Distributed consensus problem with caching on federated learning framework

Xin Yan¹, Yiming Qin¹, Xiaodong Hu² and Xiaoling Xiao³

Abstract
Federated learning framework facilitates more applications of deep learning algorithms on the existing network architectures, where the model parameters are aggregated in a centralized manner. However, some of federated learning participants are often inaccessible, such as in a power shortage or dormant state. That will force us to explore the possibility that the parameter aggregation is operated in an ad hoc manner, which is based on consensus computing. On the contrary, since caching mechanism is indispensable to any federated learning mobile node, it is necessary to investigate the connection between it and consensus computing. In this article, we first propose a novel federated learning paradigm, which supports an ad hoc operation mode for federated learning participants. Second, a discrete-time dynamic equation and its control law are formulated to satisfy the demands from federated learning framework, with a quantized caching scheme designed to mask the uncertainties from both asynchronous updates and measurement noises. Then, the consensus conditions and the convergence of the consensus protocol are deduced analytically, and a quantized caching strategy to optimize the convergence speed is provided. Our major contribution is to give the basic theories of distributed consensus problem for federated learning framework, and the theoretical results are validated by numerical simulations.

Keywords
Consensus computing, federated learning, ad hoc mode, quantized caching, convergence

Date received: 5 October 2021; accepted: 14 March 2022

Handling Editor: Peio Lopez Iturri

Introduction
Federated learning (FL) is an emerging and promising decentralized machine learning approach that performs a collaborative training of models with the local datasets on various mobile devices, with the local model updates being sent to a server to aggregate and the updated global model being fed back for the next round of local training, instead of transmitting the raw data to a data center. Hence, training models are shared and the privacy of local datasets is preserved, while the communication cost can be reduced greatly.¹ Given these compelling benefits, a rapidly increasing research attention has been dedicated to applying FL in the field of wireless communications to support more intelligent, more convenient and applicable applications,² for example, an image classification task for vehicular edge computing,³ a content popularity

¹School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
²The Faculty of Education, The University of Hong Kong, Hong Kong, China
³School of Computer Science, Yangtze University, Jingzhou, China

Corresponding author:
Xin Yan, School of Computer Science and Artificial Intelligence, Wuhan University of Technology, 122 Luoshi Road, Wuhan 430070, China.
Email: yanxin@whut.edu.cn

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
prediction for augmented reality (AR) applications,\textsuperscript{4} a
signal classification or a deep anomaly detection in industrial distributed wireless sensor networks, etc.\textsuperscript{5,6}

Recent 2 years have begun to witness an increasing interest in studying how to employ FL on mobile ad hoc networks (MANET) or multi-agent systems (MAS), such as unmanned aerial vehicles (UAV),\textsuperscript{7} and vehicular Internet of Things (IoT).\textsuperscript{8} Nevertheless, these mobile devices acting as FL clients are designed to directly communicate with an FL server (or a cluster center that plays a role in relaying), rather than an ad hoc operation mode. This situation is also present when FL is applied to wireless sensor networks.\textsuperscript{5,6} The distributed consensus problem is the theoretical basis for supporting FL parameter synchronization (namely, model updates’ aggregation) in the context of ad hoc operation mode.

The distributed consensus problem (aka consensus computing) means to make the scalar states of a set of nodes (or agents) converge to the same value or the average under local communication constraints. A distributed consensus algorithm or protocol is an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network. The essence of this algorithm is that, in each round, one or more agents can communicate information with its immediate neighbors, and then each agent updates its estimate of a quantity of states by combining the estimate with those of its neighbors.

What are the benefits brought by the ad hoc operation mode for FL framework? (1) Either FL server or cluster center may be inaccessible in some scenarios, such as in a power shortage or device dormant state, (2) when a large number of FL clients send their requests to link an FL server or a cluster center, it may be overwhelmed due to its limited capability, (3) this operation mode enables short-range communications such as ultra-wideband by which a large channel capacity can be obtained for local model updates’ transmissions that are capacity-consuming, (4) in this mode, FL parameter synchronization is enabled by mobile devices in a coordinated manner in the absence of FL server or cluster center, and (5) the asynchronous transmissions of local model updates are allowed to a considerable extent in this operation mode.

In view of the above reasons, a novel FL paradigm is proposed at the beginning of this article, as shown in Figure 1. It is obvious that the most salient challenge for this paradigm should be the communication overhead for parameter synchronization. Thanks to the sparsification technology,\textsuperscript{9} combining with quantization, compression, or selective communication technologies,\textsuperscript{2} the size of model or gradient updates can be reduced tremendously. It can be seen that our scheme is sufficiently practicable if ultra-wideband communication is employed in this paradigm as well.

In a word, to the best of our knowledge, as of now the study of the distributed consensus problem over FL framework has not been found yet. That is our motivation of exploring this issue.

**Related work**

As mentioned above, until now we have not seen an FL framework whose mobile clients operate in an ad hoc manner yet. The existing studies of FL over MANET (or MAS) are concentrated on communication cost reduction, FL client selection, data privacy and security, so on.

Zhang and Hanzo\textsuperscript{7} proposed an FL-aided multi-UAV system to conduct classification tasks for exploration scenarios, where each of UAV is coordinated by a ground fusion center as FL server to form a cooperative network. An algorithm of weighted zero-forcing precoding is used by each of UAV to mitigate the interference to the FL server. Bao et al.\textsuperscript{8} proposed an edge computing-based joint client selection and networking scheme for vehicular IoT, where some of vehicles are assigned to act as both edge nodes (aka cluster centers) and FL clients via a distributed approach. The selected clients play a role of forworders between common vehicles and FL server. Lu et al.\textsuperscript{10} employed an FL architecture empowered by blockchain to address data privacy concerns on Internet of Vehicles, where the security of shared data is guaranteed by integrating learning
parameters into a blockchain. Regarding the aforementioned sparsification technology, Sun et al.\textsuperscript{9} presented a general gradient sparsification framework as another way to reduce communication cost for FL parameter synchronization on IoT, where validation data sets are maintained with top-1 accuracy when 99.9\% gradients are sparsified. As for the FL applications in distributed sensor networks, Liu et al.\textsuperscript{5} used an FL paradigm to fuse the learning process and recognition results of each sensor node for the modulation recognition of wireless signals. Liu et al.\textsuperscript{5} proposed an on-device FL-based deep anomaly detection framework for sensing time-series data. Both of the FL frameworks require parameter aggregators as FL servers, instead of an ad hoc operation mode.

