in a larger network topology. Hence, similar experiments are performed again by enlarging the area from $100 \times 100 \text{ m}^2$ to $200 \times 200 \text{ m}^2$ (Fig. 19), while maintaining all other network parameters unchanged.

Similarly, after 20 nodes are randomly deployed, they keep sending CBR packets to sink node (node 3) through

intermediate nodes 2, 10, 11, 14 and 15. Meanwhile, the performance metrics also include four aspects: average delay, loss ratio, throughput and queue length.

4.2.2.1 Average delay in $200 \times 200 \text{ m}^2$ Figure 20 shows the comparison of average delay (AD) for FSMC and traditional approaches in $200 \times 200 \text{ m}^2$. Similar to Fig. 13, AD of them increases significantly at 70 kbps and becomes stable at 150 kbps. Although the stabilized AD of FSMC is the lowest (around 1.1 s) among all methods involved in

Fig. 16 Comparison of throughput for FSMC and traditional approaches in $100 \times 100 \text{ m}^2$

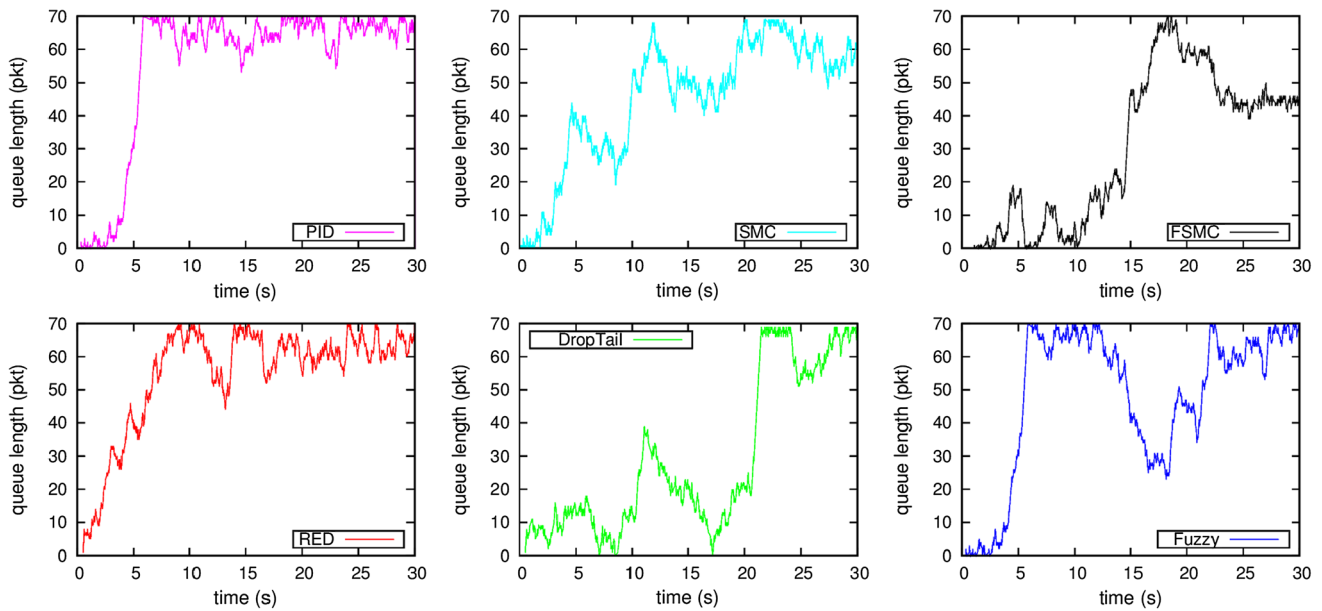
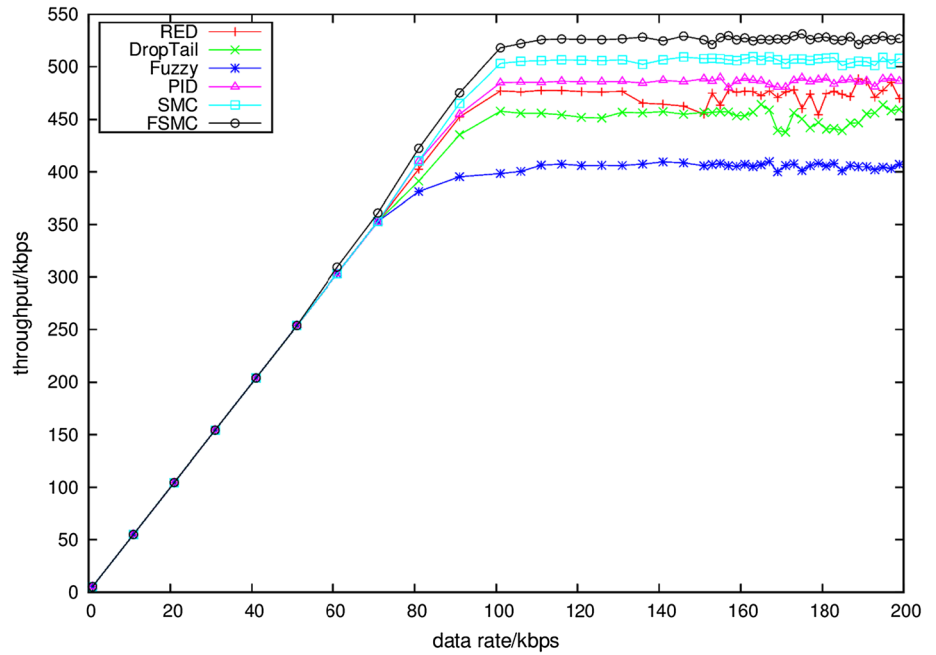