To date, the distributed consensus problem of perfect models, which are assumed that each agent (or node) can obtain its neighbor information timely and precisely, instantaneous transmission, perfect clock synchronization, concurrent updates, identical agent dynamics, even fixed network topologies, has reached a reasonable degree of maturity.\textsuperscript{11} Nonetheless, wireless networked systems in practical applications often operate in uncertain communication environments and are inevitably subjected to communication latency, asynchronous clock and updates, agents’ heterogeneity, topological dynamics, as well as measurement noises (including additive and multiplicative noises). Then what are the specific demands on consensus computing in terms of FL framework over MANET (or MAS)? (1) Due to the intermittent and unreliable communications arising from topological dynamics and channel noise, measurement noises that may be treated as time delay (i.e. additive noise) and packet loss (i.e. multiplicative noise) should be taken into account. Furthermore, time delay on each link (including channel and queuing delay) is asked to be non-identical and time-varying. Packet loss rate on each link (including channel and queuing packet loss) may be non-identical and time-varying as well, (2) given that some of nodes are sometimes inaccessible (e.g. in dormant state), FL parameter aggregation must be achieved in an asynchronous manner, (3) in consideration of nodes’ heterogeneity, caching on each node ought to be non-identical, and (4) consensus computing is requested to make good use of the broadcast nature of wireless communications, in that its convergence can be accelerated tremendously.\textsuperscript{12}

Although the existing distributed consensus algorithms explore some of four aspects mentioned above, to the best of our knowledge, an algorithm that can fulfill all of requests has not been seen yet. Moreover, the caching issue along with consensus computing has not been received any concerns up to now. Olfati-Saber and Murray\textsuperscript{13} provided the consensus protocols and their convergence analysis for directed balanced networks with constant time-delays by introducing disagreement functions, while a direct connection between the algebraic connectivity of a graph and the convergence of a linear consensus protocol is established. Savino et al.\textsuperscript{14} contributed a sufficient condition of consensus for discrete-time switching networks, based on linear matrix inequalities that consider the joint effect of time-varying delays and topological uncertainty. Under an assumption that delay is time-varying and undirected network is connected, Wang et al.\textsuperscript{15} derived the conditions to guarantee consensus for continuous-time multi-agent systems. For the consensus problem of a switched multi-agent system composed of continuous-time and discrete-time subsystems, Zheng and Wang\textsuperscript{16} proposed a linear consensus protocol and proved that this consensus problem is solvable under arbitrary switching with undirected connected graph, directed graph, and switching topologies, respectively. Kar and Moura\textsuperscript{17} studies the distributed average consensus with intermittent topologies and noisy channels in sensor networks, which leads to a bias-variance dilemma, that is, running consensus for long reduces the bias of the final average estimate but increases its variance, and presented two versions of consensus compromise to this tradeoff. Zong et al.\textsuperscript{18} investigated the stochastic consensus conditions of linear MAS with fixed time-delays and stochastic multiplicative noises. First, the stochastic stability for stochastic differential delay equations driven by multiplicative noises is examined. Then, sufficient conditions are deduced for the mean-square and a.s. consensus. Zheng et al.\textsuperscript{19} studied the mean-square consensus problem of discrete-time linear MAS over directed networks with constant delay and non-identical packet dropouts. Sufficient consensus conditions are obtained in terms of delay, packet dropout rates, network topology and agent dynamics. On the basis of a first-order average-consensus protocol with switching networks and additive noises, Chen et al.\textsuperscript{20} gave a quantitative description of relation between convergence speed and connectivity of topologies by using stochastic approximation methods and establishing a critical consensus condition for network topologies.

In short, inspired by the fact that caching plays a critical role in FL operations\textsuperscript{21,22} we will investigate the connection between caching and consensus computing, while discussing the condition for reaching consensus.

### Problem formulation

#### Algebraic graph theory

A MANET (or MAS) is described as a sequence of weighted digraphs $G(k) = \{V, E(k)\}$, with graph index $k \in \{1, 2, \ldots, \infty\}$, where $V$ is the set of $N$ nodes (or agents) and $E(k)$ is the set of all existing wireless links.
Let $A(k) = [a_{ij}] \in \mathbb{R}^{N \times N}$ be the adjacency matrix of $G(k)$, with elements below

$$a_{ij} = \begin{cases} w_{ij} & (i,j) \in E(k) \\ 0 & (i,j) \notin E(k) \end{cases}$$  \quad (1)$$

where $w_{ij}$ is the weight of link $(i,j)$.

Suppose that the probability that there exists a link $(i,j)$ is $p_{ij}$ and the packet loss rate over the entire network is $p$. Let $L(k) = [l_{ij}] \in \mathbb{R}^{N \times N}$ be the Laplacian matrix of $G(k)$, with elements below

$$l_{ij} = \begin{cases} -p_{ij}v_i(k), & i \neq j \\ \sum_{j \in V_i} p_{ij}v_i(k), & i = j \end{cases}$$  \quad (2)$$

where $p_{ij}$ indicates whether there exists a link $(i,j)$, $v_i(k)$ indicates whether a packet loss occurs over the link $(i,j)$, and both $P_{ij}(k)$ and $v_i(k)$ follow a Bernoulli process. As a result, $Pr(P_{ij}(k) = 1) = p_{ij}$, $Pr(P_{ij}(k) = 0) = 1 - p_{ij}$, $Pr(v_i(k) = 0) = p$, and $Pr(v_i(k) = 1) = 1 - p$.

In addition, let $\lambda_i$ denote the $i$th eigenvalue of the Laplacian in average $L(k)$.