Fig. 17 Comparison of instantaneous queue length of node 1 using FSMC and traditional approaches when the data rate is 100 kbps and the monitoring area is $100 \times 100 \text{ m}^2$

this study, it is still 0.2 s bigger than AD of FSMC in case 1. This is because when region of interest (ROI) becomes broader, the distance from source node to destination node also gets longer. Hence, transmitting the same number of packets will consume more time, which leads to higher delay.

4.2.2.2 Loss ratio in $200 \times 200 \text{ m}^2$ Figure 21 shows the comparison of packet loss ratio for traditional methods and FSMC in $200 \times 200 \text{ m}^2$. Similar to Fig. 14, loss ratio of

these approaches starts to climb at 70 kbps. Compared with traditional approaches, FSMC has the lowest loss ratio (around 54%) at 200 kbps, but it is 6% higher than loss ratio of FSMC in case 1. This is because longer transmission distance will increase the risk of losing packets.

4.2.2.3 Throughput in $200 \times 200 \text{ m}^2$ Figure 22 shows the comparison of throughput for traditional methods and FSMC in $200 \times 200 \text{ m}^2$. Similar to Fig. 15, throughput of these strategies increases proportionally at the beginning

Fig. 18 Comparison of average queue length of five sensor nodes using FSMC and traditional approaches when the data rate is 100 kbps and the monitoring area is $100 \times 100 \text{ m}^2$