**Distributed consensus problem**

Each node (or agent) in a networked system can be described by a discrete-time dynamic equation, that is

$$x_i(k + 1) = Ax_i(k) + Bu_i(k), i = 1, 2, \ldots, N$$  \quad (3)$$

where $x_i \in \mathbb{R}^m$ and $u_i \in \mathbb{R}^m$ are the state and input of node $i$, respectively. Both $A$ and $B$ are node $i$’s coefficient matrices, which are employed to characterize this node.

In view of the results, we can assume that all eigenvalues of $A$ are either on or outside a unit circle, $[\{A|B\}$ is controllable, and the union of a sequence of weighted directed graphs, that is, $G = \cup G(k)$, contains a directed spanning tree.

Considering both delay and packet loss, $u_i$ can be expressed as

$$u_i = K \sum_{j \in \mathbb{N}_i} P_{ij}(k - \tau)v_j(k - \tau)(x_j(k - \tau) - x_i(k - \tau))$$  \quad (4)$$

where $K$ is the control gain, $\tau$ is the delay denoted by an integer, and $\mathbb{N}_i$ is the neighbor set of node $i$. Therefore, we can formulate the update equation as

$$x_i(k + 1) = Ax_i(k) + BK \sum_{j \in \mathbb{N}_i} P_{ij}(k - \tau)v_j(k - \tau)$$

$$(x_j(k - \tau) - x_i(k - \tau))$$  \quad (5)$$

It is said this networked system can reach a mean-square consensus if there exists a control gain $K$ such that

$$\lim_{k \to \infty} E(||x_i(k) - x_j(k)||^2) = 0, \forall i,j \in \{1, \ldots, N\}$$  \quad (6)$$

where $\|\cdot\|$ represents the Euclidean norm of the vector.

**Consensus protocol**

In this section, we will derive a criterion to evaluate the consensus of the networked system and present the consensus protocol.

**Consensus conditions**

For a networked system with delay and packet loss, the update equation is

$$x(k + 1) = Ax(k) + v(k)BKx(k - \tau)$$  \quad (7)$$

**Lemma 1.** For equation (7), as long as there exists a positive-definite matrix $P$ satisfying equation (8), then we can say that the system is mean-square consensus, that is

$$\begin{align*}
(\mu BK + A - I)^TP + (\tau + 1)(\mu BK + A - I)^T
P(\mu BK + A - I) + P(\mu BK + A - I)^T
\tau(\mu BK)^TP(\mu BK) + \sigma^2 K^TB^TPB < 0
\end{align*}$$  \quad (8)$$

where $\mu = E(\varepsilon(k)) = 1 - p$, $\sigma^2$ is the variance of $\varepsilon(k)$ and $\sigma^2 = p(1 - p)$. The proof of this lemma is given in Appendix 1.

We employ the control gain $K$ defined as

$$K : \frac{\omega^*}{2\tau(1-p) + 1} (B^TQB)^{-1}B^TQ[(\tau + 1)A - \tau \cdot I]$$  \quad (9)$$

$$\omega^* := \arg \min_{\omega \in \mathbb{R}} \max_{i = 2, \ldots, N} |1 - \omega \lambda_i|^2$$  \quad (10)$$

$Q$ in equation (9) is a positive-definite matrix that meets the Riccati inequality

$$Q > \tilde{A}^TQ\tilde{A} - \gamma\tilde{A}^TQB(B^TQB)^{-1}B^TQ\tilde{A}$$  \quad (11)$$

where $\tilde{A} = (\tau + 1)A - \tau \cdot I$ and $\gamma = (\tau + 1)(1 - p)/(2\tau(1-p) + 1)(1 - \omega^*)$.

The consensus error is defined as

$$\delta_i(k) = x_i(k) - \sum_{j=1}^N r_jx_j(k)$$  \quad (12)$$
where \( r_j \) is the element of vector \( \mathbf{r} = [r_1, r_2, ..., r_N]^T \) enabling \( \mathbf{r}^T \cdot \mathbf{L}(k-\tau) = 0 \) and \( \mathbf{r}^T \cdot \mathbf{1}_N = 1 \).

Since \( G \) has a directed spanning tree, there exist matrices \( Y \) and \( S \) such that \( \psi = [I_N \, Y], \psi \psi^T = [r^T S]^T \), and \( \tilde{\delta}(k) = (\psi^{-1} \otimes I_N)E[\delta(k)] \), then

\[
\tilde{\delta}_i(k + 1) = \tilde{\delta}_i(k) - \varepsilon(k) \tilde{A}_i \tilde{B}_i \tilde{\delta}_i(k - \tau), i = 1, 2, ..., N \tag{13}
\]

We can see that equation (6) is also equal

\[
\lim_{k \to \infty} \mathbb{E}[(|\tilde{\delta}_i(k)|^2)] = \mathbb{E}[(|\tilde{\delta}_i(k)|^2)] = 0.19
\]

Thus, if this equation always holds for \( \forall i \in \{1, 2, ..., N\} \), the consensus can be reached in a mean-square manner.

To get the further study of the topological consensus conditions, we can define Lemma 2. It is difficult to form a balanced digraph (or a balanced joint digraph) for broadcast-based networks. That is why a bias of convergence result from its accurate value occurs sometimes, which is questioned.24

**Lemma 2.** If the network is a balanced digraph, the final convergence value of it is the mean value of its initial value.

The proof is given in Appendix 2.

**Quantized caching**

Due to the delay occurring when sending messages between nodes, the asynchronous problem is inevitable. Hence, it is necessary to consider the asynchronous problem of communications. We use a quantized caching mechanism to mask the uncertainties from asynchronous updates and varying delays. As shown in Figure 2, node \( i \) receives messages with different delays (from \( t - \tau \Delta t \) to \( t \)) and caches them, respectively. Then, node \( i \) deals with them at the moment \( t \). The messages are sent by the neighbor node \( j \) at the moment \( t - \tau \Delta t \), which may arrive at \( t - (\tau - 1)\Delta t \) or \( t - (\tau - 2)\Delta t \) due to the delay. The quantized caching mechanism used caches these messages and deals with them at moment \( t \) after they arrived.