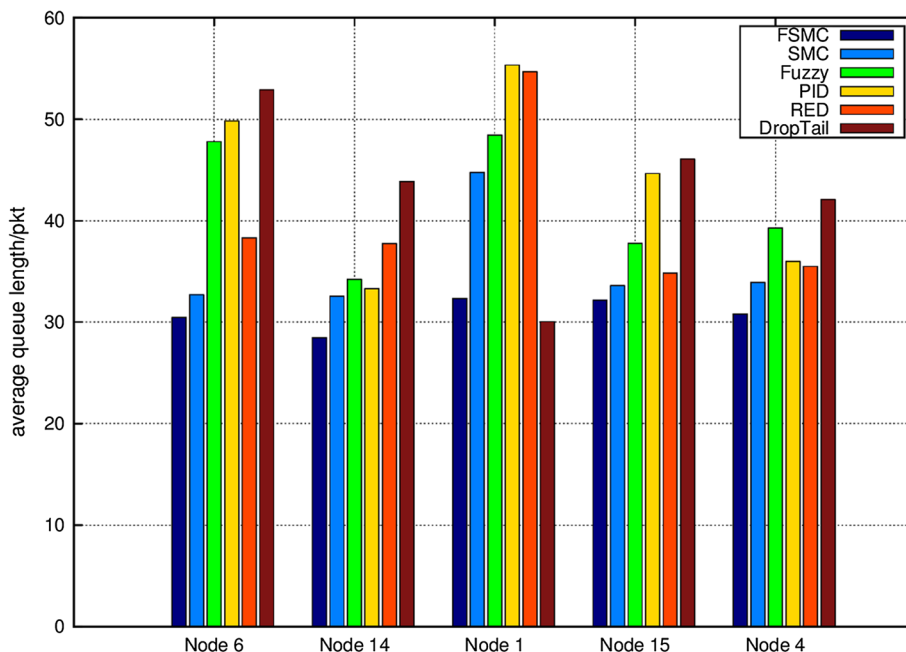
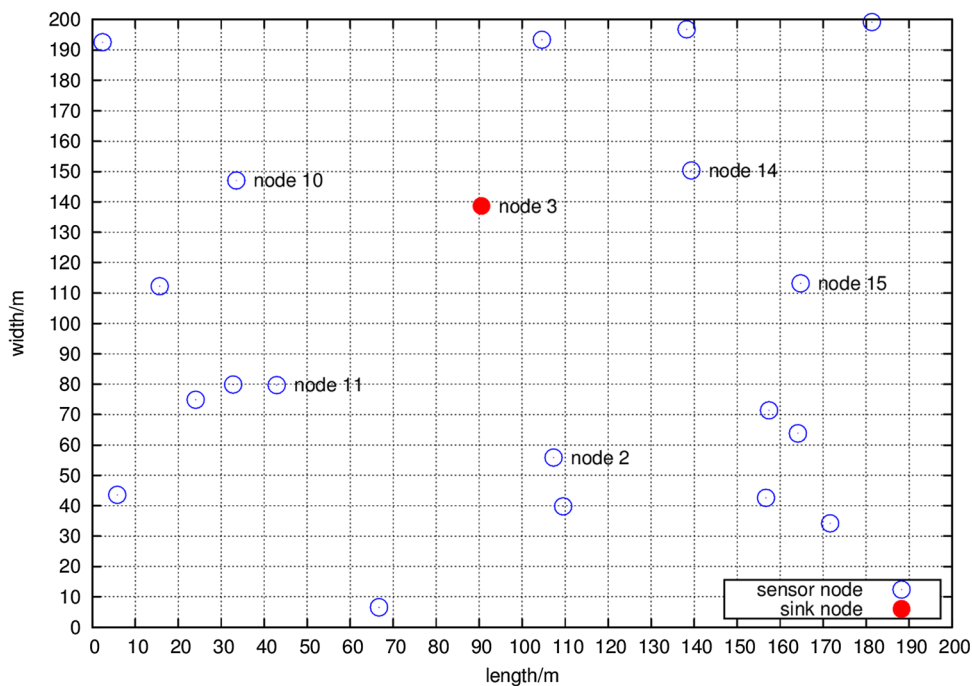


Fig. 19 Distribution of 20 nodes in a $200 \times 200 \text{ m}^2$ area



and stabilizes when the data rate is 80 kbps. Among them, FSMC possesses the largest throughput (around 450 kbps). Nevertheless, compared with the throughput of FSMC in case 1, that of FSMC in this case reduces 75 kbps since ROI is enlarged from $100 \times 100 \text{ m}^2$ to $200 \times 200 \text{ m}^2$.

4.2.2.4 Queue length in $200 \times 200 \text{ m}^2$ Figure 23 shows the comparison of instantaneous queue length of node 14 using FSMC and traditional approaches when the data rate is 100 kbps and the monitoring area is $200 \times 200 \text{ m}^2$.

Different from Fig. 16, the queue length of these methods increases sharply during the first 5 s and then fluctuates around the boundary (70 packets). In contrast, FSMC achieves better performance owing to its temporary “full-queue” state (during 9–15 s) and lower convergent queue length (around 45). However, the stability of FSMC in this case is worse than that of FSMC in case 1 due to its bigger steady-state error.

Figure 24 shows the comparison of average queue length of five sensor nodes (nodes 2, 10, 11, 14 and 15)

Fig. 20 Comparison of average delay for FSMC and traditional approaches in $200 \times 200 \text{ m}^2$

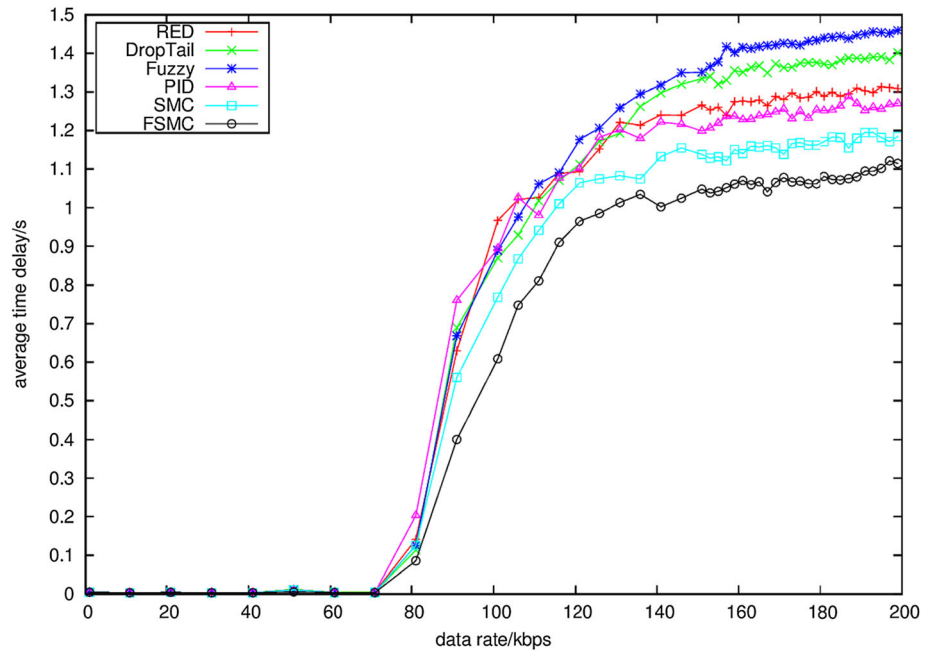
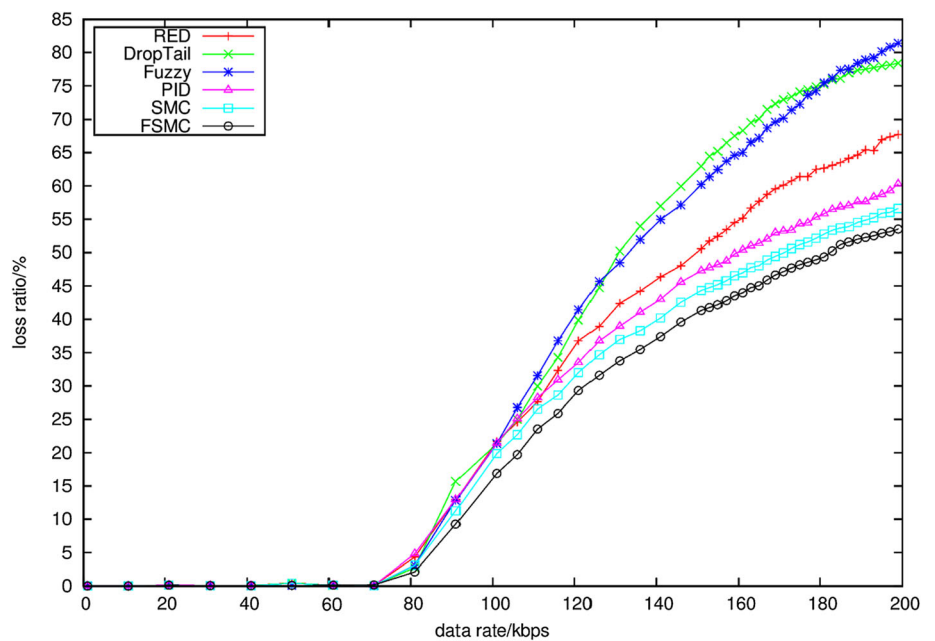