The algorithm given in Table 1 is used to simulate the consensus process of the networked system.

We give the complexity of message overhead of this consensus algorithm prior to its convergence analysis. Assume that the number of messages sent by all nodes over an entire network is \( d \) in a certain period of time \( \tau_{\text{max}} \) that denotes the allowed maximum (without packet loss) of time delay \( \tau \). Thus, the message overhead in a round of communication is \( \tau(d/\tau_{\text{max}}) \) over the entire network. When considering the number of iterated rounds \( k \), the complexity of message overhead should be \( k\tau(d/\tau_{\text{max}}) \).

![Figure 2. Illustration of quantized caching on the FL participants.](image)

**Convergence analysis**

In this section, we will discuss the impact of time-delay and packet loss on the convergence speed of the network.

**Convergence speed**

Inequality equation (66) in Appendix 1 indicates that the convergence speed is determined by \( \gamma^2 \). The larger the value of \( \gamma \) is, the faster the algorithm converges. Now, let \( s^* := e^T, s^* \in (1, s_0) \). If \( s_0 \) becomes larger, \( s^* \) should also become larger. Suppose that \( s_0 \) is the solution of \( H(s) = 0 \) when \( s > 1 \), as shown in Appendix 1. Therefore, the convergence speed of the algorithm turns into the range of zero point values of function \( H(s) \) when \( s > 1 \). The larger the zero point value of function \( H(s) \) is, the faster the algorithm converges.

Based on equation (13), we have

\[
E[P_1] = (\mu \lambda, BK)P[-2 + (2\mu + 1)\lambda, BK], \tag{14}
\]

where \( \mu = 1 - p \).

According to equation (63) in Appendix 1, we get

\[
H(s) = C_2(s)(1 - s^{-1}) - \lambda_{\text{min}}(-E[P_1]) \tag{15}
\]

where

\[
C_2(s) = 2\|P\| + 3\tau^2\|P\| \cdot E[\|eBK\|]^2 \cdot s^7 \approx 0 \tag{16}
\]

Lemma 1 indicates that it is necessary to satisfy \( E[P_1] < 0 \) to ensure the mean-square consensus of the networked system. By introducing the control gain \( K \), we get

\[
BK = \frac{\omega^*}{2\tau(1 - p) + 1} \cdot I \tag{17}
\]

and

\[
E[P_1] = (\mu \hat{\lambda}, BK)P(\hat{\lambda}, \omega^* - 2) \tag{18}
\]

Since \( \omega^* = 2/(\hat{\lambda}_2 + \hat{\lambda}_{\text{max}}) \) and \( \hat{\lambda}, \omega^* = 2\hat{\lambda}_1/(\hat{\lambda}_2 + \hat{\lambda}_{\text{max}}) < 2,19 \) as long as \( P \) is a positive-definite matrix, \( E[P_1] < 0 \) holds. We also assume that \( A = 1, B = b \in \mathbb{R}^{1 \times 1}, K = k \in \mathbb{R}^{1 \times 1}, \) and all of the digraphs are balanced.
Table 1. Consensus algorithm.

| Input | Initial states $x_1(0), x_2(0), ..., x_N(0)$ |
| Output | Alignment value $x$ |

1: Generate the initial state of $N$ nodes $x_1(0), x_2(0), ..., x_N(0)$
2: $N$ nodes start their broadcasting
3: For $i = 1$ to $k$
4: Node $i$ receives messages from its neighbors
5: Node $i$ updates its state after $\tau$ based on $x_i(k + 1) = Ax_i(k) + Bu_i(k)$
6: Node $i$ broadcasts its new state
7: If $x_i(k) = x_j(k)$ for $\forall i, j$, the alignment is reached at round $k$

When each node state is a one-dimensional vector, $H(s)$ may be expressed as

$$H(s) = s^{-1} \{(2 + g)s + f \cdot s^2(s - 1) - 2\}$$  \hspace{1cm} (19)

where

$$g = \frac{1 - p}{2\tau(1 - p) + 1} \left(\tilde{\lambda}, \omega^*\right)$$
$$f = 3\tau^2 \frac{1 - p}{2\tau(1 - p) + 1} \tilde{\lambda}, \omega^*$$  \hspace{1cm} (20)

Let $h(s) := (2 + g)s + f \cdot s^2(s - 1) - 2$  \hspace{1cm} (21)

The zero point of $H(s)$ is also the zero point of $h(s)$ when $s > 1$. Since we have already proved that $H(s)$ has a unique zero in the region of $s > 1$, $h(s)$ also has a unique zero.

Combining these with the analysis of $H(s)$, we can conclude that the larger the zero point of $h(s)$ in the region of $s > 1$ is, the faster the algorithm converges.

**Impact of time delay and packet loss rate**

From equation (21), we know that $h(s)$ is related to packet loss rate $p$ and delay $\tau$. In this subsection, we will analyze the effects of these parameters on the convergence speed of the networked system.

We first investigate the impact of packet loss rate.

Let $\omega := \tilde{\lambda}, \omega^*$ and $y := s$. We know that the zero point of $h(y)$ is an implicit function based on $p, \tau$ and $\omega$.

To study the impact of packet loss rate on the convergence speed, let $\tau$ and $\omega$ be constants. The relationship between $y$ and $p$ can be analyzed based on the existence theorem of the implicit function, thus

$$y'(p) = -\frac{h_p(p, y)}{h_y(p, y)}$$  \hspace{1cm} (22)

where

$$h_p(p, y) = 2 + g + f \left(\tau \cdot y^{-1}(y - 1) + y'\right)$$  \hspace{1cm} (23)

Since $\omega \in (0, 2)$, $\omega(\omega - 2) \in (-1, 0)$ and $(1 - p)/\left(2\tau(1 - p) + 1\right) \in (0, 1 - p)$, there exists $g \in (p - 1, 0]$ and $2 + g > 0$.

Besides, $f > 0$, $y > 1$ and $f \left[\tau \cdot y^{-1}(y - 1) + y'\right] > 0$, which gives $h_p(p, y) > 0$.