Fig. 21 Comparison of loss ratio for FSMC and traditional approaches in $200 \times 200 \text{ m}^2$



under different queuing methods when the data rate is 100 kbps and the monitoring area is $200 \times 200 \text{ m}^2$. Similar to Fig. 17, whether in nodes 2, 10, 11, 14 or 15, the average queue length of FSMC is smaller than that of other traditional methods, which contributes to reducing loss rate and improving throughput of WSN systems. Please note that longer transmission distance can easily lead to congested wireless channel and increased queue size. As a

result, the average queue length of FSMC in this case is bigger than that of FSMC in case 1.

5 Conclusion

By applying the received SNR of wireless channel to dynamic TCP model, this paper proposes a cross-layer congestion control model between transmission layer and

Fig. 22 Comparison of throughput for FSMC and traditional approaches in $200 \times 200 \text{ m}^2$

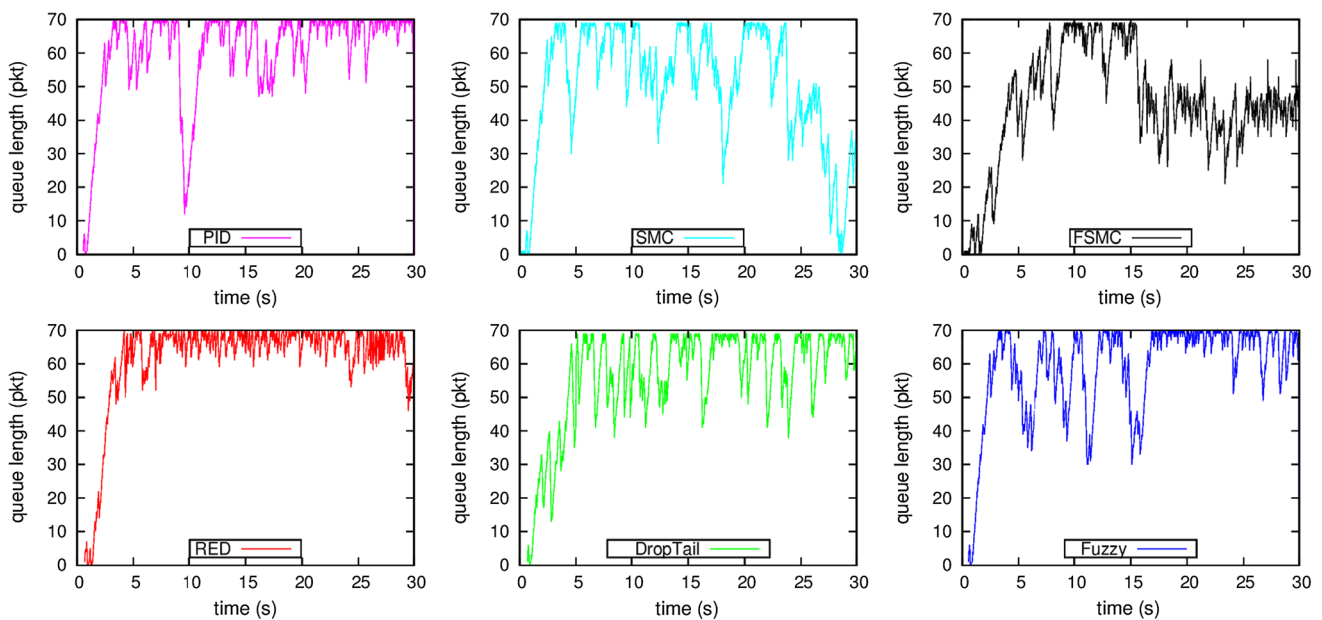
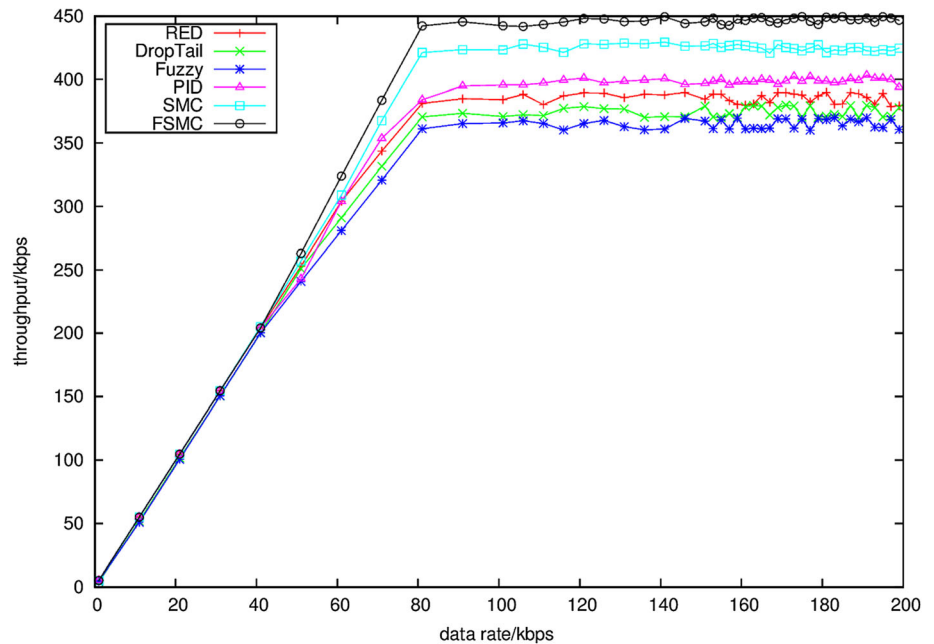
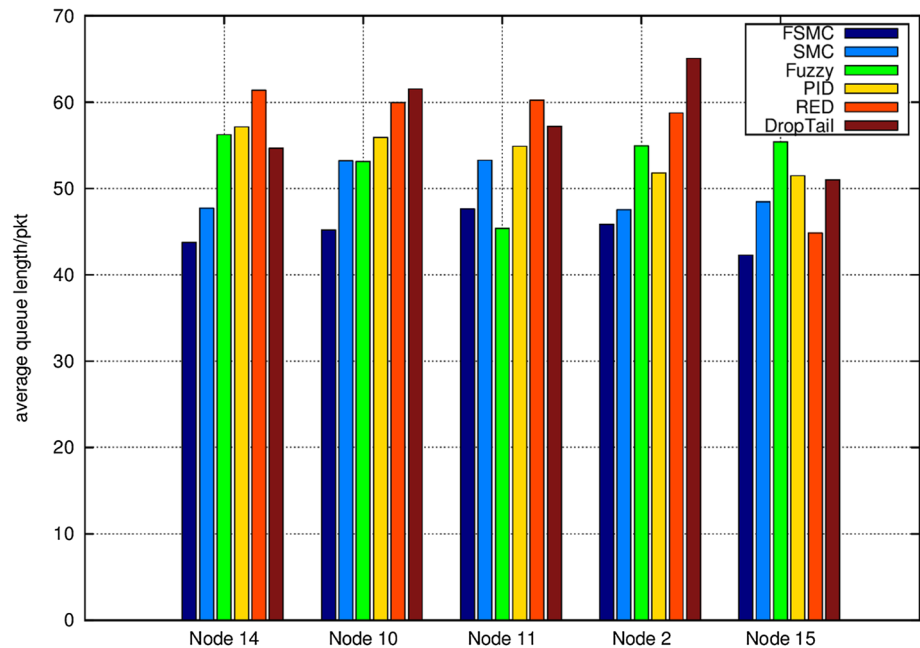


Fig. 23 Comparison of instantaneous queue length of node 14 using FSMC and traditional approaches when the data rate is 100 kbps and the monitoring area is $200 \times 200 \text{ m}^2$

MAC layer as well as a fuzzy sliding mode control algorithm for WSNs. Simulation results show that the proposed algorithm not only adaptively regulates the queue length of buffer in intermediate nodes and effectively avoids the occurrence of congestion, but also significantly reduces the queuing time and increases the speed and stability of convergence and thus improves WSNs' performance in terms of average delay, packet loss ratio and throughput, whether ROI is $100 \times 100 \text{ m}^2$ or $200 \times 200 \text{ m}^2$.

In the future, physical experiments will be performed to verify the effectiveness and prominence of the proposed congestion control scheme in realistic WSN-based IoT applications such as smart home and agriculture automation. And the impact of algorithm parameters on network performance will be taken into account to optimize FSMC, so as to provide a more stable and reliable WSNs' cross-layer congestion control.

Fig. 24 Comparison of average queue length of five sensor nodes using FSMC and traditional approaches when the data rate is 100 kbps and the monitoring area is $200 \times 200 \text{ m}^2$



Acknowledgements This work was financially supported by National Natural Science Foundation of China (Grant Nos. 61673190/F030101), self-determined research funds of CCNU from the colleges' basic research and operation of MOE (CCNU 18TS042), and Graduate Innovation Program of CCNU (2019CXZZ102).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest. The article is considered for publication on the understanding that the article neither has been published nor will be published anywhere else before being published in the journal of *Neural Computing and Applications*.