We can get the partial derivative of $h(p, y)$ w.r.t. the packet loss rate as

$$h_p(p, y) = \left[y\omega(\omega - 2) + 3\tau^2\omega y^{-1}(y - 1)\right] \left(-\frac{1}{2\tau(1 - p) + 1}\right)^2.$$  \hspace{1cm} (24)

We also have

$$h(p, y) = \left[y\omega(\omega - 2) + 3\tau^2\omega y^{-1}(y - 1)\right] \left(\frac{1 - p}{2\tau(1 - p) + 1}\right) + 2y - 2$$  \hspace{1cm} (25)

$$= 0.$$  \hspace{1cm} (26)

Since $y > 0$,

$$y\omega(\omega - 2) + 3\tau^2\omega y^{-1}(y - 1) < 0$$  \hspace{1cm} (27)

which means that $h_p(p, y) > 0$ holds.

Considering these, we can prove that $y'(p) < 0$, that is, the zero point of $h(y)$ decreases with the increase of packet loss rate.

The impact of delay analysis is the same as the packet loss rate. To do that, we first need to analyze the positive and negative of $y'(p)$ as

$$y'(p) = -\frac{h_\tau(\tau, y)}{h_y(\tau, y)}$$  \hspace{1cm} (28)

where the partial derivative of $h(\tau, y)$ w.r.t. $\tau$ is

$$h_\tau(\tau, y) = \frac{y}{ \frac{1}{\tau} + y} \ln y + \frac{d\tau}{d\tau}(y - 1)y^{-1}$$  \hspace{1cm} (29)

and based on equation (20), we can get

$$\frac{dg}{d\tau} = \omega(\omega - 2) \left(\frac{-2(1 - p)^2}{2\tau(1 - p) + 1}\right) > 0$$
$$\frac{df}{d\tau} = \frac{2\tau^2(1 - p)^2 + 6\tau(1 - p) - 1}{2\tau(1 - p) + 1} > 0$$

As a result, we have $h_\tau(\tau, y) > 0$ and $y'(p) < 0$. To sum up, a remark is concluded as follows.

**Lemma 3.** The convergence speed of the consensus algorithm decreases with the increase of time delay or packet loss rate.

**Convergence optimization**

Assume that the maximum time delay is $\tau_{\text{max}}$, the data arrival rate in each round of communication follows a
Poisson distribution, and the neighbor nodes send $d$ messages on average at a certain time. Therefore, the average number of messages received in a round of communication is $d/\tau_{\text{max}}$. Then, the parameters of Poisson distribution is $\lambda = (d/\tau_{\text{max}})$. When the cache is set to $\tau$, there exists a probability distribution, which is

$$P[N(\tau) = n] = \frac{(\lambda \tau)^n e^{-\lambda \tau}}{n!},$$

and the expected number of received messages is $\tau(d/\tau_{\text{max}})$. Thus, the packet loss rate caused by queuing or caching is

$$p_{qu} = \frac{1}{d} \left( d - \tau \frac{d}{\tau_{\text{max}}} \right) = 1 - \frac{\tau}{\tau_{\text{max}}}$$

and the total packet loss rate is

$$\rho = 1 - (1 - p_{ch})(1 - p_{qu})$$

$$= 1 - (1 - p_{ch}) \frac{\tau}{\tau_{\text{max}}}$$

$$\tau_{\text{max}} = \frac{1}{2} \left( p_{ch} + \sqrt{p_{ch}^2 + \frac{4a}{\tau_{\text{max}}}} \right)$$

where $a = 24\omega + 2\omega^2(\omega - 2) + 2\omega\sqrt{384\omega^3} + 2304\omega^6 + 5328\omega^9 + 6081\omega^{12} + 3292\omega^{15} - 548\omega^2 + 464\omega + 144$.

Introducing $p$ into $h(y)$, we can get

$$h(y) = (2 + g)y + f \cdot y^2(y - 1) - 2$$

where

$$g = \frac{\frac{1}{\tau_{\text{max}}}(1 - p_{ch})}{\frac{\tau}{\tau_{\text{max}}}(1 - p_{ch}) + 1} \omega(\omega - 2)$$

$$f = \frac{\frac{1}{\tau_{\text{max}}}(1 - p_{ch})}{\frac{\tau}{\tau_{\text{max}}}(1 - p_{ch}) + 1}$$

It is necessary to analyze the positive and negative of $y'(\tau)$ according to the existence theorem of the implicit function as

$$y'(\tau) = -\frac{h_y(\tau, y)}{h_x(\tau, y)}$$

By calculating $h_x(\tau, y)$ and $h_y(\tau, y)$, respectively, we can get

$$h_x(\tau, y) = 2 + g + f \left[ By \tau^{-1}(y - 1) + y^2 \right] > 0$$

and

$$h_y(\tau, y) = y \frac{d\tau}{d\tau} + By (y - 1) \ln y + \frac{df}{d\tau} y^2(y - 1)$$

When $p$ is incorporated into $y'(\tau)$, the positive and negative of $y'(\tau)$ cannot be analyzed. When $\tau = 0$, the packet loss rate is 1, and the convergence speed is 0. When $\tau$ is increasing, it can effectively reduce the packet loss rate, and thus improve the convergence speed. When $\tau \to \tau_{\text{max}}$, the additional packet loss rate caused by caching (or queuing) will infinitely approach 0. The convergence speed decreases with the increase of $\tau$.

The extreme point of $y(\tau)$ can be obtained by solving the following equations

$$\begin{cases}
(2 + g)y + f \cdot y^2(y - 1) - 2 = 0 \\
y \frac{d\tau}{d\tau} + By (y - 1) \ln y + \frac{df}{d\tau} y^2(y - 1) = 0
\end{cases}$$

and the solution is

$$\tau = \frac{\tau_{\text{max}}}{2} \left(p_{ch} + \sqrt{p_{ch}^2 + \frac{4a}{\tau_{\text{max}}}}\right)$$

Simulation results

RWP (Random Waypoint) mobility model is selected to simulate the movement trajectories and information exchanges of nodes moving within a circular area with a radius of 10 m. Assume that the initial state values $x_i \in \mathbb{R}^{1 \times 1}$ on each node $i$ are uniformly distributed between 0 and 1. Table 2 provides the simulation parameters.

Figure 3(a)–(d) illustrates the variations of node states for $\tau = 4$, $\tau = 200$, $\tau = 400$ and $\tau = 310$, respectively. The $x$-axis represents the communication rounds, and the $y$-axis represents the state values of each node. It can be seen from Figure 3(a) that our

| Table 2. Simulation scenario settings. |
|---------------------------------------|
| Simulation parameters | Values |
| Mobile network model | RWP |
| Node communication radius | 2 m |
| Number of nodes | 40 |
| Moving speed | [0 m/s, 10 m/s] |
| Halt time | [2 s, 6 s] |
| Control gain coefficient $\omega$ | 1 |
| Maximum delay $\tau_{\text{max}}$ | 400 |
| Packet loss by channel noises | 0.01 |
algorithm converges just after 800 rounds, which is faster than 1000 rounds in Figure 3(b) and over 1000 rounds in Figure 3(c). It reveals that the convergence speed of consensus algorithm would decline as time delay increases.

Using equation (39), we can calculate that the optimal delay setting should be 310, as shown in Figure 3(d). It can be observed evidently from this subfigure that the convergence process starts with 400 rounds, which is much faster than both of the results above when \( \tau < 300 \) and \( \tau > 300 \) respectively.

Figure 4 reflects the impact of packet loss rate on the convergence speed. Therein, the x-axis represents the communication rounds, and the y-axis represents the state values of each node. Figure 4(a)–(c) illustrates the variations of node states for \( p_{ch} = 0 \), \( p_{ch} = 0.1 \) and \( p_{ch} = 0.3 \) when \( \tau = 310 \), respectively. It can be observed from Figure 4(a) that our algorithm converges just after 200 rounds, which is much faster than that in Figure 4(b) and (c) in the case of the same delay. This indicates that the convergence speed of consensus algorithm would decline as packet loss rate increases. Therefore, packet loss rate plays a more significant role in convergence speed in comparison with time delay.

### Conclusion

It can be seen that both time delay and packet loss rate on each link are allowed to be non-identical even time-varying under our control law by employing different quantized caching policies for different nodes. FL parameter aggregation can also be achieved in an asynchronous manner by caching some messages on a node for a period of time prior to being updated. Besides, the exchange of messages between neighbor nodes proceeds by broadcasting under our control law. However, that will probably lead to a bias of convergence result from its accurate value, which is our future work. The consensus conditions deduced analytically revealed that neither time delay nor packet loss rate affect the convergence of the consensus protocol, except its convergence speed. Nevertheless, the union of a sequence of directed network graphs is requested to be able to contain a directed spanning tree. As a
result, it is observed that the caching on mobile devices actually plays a critical role in consensus computing. It can be concluded that it is possible to operate in an ad hoc manner for FL participants, although the centralized operation mode cannot be replaced completely.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by the National Natural Science Foundation of China (grant no. 61771354).

ORCID iD

Xin Yan https://orcid.org/0000-0003-3630-8173

References

1. Wahab OA, Mourad A, Otrok H, et al. Federated machine learning: survey, multi-level classification, desirable criteria and future directions in communication and networking systems. *IEEE Commun Surv Tutor* 2021; 23(2): 1342–1397.
2. Lim WYB, Luong NC, Hoang DT, et al. Federated learning in mobile edge networks: a comprehensive survey. *IEEE Commun Surv Tutor* 2020; 22(3): 2031–2063.
3. Ye D, Yu R, Pan M, et al. Federated learning in vehicular edge computing: a selective model aggregation approach. *IEEE Access* 2020; 8: 23920–23935.
4. Niknam S, Dhillon HS and Reed JH. Federated learning for wireless communications: motivation, opportunities, and challenges. *IEEE Commun Mag* 2020; 58(6): 46–51.
5. Liu M, Yang K, Zhao N, et al. Intelligent signal classification in industrial distributed wireless sensor networks based industrial internet of things. *IEEE Trans Ind Inform* 2021; 17(7): 4946–4956.
6. Liu Y, Garg S, Nie J, et al. Deep anomaly detection for time-series data in industrial IoT: a communication-efficient on-device federated learning approach. *IEEE Intern Things J* 2021; 8(8): 6348–6358.
7. Zhang H and Hanzo L. Federated learning assisted multi-UAV networks. *IEEE Trans Veh Technol* 2020; 69(11): 14104–14109.

8. Bao W, Wu C, Guileg S, et al. Edge computing-based joint client selection and networking scheme for federated learning in vehicular IoT. *Chin Commun* 2021; 18(6): 39–52.

9. Sun H, Li S, Yu FR, et al. Toward communication-efficient federated learning in the internet of things with edge computing. *IEEE Intern Things J* 2020; 7(11): 11053–11067.

10. Lu Y, Huang X, Zhang K, et al. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. *IEEE Trans Veh Technol* 2020; 69(4): 4298–4311.

11. Ren W and Cao Y. Distributed coordination of multi-agent networks: emergent problems, models, and issues. London: Springer-Verlag, 2011.

12. Vecchio M, Amendola G and Ducange P. An integrated topology control framework to accelerate consensus in broadcast wireless sensor networks. *IEEE Trans Autom Control* 2016; 63(2): 1258–1267.

13. Olfati-Saber R and Murray RM. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Autom Control* 2004; 49(9): 1520–1533.

14. Savino HI, dos Santos CRP, Souza FO, et al. Conditions for consensus of multi-agent systems with time-delays and uncertain switching topology. *IEEE Trans Ind Electron* 2016; 63(2): 1258–1267.

15. Wang Z, Xu J, Song X, et al. Consensus conditions for multi-agent systems under delayed information. *IEEE Trans Circ Syst II Exp Brief* 2018; 65(11): 1773–1777.

16. Zheng Y and Wang L. Consensus of switched multiagent systems. *IEEE Trans Circ Syst II Exp Brief* 2016; 63(3): 314–318.