References

- Mittal N, Singh U, Sohi BS (2019) An energy-aware cluster-based stable protocol for wireless sensor networks. *Neural Comput Appl* 31:7269–7286
- Zhao M, Tian Z, Chow TW (2019) Fault diagnosis on wireless sensor network using the neighborhood kernel density estimation. *Neural Comput Appl* 31:4019–4030
- Binh HTT, Hanh NT, Dey N (2018) Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks. *Neural Comput Appl* 30:2305–2317
- Wu B, Luo J, Yang C (2018) Wireless sensor network minimum beacon set selection algorithm based on tree model. *Neural Comput Appl* 30:965–976
- Sergiou C, Antoniou P, Vassiliou V (2014) A comprehensive survey of congestion control protocols in wireless sensor networks. *IEEE Commun Surv Tutor* 16:1839–1859
- Jan MA, Jan SRU, Alam M, Akhuzada A, Rahman IU (2018) A comprehensive analysis of congestion control protocols in wireless sensor networks. *Mobile Netw Appl* 23:456–468
- Javaid S, Fahim H, Hamid Z, Hussain FB (2018) Traffic-aware congestion control (TACC) for wireless multimedia sensor networks. *Multimed Tools Appl* 77:4433–4452
- Alipio MI, Tiglao NMC (2018) RT-CaCC: a reliable transport with cache-aware congestion control protocol in wireless sensor networks. *IEEE Trans Wirel Commun* 17:4607–4619
- Zhuang Y, Yu L, Shen H, Kolodzey W, Iri N, Caulfield G, He S (2018) Data collection with accuracy-aware congestion control in sensor networks. *IEEE Trans Mob Comput* 18:1068–1082
- Sonmez C, Incel OD, Isik S, Donmez MY, Ersoy C (2014) Fuzzy-based congestion control for wireless multimedia sensor networks. *EURASIP J Wirel Commun Netw* 2014:63
- Yang X, Chen X, Xia R, Qian Z (2018) Wireless sensor network congestion control based on standard particle swarm optimization and single neuron PID. *Sensors* 18:1265
- Kafi MA, Djenouri D, Ben-Othman J, Badache N (2014) Congestion control protocols in wireless sensor networks: a survey. *IEEE Commun Surv Tutor* 16:1369–1390
- Nikokheslat HD, Ghaffari A (2017) Protocol for controlling congestion in wireless sensor networks. *Wirel Pers Commun* 95:3233–3251
- Kafi MA, Ben-Othman J, Ouadjaout A, Bagaa M, Badache N (2017) REFIACC: reliable, efficient, fair and interference-aware congestion control protocol for wireless sensor networks. *Comput Commun* 101:1–11
- Li M-W, Jing Y-W, Chen X-Y (2012) Cross-layer congestion control for wireless sensor network based on sliding mode variable structure. *Control Decis* 27:451–454
- Hollot CV, Misra V, Towsley D, Gong W (2002) Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Trans Autom Control* 47:945–959
- Ma T, Yu T, Huang J, Yang X, Gu Z (2020) Adaptive odd impulsive consensus of multi-agent systems via comparison system method. *Nonlinear Ana Hybrid Syst* 35:100824
- Ma T, Cui B, Wang Y, Liu K (2019) Stochastic synchronization of delayed multiagent networks with intermittent communications: an impulsive framework. *Int J Robust Nonlinear Control* 29:4537–4561
- Fu J, Bai J, Lai J, Li P, Yu M, Lam H-K (2019) Adaptive fuzzy control of a magnetorheological elastomer vibration isolation

- system with time-varying sinusoidal excitations. *J Sound Vib* 456:386–406
20. Fu J, Dai Z, Yang Z, Lai J, Yu M (2019) Time delay analysis and constant time-delay compensation control for MRE vibration control system with multiple-frequency excitation. *Smart Mater Struct* 29:014001
 21. Liu J, Wang X (2012) *Advanced sliding mode control for mechanical systems*. Springer, Berlin
 22. Azar AT, Zhu Q (2015) *Advances and applications in sliding mode control systems*. Springer, Cham
 23. Biddut MJH, Islam N, Arif MFH, Rahman MS (2016) On the analysis of RED algorithm in ZigBee network for queue management. In: 2016 5th international conference on informatics, electronics and vision (ICIEV), pp 408–412
 24. Rastogi S, Zaheer H (2016) Comparative analysis of queuing mechanisms: Droptail, RED and NLRED. *Soc Netw Anal Min* 6:70
 25. Rajaram ML, Kougianos E, Mohanty SP, Choppali U (2016) Wireless sensor network simulation frameworks: a tutorial review: MATLAB/Simulink bests the rest. *IEEE Consum Electron Mag* 5:63–69
 26. Gholipour M, Haghighat A, Meybodi MR (2018) Congestion avoidance in cognitive wireless sensor networks using TOPSIS and response surface methodology. *Telecommun Syst* 67:519–537
 27. Zhao L, Qu S, Yi Y (2018) A modified cluster-head selection algorithm in wireless sensor networks based on LEACH. *EURASIP J Wirel Commun Netw* 2018:287
 28. Singh K, Singh K, Aziz A (2018) Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm. *Comput Netw* 138:90–107

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.