17. Kar S and Moura JMF. Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise. *IEEE Trans Signal Process* 2009; 57(1): 355–369.

18. Zong X, Li T, Yin G, et al. Stochastic consensusability of linear systems with time delays and multiplicative noises. *IEEE Trans Autom Control* 2018; 63(4): 1059–1074.

19. Zheng J, Xu L, Xie L, et al. Consensusability of discrete-time multiagent systems with communication delay and packet dropouts. *IEEE Trans Autom Control* 2019; 64(3): 1185–1192.

20. Chen G, Wang LY, Chen C, et al. Critical connectivity and fastest convergence rates of distributed consensus with switching topologies and additive noises. *IEEE Trans Autom Control* 2017; 62(12): 6152–6167.

21. Wang X, Li R, Wang C, et al. Attention-weighted federated deep reinforcement learning for device-to-device assisted heterogeneous collaborative edge caching. *IEEE J Select Area Commun* 2021; 39(1): 154–169.

22. Yu Z, Hu J, Min G, et al. Mobility-aware proactive edge caching for connected vehicles using federated learning. *IEEE Trans Intell Transp Syst* 2021; 22(8): 5341–5351.

23. Xu L, Zheng J, Xiao N, et al. Mean square consensus of multi-agent systems over fading networks with directed graphs. *Automatica* 2018; 95: 503–510.

24. Yang Y and Blum RS. Broadcast-based consensus with non-zero-mean stochastic perturbations. *IEEE Trans Inform Theory* 2013; 59(6): 3971–3989.

25. Zong X, Li T and Zhang J. Consensus control of discrete-time multiagent systems with time delays and multiplicative measurement noises. *Sci Sin Math* 2016; 46(10): 1617–1636.

26. Wu F, Mao X and Kloeden PE. Discrete Razumikhin-type technique and stability of the Euler–Maruyama method to stochastic functional differential equations. *Discr Cont Dyn Syst* 2013; 33(2): 885–903.

**Appendix I**

**Proof of Lemma 1**

**Proof.** Let

\[ z(k) := x(k) + \sum_{j=k-\tau}^{k-1} \varepsilon BKx(j) \]  

(41)

thus

\[ z(k + 1) = z(k) + (\varepsilon BK + A - I)x(k) \]  

(42)

We build a Lyapunov energy function as

\[ V_1(k) = z^T(k)Pz(k) \]  

(43)

where \( P \) is a positive-definite matrix and \( P^T = P \).

Substituting equation (43) into (42) gives

\[ V_1(k + 1) = z^T(k)Pz(k) + x^T(k)(PT + T^TP + T^TP^T)x(k) + 2[Tx(k)]^TP \sum_{j=k-\tau}^{k-1} \varepsilon BKx(j) \]  

(44)

where \( T = \varepsilon BK + A - I \). Since \( P^T = P \), there exists

\[ 2[Tx(k)]^TP \sum_{j=k-\tau}^{k-1} \varepsilon BKx(j) = \]

(45)

Due to \( 2x^T \gamma \leq \gamma x^2 + \frac{1}{2} ||y||^2 \), we have

\[ 2[Tx(k)]^T \left( \sqrt{P} \sum_{j=k-\tau}^{k-1} \varepsilon BKx(j) \right) \leq \]

(46)

\[ \delta \tau x^T(k)PTx(k) + \frac{1}{\tau} \left[ \sum_{j=k-\tau}^{k-1} \varepsilon BKx(j) \right]^T P \left[ \sum_{j=k-\tau}^{k-1} \varepsilon BKx(j) \right] \]

and
where

\[
\begin{align*}
&\sum_{j=-\tau}^{k-1} e^{BKx(j)} P \left[ \sum_{j=-\tau}^{k-1} e^{BKx(j)} \right] \\
&\tau \sum_{j=-\tau}^{k-1} x^T(j)(e^{BK})^T P(e^{BK})x(j)
\end{align*}
\]

Thus

\[
\begin{align*}
&2[\text{Tx}(k)]^T P \sum_{j=-\tau}^{k-1} e^{BKx(j)} \\
&\tau x^T(k)T^T PTx(k) + \\
&\sum_{j=-\tau}^{k-1} x^T(j)(e^{BK})^T P(e^{BK})x(j)
\end{align*}
\]

Then, we obtain

\[
\begin{align*}
V_1(k + 1) &\leq V_1(k) + \\
&\tau x^T(k)T^T PTx(k) + x^T(k)A_s x(k) + \\
&\sum_{j=-\tau}^{k-1} x^T(j)(e^{BK})^T P(e^{BK})x(j)
\end{align*}
\]

where \( A_s = PT + T^T P + T^T PT \).

We build another Lyapunov energy function as

\[
V_2(k) = \\
\sum_{s=-\tau}^{-1} \sum_{j=-\tau}^{k+s} x^T(j)(e^{BK})^T P(e^{BK})x(j)
\]

where \( x(j) \) is replaced with its update equation, that is

\[
\begin{align*}
&V_2(k + 1) = V_2(k) + \\
&x^T(k)(e^{BK})^T P(e^{BK})x(k) - \\
&\sum_{j=-\tau}^{k-1} x^T(j)(e^{BK})^T P(e^{BK})x(j)
\end{align*}
\]

Letting \( V(k) = V_1(k) + V_2(k) \), we get

\[
V(k + 1) \leq V(k) + x^T(k)P_1x(k)
\]

where

\[
P_1 = (\tau + 1)T^T PT + \tau(e^{BK})^T P(e^{BK}) + PT + T^T + \sigma^2(BK)^T P(BK)
\]

Hence, it is observed that \( \Pi_x = E[P_1] \).

For any \( s \geq 1 \), there exists

\[
\begin{align*}
&s^{k+1} E[V(k + 1)] \leq s^{k+1} E[V(k)] + \\
&s^{k+1} E[x^T(k)P_1x(k)]
\end{align*}
\]

where

\[
\begin{align*}
&s^{k+1} E[V(k + 1)] \leq E[V(0)] + \\
&\sum_{j=0}^{k} (s^{j+1} - s^j) E[V(j)] + \\
&\sum_{j=0}^{k} s^{j+1} E[x^T(j)P_1x(j)]
\end{align*}
\]

In the light of

\[
\begin{align*}
&V(k) = 2x^T(k)P \sum_{j=-\tau}^{k-1} e^{BKx(j)} + \\
&\sum_{s=-\tau}^{-1} \sum_{j=-\tau}^{k+s} x^T(j)(e^{BK})^T P(e^{BK})x(j) + \\
&\sum_{j=-\tau}^{k-1} x^T(j)(e^{BK})^T P(e^{BK})x(j) + \\
&x^T(k)P_1x(k)
\end{align*}
\]

and

\[
\begin{align*}
&E[V(k)] \leq 2\|x(k)\|^2\|P\| + \\
&3\tau^2\|P\| \cdot E[\|e^{BK}\|] \cdot \sum_{j=-\tau}^{k-1} E[\|x(j)\|^2]
\end{align*}
\]

the following inequality holds

\[
\begin{align*}
&s^{k+1} E[V(k + 1)] \leq C_1(s) + \\
&C_2(s) \left[ 1 - \frac{1}{s} \right] \sum_{j=0}^{k} s^{j+1} E\left[\|x(j)\|^2\right] + \\
&\sum_{j=0}^{k} s^{j+1} E[x^T(j)P_1x(j)]
\end{align*}
\]

where

\[
\begin{align*}
&C_1(s) = E[V(0)] + \\
&3\tau^2\|P\| \cdot E[\|e^{BK}\|] \cdot s\left(1 - \frac{1}{s}\right) \cdot \sum_{j=-\tau}^{k-1} E[\|x(j)\|^2]
\end{align*}
\]

\[
\geq 0.
\]

Considering that \( \Pi_x = E[P_1] < 0 \) and \( E[P_1] + \lambda_{\min}(-E[P_1])I_n \leq 0 \), we have

\[
\begin{align*}
&E[x^T(j)P_1x(j)] - \\
&\lambda_{\min}(E[P_1]x(j)] \leq 0
\end{align*}
\]

\[
\begin{align*}
&E[x^T(j)(-\lambda_{\min}(E[P_1]I_n)x(j)] = E[x^T(j)(\lambda_{\min}(E[P_1]I_n)x(j)] \leq 0
\end{align*}
\]
where
\[
E[x^T(j)P_1x(j)] = E[x^T(j)(-\lambda_{\text{min}}(I-E[P_1])x(j)] = \lambda_{\text{min}}E[P_1]E[x^T(j)x(j)]
\]
(61)

As a result, the following inequality holds
\[
s^{k} + 1E[V(k + 1)] \leq C_1(s) + 
\left[C_2(s)(1-s^{-1}) - \lambda_{\text{min}}(I-E[P_1])\right]\sum_{j=0}^{k}s^{j} + 1E\left[\|x(j)\|^2\right]
\]
(62)

Constructing the function
\[
H(s) = C_2(s)(1-s^{-1}) - \lambda_{\text{min}}(I-E[P_1])
\]
(63)
it can be easily seen that \(H(1) = -\lambda_{\text{min}}(I-E[P_1])<0\). In addition, since \(H'(s)>0\) and \(\lim H(s)>0\), there must be a unique zero point of \(H(s)\), that is \(s_0\), which enables \(H(s_0) = 0\). Let \(s^* \in (1, s_0)\) and \(H(s^*)<0\), we have
\[
s^{k} + 1E[V(k + 1)] \leq C_1(s^*) + 
H(s^*)\sum_{j=0}^{k}s^{j} + 1E\left[\|x(j)\|^2\right]
\]
(64)
which can be expressed as
\[
-H(s^*)\sum_{j=0}^{k}s^{j} + 1E\left[\|x(j)\|^2\right] \leq C_1(s^*).
\]
(65)
Choosing \(\gamma>0\) and \(s^* := e^\gamma\), there exists a constant \(C_3>0\) such that
\[
e^{\gamma k}E\left[\|x(j)\|^2\right] \leq C_3.
\]
(66)

According to the conclusion of moment exponential stability of discrete-time stochastic systems, it can be seen that the protocol can reach an alignment.

The lemma holds.

\section*{Appendix 2}

\subsection*{Proof of Lemma 2}

\textbf{Proof.} Assume that \(r = [r_1, ..., r_N]^T\), with the corresponding zero eigenvalue, is the left eigenvector of the Laplacian matrix in average \(\bar{L}(k)\). If \(\alpha\) denotes the convergence result
\[
r^T \cdot \bar{L}(k) = 0
\]
(67)

Let \(J(k)\) be the Jordan form associated with \(\bar{L}(k)\), thus \(\bar{L}(k) = PJ(k)P^{-1}\) where \(P\) is a reversible matrix. Since \(\bar{L}(k)P = PJ(k)\) and \(P^{-1}\bar{L}(k) = J(k)P^{-1}\), \(r\) is the first column and \(r^T\) is the first row of \(P\).

If \(J(k) = \tilde{Q}\) when \(k \to \infty\), we have
\[
R = PQP^{-1} = r \cdot r^T
\]
(68)
and
\[
\alpha \cdot 1 = R \cdot x(0)
\]
(69)
where \(1 = [1, ..., 1]^T\) is a unit vector and \(x(0) = [x_1(0), ..., x_N(0)]^T\) is the initial state vector.

In addition, there exists
\[
r^T \cdot (\alpha \cdot 1) = r^T \cdot R \cdot x(0)
\]
(70)
that is
\[
\alpha = r^T \cdot R \cdot x(0)/\sum_{i=1}^{N} r_i
\]
(71)
The unit vector, with the corresponding zero eigenvalue, is the left eigenvector of \(\bar{L}(k)\) for a balanced digraph, then
\[
1^T \cdot \bar{L}(k) = 0.
\]
(72)
As a result, we get
\[
\alpha = 1^T \cdot x(0)/N = \frac{1}{N}\sum_{i=1}^{N} x_i(0)
\]
(73)
The lemma holds.